

**ROMA Connect**

# **User Guide**

**Date**      **2024-07-10**

---

# Contents

---

<b>1 Service Overview.....</b>	<b>1</b>
1.1 What Is ROMA Connect?.....	1
1.2 Application Scenarios.....	6
1.2.1 Smart Campus Integration.....	6
1.2.2 Industrial Internet Integration.....	8
1.2.3 Application & Data Integration of Corporation Groups.....	9
1.3 Product Specifications.....	11
1.4 Resource Quotas.....	14
1.5 Constraints.....	16
1.6 Basic Concepts.....	17
1.7 Related Services.....	20
<b>2 Getting Started.....</b>	<b>22</b>
2.1 Overview.....	22
2.2 Using ROMA Connect for Data Integration.....	23
2.3 Using ROMA Connect for Service Integration.....	29
2.4 Using ROMA Connect for Message Integration.....	40
2.5 Using ROMA Connect for Device Integration.....	46
<b>3 Getting Started.....</b>	<b>57</b>
<b>4 Instances.....</b>	<b>59</b>
4.1 Creating a ROMA Connect Instance.....	59
4.1.1 Preparing Required Resources.....	59
4.1.2 Creating an Instance.....	61
4.2 Managing Instances.....	65
4.3 Instance Configuration Parameters.....	69
<b>5 Integration Application Management.....</b>	<b>72</b>
5.1 Creating an Integration Application.....	72
5.2 Configuring Integration Application Authorization.....	74
<b>6 Data Integration Guide.....</b>	<b>77</b>
6.1 Usage Introduction.....	77
6.2 Connecting to Data Sources.....	79
6.3 Creating a Data Integration Task.....	82

6.3.1 Configuring Basic Information.....	82
6.3.2 Configuring Source Information.....	85
6.3.3 Configuring Destination Information.....	180
6.3.4 Configuring a Data Mapping Rule.....	225
6.3.5 (Optional) Configuring Fault Information Storage.....	228
6.3.6 (Optional) Configuring the Post-Integration Operation.....	228
6.4 Creating a Composite Data Integration Task.....	230
6.4.1 Configuring Oracle CDC (LogMiner).....	230
6.4.2 Configuring Oracle CDC (XStream).....	235
6.4.3 Configuring MySQL CDC (Binlog).....	244
6.4.4 Configuring SQL Server CDC.....	246
6.4.5 Creating a Composite Task.....	250
6.5 Starting or Stopping a Data Integration Task.....	256
6.6 Managing a Data Integration Task.....	257
6.6.1 Viewing Data Integration Tasks.....	257
6.6.2 Importing or Exporting Data Integration Tasks.....	259
<b>7 Service Integration Guide.....</b>	<b>260</b>
7.1 Usage Introduction.....	260
7.2 Exposing APIs.....	262
7.2.1 Creating an API Group.....	263
7.2.2 Binding Domain Names.....	264
7.2.3 (Optional) Creating a Load Balance Channel.....	265
7.2.4 Creating an API.....	268
7.2.5 Debugging an API.....	280
7.2.6 (Optional) Creating the Environment and Environment Variables.....	281
7.2.7 Publishing an API.....	283
7.2.8 (Optional) Authorizing Apps to Call an API.....	284
7.3 Creating and Exposing Data APIs.....	285
7.3.1 Connecting to Data Sources.....	286
7.3.2 Creating a Data API.....	286
7.3.3 (Optional) Authorizing Apps to Call an API.....	292
7.4 Creating and Exposing Function APIs.....	293
7.4.1 Creating a Function API.....	293
7.4.2 (Optional) Authorizing Apps to Call an API.....	296
7.5 Calling an API.....	297
7.5.1 Calling an Open API.....	298
7.5.2 Configuring CORS for APIs.....	300
7.5.3 Appendix: API Error Codes.....	304
7.6 API Management.....	307
7.6.1 Viewing API Calling Information.....	307
7.6.2 Taking an API Offline.....	308
7.6.3 Taking a Custom Backend Offline.....	308

7.6.4 Importing and Exporting APIs.....	309
7.6.5 Importing and Exporting Custom Backends.....	312
7.6.6 Appendix: Extended Swagger Definition of APIs.....	314
7.6.7 Appendix: Extended Definition of Swagger for Custom Backends.....	330
7.7 Control Policy Management.....	334
7.7.1 Configuring a Request Throttling Policy.....	334
7.7.2 Configuring an Access Control Policy.....	337
7.7.3 Configuring a Client Quota Policy.....	339
7.7.4 Configuring a Client Access Control Policy.....	340
7.8 Configuring a Custom Authorizer.....	341
7.8.1 Creating a Frontend Custom Authorizer.....	341
7.8.2 Creating a Backend Custom Authorizer.....	345
7.9 Configuring Signature Verification for Backend Services.....	347
7.10 Configuring API Cascading.....	349
<b>8 Message Integration Guide.....</b>	<b>355</b>
8.1 Usage Introduction.....	355
8.2 Creating a Topic.....	356
8.3 (Optional) Granting Permissions for Topics.....	358
8.4 Connecting to a Topic.....	359
8.5 Topic Management.....	363
8.5.1 Querying Topic Messages.....	363
8.5.2 Importing and Exporting Topics.....	364
<b>9 Device Integration Guide.....</b>	<b>365</b>
9.1 Usage Introduction.....	365
9.2 Creating a Product.....	367
9.3 Registering a Device.....	373
9.4 Connecting Devices to ROMA Connect.....	375
9.5 Product Management.....	377
9.5.1 Viewing Products.....	377
9.5.2 Importing and Exporting Products.....	378
9.5.3 Creating a Product Template.....	379
9.6 Device Management.....	383
9.6.1 Viewing Devices.....	383
9.6.2 Importing and Exporting Devices.....	389
9.6.3 Creating a Device Group.....	390
9.6.4 Appendix: EdgeShell Access Description.....	391
9.7 Rule Engine.....	397
9.7.1 Configuring Data Forwarding Rules.....	397
9.7.2 Importing or Exporting Rules.....	404
9.8 Subscription Management.....	405
9.8.1 Subscribing to Device Notifications.....	405
9.8.2 Appendix: Packets of Subscribed Notification Messages.....	406

<b>10 Data Source Management.....</b>	<b>408</b>
10.1 Data Sources Supported by ROMA Connect.....	409
10.2 Connecting to an API Data Source.....	411
10.3 Connecting to an ActiveMQ Data Source.....	414
10.4 Connecting to an ArtemisMQ Data Source.....	417
10.5 Connecting to a DB2 Data Source.....	419
10.6 Connecting to a DIS Data Source.....	420
10.7 Connecting to a DWS Data Source.....	422
10.8 Connecting to a GaussDB 100 Data Source.....	423
10.9 Connecting to a GaussDB 200 Data Source.....	425
10.10 Connecting to an FTP Data Source.....	427
10.11 Connecting to an FI HDFS Data Source.....	428
10.12 Connecting to an FI Hive Data Source.....	430
10.13 Connecting to an FI Kafka Data Source.....	432
10.14 Connecting to an HL7 Data Source.....	433
10.15 Connecting to a HANA Data Source.....	435
10.16 Connecting to a HIVE Data Source.....	437
10.17 Connecting to an LDAP Data Source.....	438
10.18 Connecting to an IBM MQ Data Source.....	439
10.19 Connecting to a Kafka Data Source.....	441
10.20 Connecting to a MySQL Data Source.....	442
10.21 Connecting to a MongoDB Data Source.....	445
10.22 Connecting to an MRS Hive Data Source.....	447
10.23 Connecting to an MRS HDFS Data Source.....	449
10.24 Connecting to an MRS HBase Data Source.....	451
10.25 Connecting to an MRS Kafka Data Source.....	453
10.26 Connecting to an OBS Data Source.....	455
10.27 Connecting to an Oracle Data Source.....	456
10.28 Connecting to a PostgreSQL Data Source.....	459
10.29 Connecting to a Redis Data Source.....	461
10.30 Connecting to a RabbitMQ Data Source.....	462
10.31 Connecting to an SAP Data Source.....	464
10.32 Connecting to an SNMP Data Source.....	465
10.33 Connecting to a SQL Server Data Source.....	468
10.34 Connecting to a TaurusDB Data Source.....	469
10.35 Connecting to a WebSocket Data Source.....	472
10.36 Connecting to a Custom Data Source.....	473
<b>11 Asset Management.....</b>	<b>475</b>
11.1 Connectors.....	475
11.1.1 Creating a Connector.....	475
11.1.2 Publishing a Connector.....	478
11.1.3 RESTful API Specifications of Connectors.....	479

11.2 Restoring/Migrating Integration Assets.....	484
11.2.1 Overview.....	484
11.2.2 Preparations.....	487
11.2.3 Modifying Instance Configurations.....	488
11.2.4 Importing Integration Assets.....	488
11.2.5 Modifying Integration Application Configurations.....	489
11.2.6 Modifying Data Source Configurations.....	489
11.2.7 Modifying API Configurations.....	490
11.2.8 Modifying Device Configurations.....	493
11.2.9 Service Interconnection Adaptation and Verification.....	494
<b>12 FAQs.....</b>	<b>496</b>
12.1 Common Operations.....	496
12.1.1 Region and AZ.....	496
12.1.2 Can I Use an Existing VPC to Create an ROMA Connect Instance?.....	497
12.1.3 Does ROMA Connect Have Quota Limits?.....	497
12.2 Data Integration Tasks.....	498
12.2.1 Which Types of Data Are Supported by FDI Databases?.....	498
12.2.2 What Is a Trace Number? Can Data Be Traced by Using a Trace Number?.....	499
12.2.3 Does an FDI Task Support Clearance of a Destination Table?.....	499
12.2.4 Can a Task Collect Data from One Table to Another Table?.....	499
12.2.5 Is the FDI Task Created by One User Visible to Another User Under the Same Account?.....	499
12.2.6 How Do I Configure FDI to Connect to MRS Hive of Other Tenants Through a Public Network?..	499
12.2.7 Will Collected Data Be Updated After a File Is Parsed from OBS to the RDS Database?.....	499
12.2.8 Why Are the Column Values Incorrect When the CSV File on the Source OBS Is Parsed to the Relational Database?.....	500
12.2.9 Can Data Be Integrated into the Destination When the Data Types of Source and Destination Fields of MRS Hive Do Not Match?.....	500
12.2.10 Is the Mapping Between MRS Hive or MRS HBase and MongoDB Case-sensitive When It Is Manually Entered?.....	500
12.2.11 Does MRS Hive Support Partitioning?.....	500
12.2.12 How Can I Configure the Time for Triggering a Scheduled Task of FDI?.....	501
12.2.13 How Do I Set the Custom Period for the API Data Source at the Source?.....	504
12.3 Composite Data Integration Tasks.....	507
12.3.1 How Do Source Database Restarts Affect Composite Tasks?.....	507
12.3.2 How Do I Handle an Exception of the CDC Function (Using the XStream Mode) Caused by Unexpected Oracle Database Restart?.....	507
12.4 Service Integration.....	507
12.4.1 Which Languages Does APIC Support for SDKs?.....	508
12.4.2 Does APIC Support Multiple Backend Endpoints?.....	508
12.4.3 Which Error Codes Will Be Displayed When I Use APIC?.....	508
12.4.4 How Can I Ensure the Security of Backend Services Invoked by APIC?.....	511
12.4.5 Do I Need to Publish an API Again After Modification?.....	511
12.4.6 What Can I Do If an API Published in a Non-RELEASE Environment Is Inaccessible?.....	511

12.4.7 Can I Invoke Different Backend Services by Publishing an API in Different Environments?.....	511
12.4.8 What Is the Maximum Size of an API Request Package?.....	511
12.4.9 Can I Modify an API After It Is Deployed on the Custom Backend?.....	511
12.4.10 How Does APIC Throttle Requests?.....	511
12.4.11 What Are the Causes of an API Calling Failure?.....	512
12.4.12 Can I Call an API If the API Is Not Bound to an EIP?.....	512
12.4.13 Which Data Sources Does APIC Support?.....	512
12.5 Message Integration.....	513
12.5.1 Does MQS Support Cross-VPC and Cross-Subnet Access?.....	513
12.5.2 Do MQS Instances Support Cross-Subnet Access?.....	513
12.5.3 How Do I Select and Configure a Security Group?.....	513
12.5.4 Can I Access an MQS Instance Through a Public Network?.....	515
12.5.5 Why Can't I View the Subnet and Security Group Information When Creating an MQS Instance? .....	515
12.6 Device Integration.....	515
12.6.1 Which Languages Does a LINK SDK Support?.....	515
12.6.2 Which Communication Protocols Are Supported by LINK?.....	515
12.6.3 Why Can't My Device Be Connected to LINK?.....	515
12.6.4 Why Does the Data Destination of a Rule Engine Fail to Receive Messages from My Device After It Is Connected to LINK?.....	516
<b>13 Troubleshooting.....</b>	<b>518</b>
13.1 Common Data Integration Tasks.....	518
13.1.1 Garbled Characters Are Displayed When Data Is Written to MRS Hive at the Destination.....	518
13.1.2 All Data Is Written to the First Field of MRS Hive.....	518
13.1.3 An Error Message Is Displayed at the Destination Indicating Task Execution Times Out.....	519
13.1.4 Error Message "could only be written to 0 of the 1 minReplication nodes. There are 2 datanode(s) running and 2 node(s) are excluded in this operation" Is Reported at the Destination During Data Integration from MySQL to MRS Hive.....	519
13.1.5 Error Message "Illegal mix of collations for operation 'UNION'" Is Displayed at the Source Database During MySQL-to-MySQL Data Integration.....	520
13.1.6 Data May Be Lost When Incremental Data Collection Is Performed from the Source MySQL on an Hourly Basis.....	520
13.1.7 Error Message "401 unauthorized" Is Displayed at the Source During API-to-MySQL Data Integration.....	520
13.1.8 Error Message "cannot find record mapping field" Is Displayed at the Destination During Kafka-to- MySQL Data Integration.....	521
13.1.9 Error Message "connect timeout" Is Displayed at the Source During Scheduled API-to-MySQL Data Integration.....	521
13.1.10 FDI Fails to Obtain Data During Real-Time Kafka-to-MySQL Data Integration Although Data Exists in MQS Topics.....	521
13.1.11 Value of the Source Field of the tinyint(1) Type Is Changed from 2 to 1 at the Destination During Scheduled MySQL-to-MySQL Data Integration.....	522
13.2 Composite Data Integration Tasks.....	522
13.2.1 Data Fails to Be Written Because the RowId Field Type Is Incorrectly Configured in the Destination Table.....	522

---

13.2.2 Error Message "binlog probably contains events generated with statement or mixed based replication forma" Is Displayed When the Binlog of the MySQL Database Is Read.....	522
13.2.3 Data Still Fails to Be Written After an FDI Task Failure Is Rectified.....	523
13.2.4 Camel Fails to Access the Database Because Table Names Contain Garbled Characters.....	523
13.2.5 Inserted Data Violates the Non-null Constraint.....	524
13.2.6 FDI Task Fails to Be Executed Because DWS Changes to the read-only State.....	524
13.2.7 Speed of Writing Data to DWS Becomes Slower.....	525
13.3 Data Sources.....	525
13.3.1 Troubleshooting a Data Source Connection Test Failure.....	525
13.3.2 MRS Hive Data Source Connection Failed.....	526
13.3.3 FTP Data Source Connection Failed.....	527
13.3.4 OBS Data Source Connection Failed.....	527
13.3.5 Kafka Data Source Connection Failed.....	527
13.4 Service Integration.....	528
13.4.1 Backend Service Fails to Be Invoked.....	528
13.4.2 Error Message "No backend available" Is Displayed When an API Is Called.....	528
13.4.3 Error Message "The API does not exist or has not been published in an environment" Is Displayed When an API Is Called Using JavaScript.....	528
13.5 Device Integration.....	529
13.5.1 Error Message "java.lang.IllegalArgumentException: {IP address}_{timestamp}" Is Displayed During Demo Running.....	529



# 1 Service Overview

---

[What Is ROMA Connect?](#)

[Application Scenarios](#)

[Product Specifications](#)

[Resource Quotas](#)

[Constraints](#)

[Basic Concepts](#)

[Related Services](#)

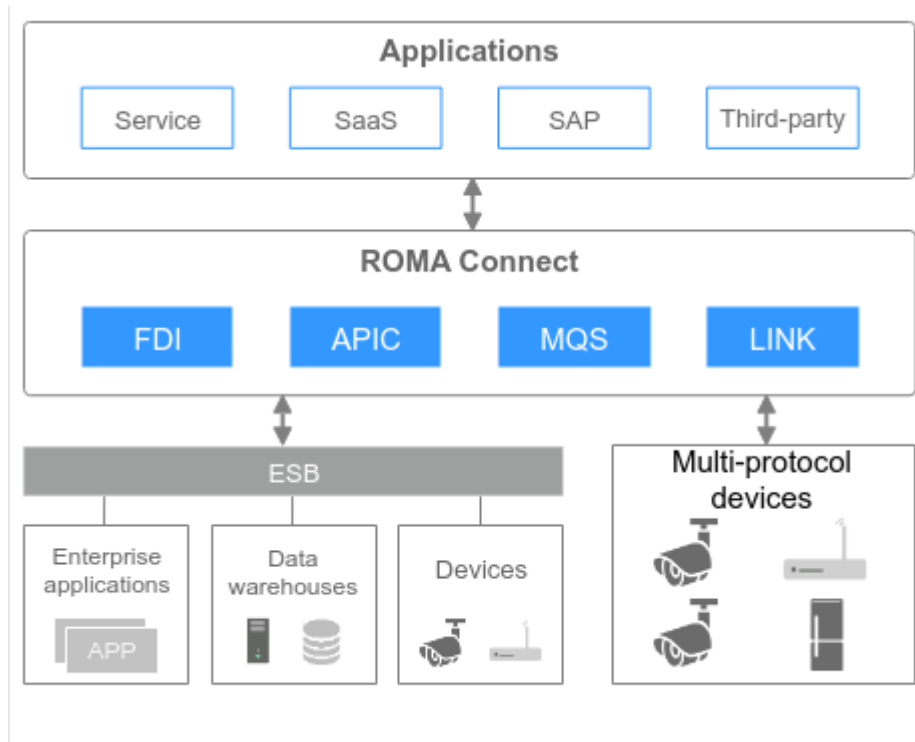
## 1.1 What Is ROMA Connect?

Enterprises face many difficulties in digital transformation:

- There is no unified way to integrate device information.
- Data formats are diverse, which makes it difficult to transmit and integrate data.
- There is no convenient way to share data and backend services with partners.
- There is no secure information channel for cloud and on-premises applications across different networks.

ROMA Connect is a full-stack application & data integration platform. It focuses on application and data connections and applies to multiple common scenarios of enterprises. ROMA Connect provides lightweight message, data, API, device, and model integration to simplify the enterprise cloudification flow and support cross-regional integration for cloud and on-premises applications, helping enterprises achieve digital transformation.

**Figure 1-1** Overall ROMA Connect architecture



ROMA Connect consists of four components: data integration (FDI, short for Fast Data Integration), service integration (APIC, short for API Connect), message integration (MQS, short for Message Queue Service), and device integration (LINK).

## FDI

FDI is a data integration component of ROMA Connect. FDI supports flexible, fast, and non-intrusive data integration between multiple data sources, such as text, messages, APIs, and relational and non-relational data. It implements data integration across equipment rooms, data centers, and clouds, and supports automatic deployment, O&M, and monitoring of integrated data.

For example, if an enterprise and its partners use different data sources, it is difficult to achieve effective information transmission. FDI provides multiple methods to convert mainstream data source formats such as MySQL, Kafka, and API. FDI can also work with other services, such as [Data Ingestion Service \(DIS\)](#) and [GaussDB\(DWS\)](#), to store, convert, and analyze big data.

**Table 1-1** FDI functions

Function	Introduction
Lifecycle management of data integration tasks	FDI allows you to modify data integration task information and view running reports, run logs, and status of data integration tasks.

Function	Introduction
Flexible data reading and writing	<ul style="list-style-type: none"> <li>• Reads and writes various types of data by fragment, such as MySQL data, text files, messages, and APIs.</li> <li>• Supports automatic recovery of tasks when the service is restored after an unexpected interruption occurs.</li> <li>• Supports task scheduling, monitoring, and resumable reading.</li> </ul>
Reliable data transmission channel	<p>FDI can continuously monitor data in data channels and supports concurrent execution of more than 100 threads. It monitors the message queue in real time and writes data to the target queue in real time.</p>
Task scheduling	<p>FDI provides comprehensive, flexible, and highly available task scheduling services and supports data integration through APIs or messages. It schedules tasks based on time and data volume rules. FDI assigns tasks to the plug-ins based on the task configuration, and monitors and records the task execution status.</p> <p>Enterprises can select different data integration modes to suit their service requirements.</p> <ul style="list-style-type: none"> <li>• Incremental real-time integration is applicable to scenarios in which data changes need to be monitored in real time, for example, collecting real-time parameters of devices on the production line.</li> <li>• Full real-time integration is ideal for scenarios in which all historical data needs to be monitored in real time, for example, collecting statistics on the supplier shipments.</li> <li>• Incremental scheduled integration is ideal for scenarios in which data changes need to be monitored for a period of time. For example, enterprises use new production policies to verify whether production efficiency meets expectations.</li> <li>• Full scheduled integration is ideal for scenarios in which all historical data needs to be monitored for a period of time, for example, collecting statistics on the number of vehicles entering or leaving a campus during peak and off-peak hours.</li> </ul>
API monitoring	<p>FDI monitors the running status of data integration tasks and processes abnormal tasks to ensure service running.</p>

## APIC

APIC is an API integration component of ROMA Connect. It opens data and backend services as APIs to simplify data sharing and service provisioning and reduce the cost on interconnection between enterprises. APIC provides SDKs and

sample code in different programming languages to simplify the process of opening up backend services as APIs.

For example, if a company headquarters integrates its IT system with those of its branches in different regions, it is too complex to directly access each other's database and information disclosure may occur. If APIs are used to access databases and security is enhanced for API call, cross-network and cross-regional collaboration can be achieved.

**Table 1-2** APIC functions

Function	Introduction
API lifecycle management	The lifecycle of an API involves creating, publishing, removing, and deleting the API.
Simple debugging tool	APIC provides an inline debugging tool to simplify API development and reduce maintenance costs.
Version management	An API can be published in different environments to meet version upgrade requirements.
Request throttling	Request throttling controls the maximum number of times an API can be called by a user or an app within a time period.  The throttling can be accurate to the second, minute, hour, or day. Special applications can be configured to be not affected by request throttling policies.
Monitoring statistics	APIC provides real-time, visualized API monitoring in terms of requests and errors.
Environment variables	When an API is published to different environments, the specified header parameters and special values are added to the API call request header to distinguish different environments. During publication, the variable is replaced with the environment variable value to ensure that the definition of the API does not change.
Custom backend	The custom backend supports data APIs and function APIs. <ul style="list-style-type: none"> <li>• A custom data API allows enterprises to connect a database to APIC as a backend service and convert data service capabilities into REST APIs.</li> <li>• A custom function API is similar to a simplified function service. You can compile custom scripts or functions on the APIC backend as a backend service for the frontend to invoke.</li> </ul>

## MQS

MQS is a message integration component of ROMA Connect. Kafka-based MQS uses a unified message access mechanism to provide enterprises with secure and standard message channels for cross-network access.

For example, if an enterprise and its partners use different message systems, interconnection between the message systems is costly, and message transmission after the interconnection may not be reliable or secure. To address these issues, the Kafka protocol can be used for communication between the enterprise and its partners. In this way, MQS functions as a message transfer station to provide secure and reliable message transmission. Specifically, the enterprise can create multiple topics, set the permission for each partner to subscribe to these topics, and publish messages to the topics. Then, partners can subscribe to the topics to obtain messages.

**Table 1-3** MQS functions

Function	Introduction
Kafka-native basic functions	MQS supports topic management and message publishing and subscription after being connected to the client. It also supports visualized operations on the ROMA Connect console, including topic creation and management, user management, permission configuration, and message query.
Monitoring and alarming	MQS allows you to configure monitoring metrics from multiple dimensions, such as instances, nodes, topics, and consumer groups. In addition, MQS allows you to configure alarm rules so that alarms can be generated if an exception occurs.
Message tracking	MQS can query the complete link information of message production and consumption, obtain the status of any message, and provide valid data for troubleshooting.

## LINK

LINK is a device integration component of ROMA Connect. LINK uses the standard Message Queue Telemetry Transport (MQTT) protocol to connect devices, helping enterprises quickly and easily manage devices on the cloud.

In industrial scenarios, device information and parameters involved in the production process are scattered. If a fault occurs in a production line, it takes a long time to manually collect information and parameters for each device. LINK connects devices to IT systems or big data platforms, and uploads information such as device running status to these platforms so that enterprise customers can see information about all devices graphically and therefore quickly locate faults. In addition, enterprise customers can configure the upper thresholds for device parameters to rule engines of LINK. If real-time parameters of a device are close to the upper thresholds, an alarm notification is sent to users to remind them to stop the device and perform maintenance.

**Table 1-4** LINK functions

Function	Introduction
Publishing and subscribing to messages	LINK supports the standard MQTT protocol. Enterprises can use open-source device SDKs based on this MQTT protocol to easily connect devices to the cloud for message publishing and subscription.
Message exchange between devices and backend applications	<p>You can configure a rule engine on the LINK console to enable a device to communicate with other devices, backends, and other cloud services.</p> <p>LINK supports rule engines to forward data to MQS. Third-party services obtain data through MQS to implement asynchronous message communication between devices and third-party services.</p>
Low-latency access for massive numbers of devices	LINK supports horizontal expansion of Broker and long connections of millions of devices.
Two-way synchronization between devices and applications	<p>LINK supports profile definition and binds the profile with a device shadow. This allows users to implement two-way synchronization of configuration data and status data between devices and applications.</p> <p>On the one hand, users can set configuration parameters to the device shadow through APIs. When a device is online or goes online, the configuration parameters can be obtained from the device shadow. On the other hand, devices can report their statuses to the device shadow. When a user queries the device status, the user only needs to query the device shadow instead of directly communicating with the device.</p>
Secure information transmission	LINK provides authorization certification for devices and applications and bidirectional binding authorization for topics to ensure device security and uniqueness. It provides TLS-based data transmission channels for secure message transmission.

## 1.2 Application Scenarios

### 1.2.1 Smart Campus Integration

Many difficulties are encountered in smart campus management:

- Customized management systems hinder information collection and sharing. Buildings in a campus have different structures. Enterprises can customize subsystems for each building to collect all information on each one. However, after customization, the differences between subsystems hinder information

collection and sharing, resulting in difficulty in information transmission. This reduces the "smart" level of a campus.

- Diversified devices and complex data collection make it difficult to implement system linkage.

In scenarios such as vehicle entrance and exit management, visitor registration, and campus asset management, it is difficult to implement linkage management due to the complexity of data collection and centralization.

- The status of important devices cannot be monitored remotely in real time. For example, faulty street lamps cannot be alerted and must be manually repaired in a traditional campus, resulting in passive maintenance.

ROMA Connect has a complete set of integration solutions involving devices, data, and services to help enterprises build smart campuses.

- **Efficient interconnection with various devices from different vendors**

Information about devices from different vendors, such as cameras, turnstiles, and air conditioners, is sent to ROMA LINK using the standard MQTT protocol. In addition, LINK is connected to multiple IoT platforms, eliminating the need to collect data from each platform separately.

- **Data base construction for providing standard data services**

FDI and MQS quickly integrate all data and open the data to different backend services of an enterprise. For example, vehicle data in turnstile systems, device status in asset management systems, and switch-on/switch-off and device information in street lamp systems are transmitted to backend services in real time or in asynchronous mode for analysis and linkage management.

In addition, the high scalability design provided by ROMA Connect supports huge data transmission and storage on the campus network, improving data transmission efficiency.

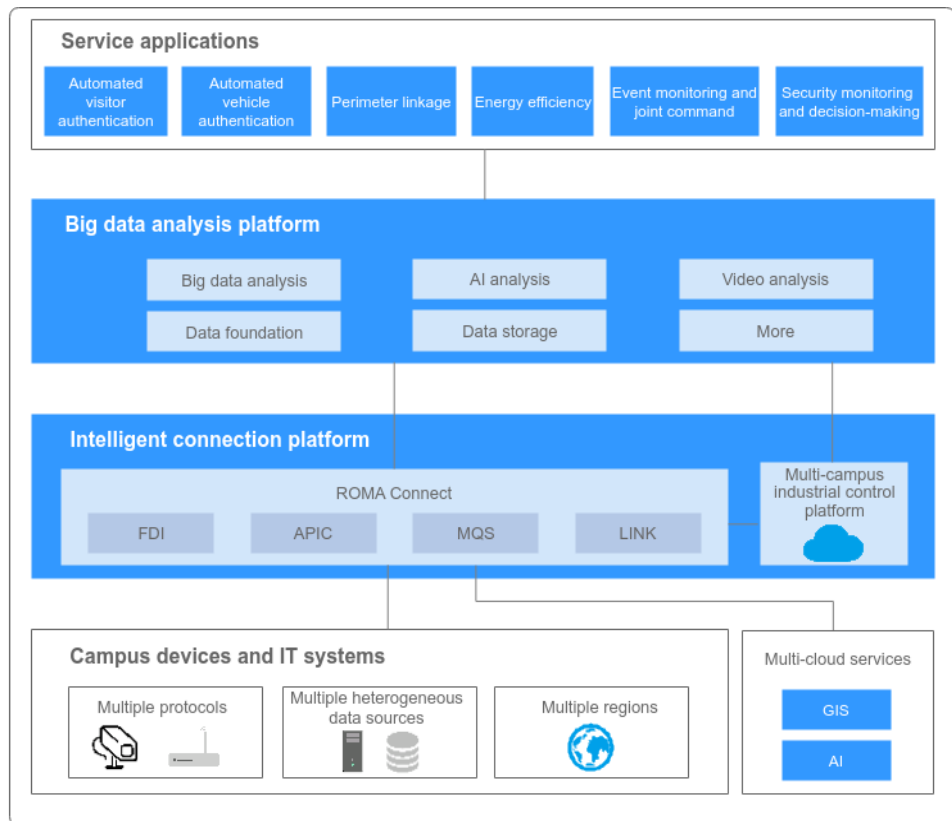
- **Integration of IT, OT, and AI for building an intelligent operation center**

ROMA Connect provides a channel for data integration and sharing. Enterprises, then, can use the enterprise-grade AI, video analysis, and big data services to build a smart campus.

- **Centralized and distributed architecture for supporting campus services**

Enterprises holding large campuses often need to manage multiple campuses. The centralized and distributed architecture of ROMA Connect helps these enterprises integrate data from multiple campuses onto the same platform and assists them in managing the distributed and centralized operations based on actual conditions.

**Figure 1-2 Smart campus integration**



## 1.2.2 Industrial Internet Integration

There are several typical problems in the digital transformation of the manufacturing industry:

- Difficulty in integrating device and environment data

To monitor and manage production devices of various brands and types in real time, device and environment data need to be collected and uploaded. However, such devices use different data formats and database standards, which makes it difficult to integrate device and environment data.

- Difficulty in preventing device faults

In a factory, any machine fault may have a huge impact on the entire assembly production line.

- Difficulty in optimizing production strategies and decision-making of enterprises

Different formats of collected data result in difficult data analysis. Therefore, it is a challenge for enterprises to optimize existing production strategies based on the collected data and to determine whether to execute new production strategies.

Leveraging the enterprise-class big data analysis solution, ROMA Connect helps the manufacturing industry transform to IoT integration through data collection and integration, and finally achieves the "smart" vision.

- **Device digitalization and integration**



ROMA Connect uses different methods, such as MQTT, and gateway, to connect various types of devices to enterprise backends, implementing bidirectional communication.

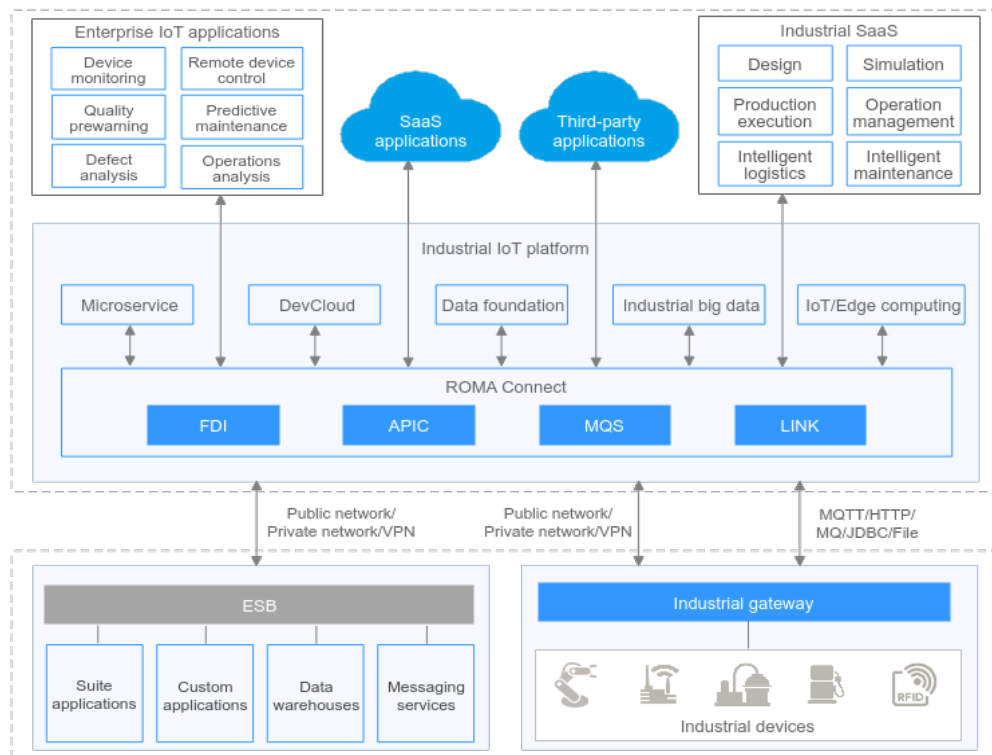
- **Fault prediction and alarming**

Information about all devices is integrated on the ROMA Connect console for real-time monitoring and prewarning analysis. Once parameters of a device become abnormal, ROMA Connect generates an alarm on the console and notifies the owner of repairing the device. If the real-time status of a device deviates from the normal data range, a notification is sent to device maintenance personnel to repair the device in a timely manner.

- **Data conversion and analysis**

ROMA Connect FDI imports data generated by industry SaaS services to ROMA Connect and transmits the data to **MapReduce Service (MRS)**, helping enterprises analyze big data and optimize production strategies.

**Figure 1-3 Industrial Internet integration**



### 1.2.3 Application & Data Integration of Corporation Groups

The integration between a parent company and its subsidiaries and between the corporate group and its partners faces bottlenecks:

- **Geographical differences**

The headquarters, branches, and partners are located in different regions and use different time zones. This reduces the timeliness and reliability of data.

- **Different cloud services**

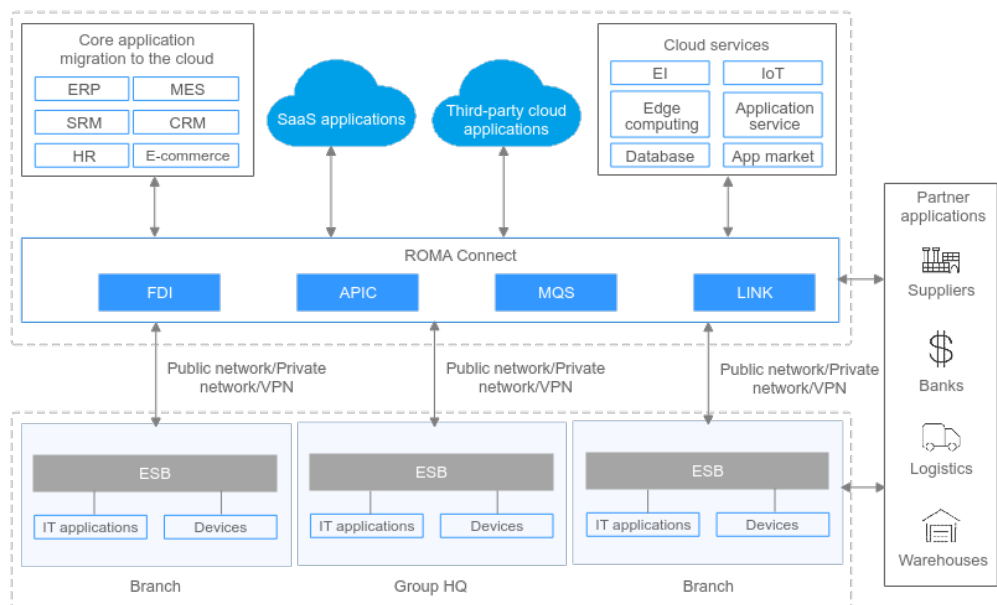
The cloud services used by the headquarters, branches, and partners are different. Therefore, it is difficult to invoke different cloud services.

- **Network differences**  
The networks used by the headquarters, branches, and partners are different. Therefore, interconnection between the public networks, private networks, and VPNs is difficult.

ROMA Connect helps corporation groups implement integration between the headquarters and branches and between the groups and their partners. As shown in **Figure 1-4**, ROMA Connect supports the following scenarios:

- **Cross-regional integration:** The headquarters, branches, and partners located in different regions transmit their device information, data, and messages to ROMA Connect. ROMA Connect performs operations such as device information visualization, alarm monitoring, data conversion, and message transmission to streamline regional restrictions, implement integration and governance for regional businesses and share group information, ensuring the reliability of service integration.
- **Cross-cloud integration:** APIC converts SaaS applications and third-party cloud applications into API data. Then, enterprises call these APIs to integrate different cloud applications, ensuring seamless interconnection between services on the cloud.
- **Cross-network integration:** ROMA Connect is used to implement secure cross-network interconnection with partners' service systems. Enterprises upload data and information required by partners to ROMA Connect. ROMA Connect then converts the data formats and integrates data based on the partners' requirements. After an enterprise integrates data and messages, partners can access ROMA Connect to obtain related information.

**Figure 1-4** Application & data integration of corporation groups



Application & data integration through ROMA Connect brings the following benefits to enterprises:

- Builds a unified platform for managing multiple cloud services and applications, simplifying management processes and helping enterprises achieve digital transformation.

- Enables information sharing between headquarters, branches, and partners.
- Supports large-scale integrated services, distributed deployment, automatic scaling, and low latency, ensuring service performance and reliability.

## 1.3 Product Specifications

### ROMA Connect Instance Specifications

The following table lists the ROMA Connect instance specifications in each edition.

**Table 1-5** ROMA Connect instance specifications

Instance Edition	Number of Systems Supported	Number of Connections Supported	Applicability
Basic	5 to 10	25	Small-sized enterprises
Professional	10 to 20	80	Small- and medium-sized enterprises
Enterprise	20 to 30	200	Medium- and large-sized enterprises
Platinum	More than 30	800	Large-sized enterprises

The number of connections and the number of systems are a reference that you can use to select a ROMA Connect instance edition. They do not limit the number of resources (such as data integration tasks, APIs, and message topics) in an instance and are not directly associated with the **resource quota** of the instance.

- **Number of systems:** A system refers to a user's service system, and the number of systems refers to the number of service systems interconnecting with a ROMA Connect instance. You can set up multiple connections between a service system and a ROMA Connect instance.
- **Number of connections:** A connection refers to an interaction between a service system and ROMA Connect. The number of connections varies depending on the functional module in ROMA Connect that you want to connect. The following table describes the mappings between the number of resources and the number of connections.

**Table 1-6** Mappings between the number of resources and the number of connections

Functional Module	Mapping
FDI	One data integration task occupies one connection.
APIC	<ul style="list-style-type: none"> <li>• 10 hosting APIs (that do not use custom backends as backend services) occupy one connection.</li> <li>• Two function APIs or data APIs occupy one connection.</li> </ul>
MQS	Three topics occupy one connection.
LINK	1000 devices occupy one connection.

## FDI Specifications

The following table lists the FDI specifications supported by a ROMA Connect instance.

**Table 1-7** FDI specifications

Instance Edition	Relational Database Collection Throughput (Mbit/s)	Message System Collection Throughput (Mbit/s)	Unstructured Storage Collection Throughput (Mbit/s)
Basic	0.5	4	2
Professional	1	5	3
Enterprise	1.5	7	4.5
Platinum	2	10	6

## APIC Specifications

The following table lists the APIC specifications supported by a ROMA Connect instance.

**Table 1-8** APIC specifications

ROMA Connect instance specifications	API Access (TPS)	API Access (Authorization Relationship Per Minute)
Basic	4000	120,000 Note: One authorization relationship is counted regardless of whether an API is called by an authorized integration application one or more times within one minute.
Professional	6000	
Enterprise	8000	
Platinum	10,000	

## MQS Specifications

ROMA Connect is compatible with open-source Kafka 1.1.0 and 2.3.0. Kafka 2.3.0 is recommended. The following table lists the MQS specifications supported by a ROMA Connect instance. When selecting the specifications, reserve 30% of the bandwidth to ensure stable running of your applications.

**Table 1-9** MQS Specifications

Instance Edition	Band width	TPS (High-Throughput)	TPS (Synchronous Replication)	Maximum Number of Partitions	Specifications
Basic	100 MB/s	100,000	60,000	300	Recommended for up to 3000 client connections, 60 consumer groups, and 70 MB/s service traffic.
Professional	300 MB/s	300,000	150,000	900	Recommended for up to 10,000 client connections, 300 consumer groups, and 210 MB/s service traffic.
Enterprise	600 MB/s	600,000	300,000	1800	Recommended for up to 20,000 client connections, 600 consumer groups, and 420 MB/s service traffic.
Platinum	1200 MB/s	1.2 million	400,000	1800	Recommended for up to 20,000 client connections, 600 consumer groups, and 840 MB/s service traffic.

## LINK Specifications

ROMA Connect supports device access using MQTT 3.1 and MQTT 3.1.1. The following table lists the LINK specifications supported by an instance.

**Table 1-10** LINK specifications

Instance Edition	Upstream Message	Downstream Message
Basic	10,000 TPS for 20,000 online devices	1,000 TPS for 20,000 online devices
Professional	20,000 TPS for 40,000 online devices	1,500 TPS for 40,000 online devices
Enterprise	40,000 TPS for 100,000 online devices	2,000 TPS for 100,000 online devices
Platinum	100,000 TPS for 500,000 online devices	5,000 TPS for 500,000 online devices

## 1.4 Resource Quotas

### Quota Limits

A quota refers to the maximum number of resources that you can create in a ROMA Connect instance. The following table provides the resource quota limitations.

**Table 1-11** Resource quota limitations

Component	Resource	Maximum Quota	Modifiable
Integration application	Number of integration applications	100	No
Data source	Number of data sources	500	No
APIC	Number of APIs	5000	Yes
	Number of API groups	1500	Yes
	Number of APIs in a single API group		Yes
	Number of environment variables in a single API group	50	Yes

Component	Resource	Maximum Quota	Modifiable
	Number of request throttling requests		Yes
	Number of access control policies		Yes
	Number of environments	10	Yes
	Number of signature keys	200	Yes
	Number of load balance channels	200	Yes
	Number of ECSs in a load balance channel	10	Yes
	Number of custom authorizers	50	Yes
	Number of backend APIs	500	Yes
	Number of client quota policies	20000	Yes
LINK	Number of product templates	100	Yes
	Number of products	500	Yes
	Number of devices	1000	Yes
	Number of rules	2000	No

## 1.5 Constraints

### FDI

**Table 1-12** Constraints

Function	Constraint
Data synchronization (covering all data source types)	<ul style="list-style-type: none"> <li>• A single data record cannot exceed 10 MB.</li> <li>• The fields in time format support down-to-second precision.</li> </ul>
FTP data source	When the parsing mode is enabled, each file cannot exceed 200 MB and can contain a maximum of 1,500,000 records. When the parsing mode is disabled, a maximum of 20,000 files are supported, with each file equal to or less than 6 MB.
OBS data source	If the parsing mode is enabled, the file cannot exceed 200 MB. Otherwise, the file cannot exceed 10 MB.
MRS data source	Only MRS clusters authenticated by Kerberos can be connected.
MRS Hive data source	<ul style="list-style-type: none"> <li>• Hive supports only RCFile and TEXTFILE read and write.</li> <li>• When MRS Hive serves as the source, only tables of up to 1 million records can be synchronized.</li> </ul>
API data source	<ul style="list-style-type: none"> <li>• If the server does not respond in 60 seconds, an error is reported during task execution.</li> <li>• When the API serves as the source, every request can read up to 20 MB data. Otherwise, paging must be enabled.</li> <li>• When the API serves as the source, only constant parameters are supported. Dynamic parameter transfer is not supported.</li> <li>• When the API serves as the destination, the source data cannot be mapped to the destination headers.</li> </ul>
Kafka data source	SASL can be used only to connect ROMA Connect MQS. The Kafka service developed by users themselves supports only plaintext connection.
Oracle database	Only the following field types are supported: CHAR, VARCHAR, DATE, NUMBER, FLOAT, LONG, NCHAR, NVARCHAR2, RAW, TIMESTAMP
SQL Server database	Only the following field types are supported: BIT, CHAR, DATE, DATETIME, DECIMAL, FLOAT, IMAGE, INT, MONEY, NUMERIC



Function	Constraint
MySQL database	Only the following field types are supported: INT, BIGINT, TINYINT, MEDIUMINT, FLOAT, DOUBLE, DECIMAL, CHAR, VARCHAR, TINYTEXT, TEXT, MEDIUMTEXT, LONGTEXT, DATETIME, TIMESTAMP, SMALLINT, YEAR, BINARY, BITDECIMAL, JSON, DOUBLE
PostgreSQL/DWS database	Only the following field types are supported: BOOL, CIDR, CIRCLE, DATE, NUMERIC, FLOAT4, FLOAT8, MONEY, PATH, POINT, INT, TIMESTAMP, TIMETZ, UUID, VARBIT, VARCHAR
Relational database	A maximum of 10 million data records can be synchronized.
Composite task (CDC)	<ul style="list-style-type: none"> <li>• The source supports only MySQL, SQL Server, Oracle, and GaussDB(for MySQL).</li> <li>• The destination supports only MySQL, SQL Server, Oracle, PostgreSQL, Kafka, and GaussDB(for MySQL).</li> <li>• The destination table must have a primary key. Otherwise, data synchronization will be affected.</li> </ul>

## 1.6 Basic Concepts

### Connector

A connector is a custom data source plug-in. ROMA Connect supports common data source types, such as relational databases, big data storage, semi-structured storage, and message systems. If the data source types supported by ROMA Connect cannot meet your data integration requirements, you can develop a read/write plug-in to connect to ROMA Connect through a standard RESTful API to enable ROMA Connect to read and write these data sources.

### Environment

An environment refers to the usage scope of an API. You can call an API only after you publish it in an environment. You can publish APIs in different custom environments, such as the development environment and test environment. RELEASE is the default environment for formal publishing.

### Environment Variable

Environment variables are specific to environments. You can create environment variables in different environments to call different backend services by using the same API.

## Load Balance Channel

A load balance channel allows ROMA Connect to access ECSs in the same VPC and use the backend services deployed on the ECSs to expose APIs. In addition, the load balance channel can balance access requests sent to backend services.

## Producer

A producer is a party that publishes messages into topics. The messages will be then delivered to other systems for processing.

## Consumer

A consumer is a party that subscribes to messages from topics. The ultimate purpose of subscribing to messages is to process the message content. For example, in a log integration scenario, the alarm monitoring platform functions as a consumer to subscribe to log messages from topics, identify alarm logs, and send alarm messages or emails.

## Partition

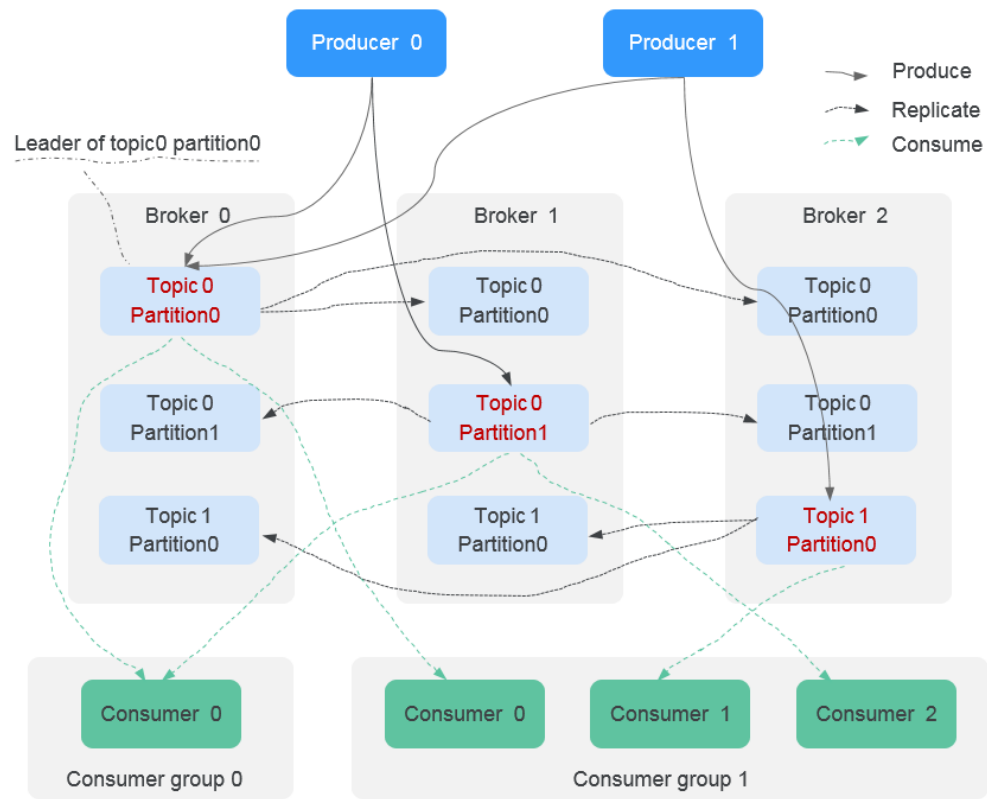
A topic is a place holder of your messages in Kafka and is further divided into partitions. Messages are stored in different partitions in a distributed manner, implementing horizontal expansion and high availability of Kafka.

## Replica

To improve message reliability, each partition of Kafka has multiple replicas to back up messages. Each replica stores all data of a partition and synchronizes messages with other replicas. A partition has one replica as the leader which handles the creation and retrieval of all messages. The rest replicas are followers which replicate the leader.

The topic is a logical concept, whereas the partition and broker are physical concepts. The following figure shows the relationship between partitions, brokers, and topics of Kafka based on the message production and consumption directions.

Figure 1-5 Kafka message flow



## Topic

A topic is a model for publishing and subscribing to messages in a message queue. Messages are produced, consumed, and managed based on topics. A producer publishes a message to a topic. Multiple consumers subscribe to the topic. The producer does not have a direct relationship with the consumers.

## Product

A product is a collection of devices with the same capabilities or features. Each device belongs to a product. You can define a product to determine the functions and attributes of a device.

## Thing Model

A thing model defines the service capabilities of a device, that is, what the device can do and what information the device can provide for external systems. After the capabilities of a device are divided into multiple thing model services, define the attributes, commands, and command fields of each thing model service.

## Device Shadow

A device shadow is a JSON file that stores the latest attribute values reported by a device and attribute values desired by applications from a device. Each device has a unique device shadow on the cloud. Applications can obtain and set device attributes through the device shadow.

## Rule Engine

A rule engine allows you to configure forwarding rules so that data reported by devices can be forwarded to other cloud services for storage or further analysis.

## 1.7 Related Services

### Virtual Private Cloud (VPC)

VPC allows you to create private, isolated virtual networks on a cloud platform. You can configure the CIDR block, subnets, and security groups, assign EIPs, and allocate bandwidth for a VPC.

ROMA Connect runs in a VPC and uses the VPC to manage IP addresses and bandwidth. When you create a ROMA Connect instance, you need to associate it with a VPC, subnet, and security group. To enable public network access for the instance, bind an EIP to the instance.

### Data Ingestion Service (DIS)

DIS builds data intake streams for custom applications capable of processing or analyzing streaming data. It provides efficient collection, transmission, and distribution of real-time data from the IoT and the Internet.

ROMA Connect uses DIS as the source and destination of data integration tasks.

### GaussDB(DWS)

GaussDB(DWS) is an online data processing database that provides scalable, fully-managed, and out-of-the-box analytic database service. It is O&M-free and supports online scale-out and multi-source data loading. It is fully compatible with the standard SQL, as well as the PostgreSQL and Oracle ecosystems.

ROMA Connect uses GaussDB(DWS) as the source and destination of data integration tasks.

### Distributed Message Service for Kafka

Distributed Message Service for Kafka (Kafka for short) is a message queuing service based on Apache Kafka. Apache Kafka is a distributed message middleware that features high throughput, data persistence, horizontal scalability, and stream data processing.

ROMA Connect uses Kafka as the source and destination of data integration tasks.

### MapReduce Service (MRS)

MRS is a cloud service that manages the Hadoop system. It provides enterprise-grade big data clusters on the cloud. Tenants can fully control clusters and easily run big data components such as Hadoop, Spark, HBase, Kafka, and Storm.

ROMA Connect uses MRS Hive, MRS HDFS, MRS HBase, or Kafka as the source and destination of data integration tasks.

## **Object Storage Service (OBS)**

OBS is an object-based cloud storage service that provides massive, secure, highly reliable, and cost-effective data storage capabilities for users to store data of any type and size.

ROMA Connect uses OBS as the source and destination of a data integration task. It can also store the data that fails to be converted to OBS during the running of data integration tasks.

## **Distributed Cache Service (DCS)**

DCS is an online, distributed, in-memory cache service. It is reliable, scalable, usable out of the box, and easy to manage. Compatible with Redis, DCS can meet your requirements for high read/write performance and fast data access.

ROMA Connect uses Redis as the destination of data integration tasks or encapsulates Redis into APIs and exposes them to external systems.

# 2 Getting Started

[Overview](#)

[Using ROMA Connect for Data Integration](#)

[Using ROMA Connect for Service Integration](#)

[Using ROMA Connect for Message Integration](#)

[Using ROMA Connect for Device Integration](#)

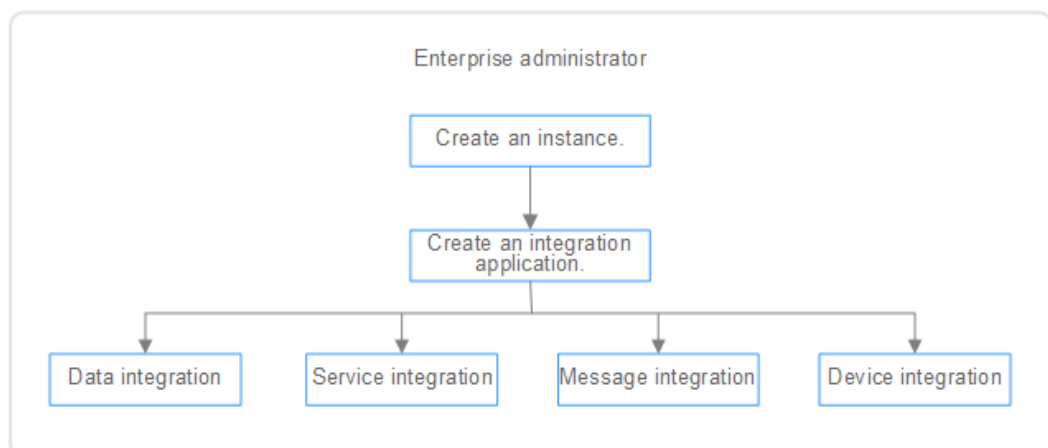
## 2.1 Overview

### Process for Using ROMA Connect

ROMA Connect is a full-stack integration platform that connects applications and data to provide data, API, message, and device integration capabilities. It applies to common enterprise system integration scenarios.

The following figure shows the process for using ROMA Connect.

**Figure 2-1** Process for using ROMA Connect



1. **Create an instance.**  
On the ROMA Connect console, create a ROMA instance and select an instance edition based on service requirements.
2. **Create an integration application.**  
Under the instance, create an integration application. All resources in an instance must belong to an integration application.
3. Select ROMA Connect functions based on service scenarios.
  - **Data integration:** Connects data sources at the source and destination. A data task can be used to integrate data from the source to the destination.
  - **Service integration:** Encapsulates existing backend services, data sources, and custom functions into RESTful APIs and exposes them to external systems.
  - **Message integration:** Creates message topics through which different systems interconnect with each other to send and receive messages.
  - **Device integration:** Defines device models and registers devices in the cloud. Devices are connected to the cloud through SDKs to send and receive messages.

## 2.2 Using ROMA Connect for Data Integration

### Overview

ROMA Connect supports access of multiple types of data sources and uses data integration tasks to integrate data from source to destination. Both homogeneous and heterogeneous data can be integrated.

This chapter provides an example for configuring data integration from API to MQS to help you get familiar with the data integration process.

The steps for integrating data with ROMA Connect are as follows:

- **Step 1: Prepare Data Sources**
- **Step 2: Connect to Data Sources**
- **Step 3: Create a Data Integration Task**
- **Step 4: Start the Data Integration Task**
- **Step 5: View the Data Integration Result**

### Step 1: Prepare Data Sources

Before using ROMA Connect to integrate data, you need to prepare one API data source and one MQS data source as the source and destination, respectively.

1. Create a function API.  
Create a function API on the current ROMA Connect instance as API data source at the source. For details, see [Preparing a Backend Service](#).  
For simplicity, the function API does not require request authentication or carry request parameters. When the function API is called, the message "ROMA TEST!" is always returned.

After the function API is created, record the request URL and method of the API.

2. Create a message topic.

Create a message topic on the current ROMA Connect instance as the topic of the MQS data source at the destination. For details, see [Creating a Message Topic](#).

After the topic is created, record the topic name. On the **Application Integrations** page of the ROMA Connect console, click the name of the integration application to which the topic belongs to view and obtain the key and secret.

## Step 2: Connect to Data Sources

Before creating a data integration task, separately connect the API and MQS data sources to ROMA Connect to ensure that data can be read from the source and written to the destination.

1. Create an integration application.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
  - b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
  - c. In the dialog box displayed, set **Name** and click **OK**.
2. Connect to the API data source.
  - a. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
  - b. On the **Default** tab page, select **API** and click **Next**.
  - c. Enter the configuration information about the API data source.

**Table 2-1** API data source configuration

Parameter	Description
Name	Enter an API data source name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in <a href="#">1</a> .
Description	Enter a brief description of the API data source.
Address	Enter the API request URL recorded in <a href="#">Step 1: Prepare Data Sources</a> .
Request Method	Select the API request method recorded in <a href="#">Step 1: Prepare Data Sources</a> .
Authentication Mode	Select <b>None</b> , indicating that no authentication is required for the API data source.



- d. After setting the parameters for the API data source, click **Check Connectivity** to test the data source connection.
    - If the test result is **Data source connected successfully**, click **Create**.
    - If the test result is **Failed to connect to the data source**, check and modify the connection parameters, and click **Recheck** until the connection is successful.
  - e. Click **Create**.
3. Connect to the MQS data source.
    - a. On the **Data Sources** page, click **Access Data Source** in the upper right corner.
    - b. On the **Default** tab page, select **MQS** and click **Next**.
    - c. Enter the configuration information about the MQS data source.

**Table 2-2** MQS data source configuration

Parameter	Description
Name	Enter an MQS data source name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in <a href="#">1</a> .
Description	Enter a brief description of the MQS data source.
Connection Address	Select the MQS intranet address of the current ROMA Connect instance. You can view the MQS intranet address on the <b>Instance Information</b> page of the ROMA Connect console.
Enable SSL	Set this parameter to <b>Yes</b> if <b>MQS SASL_SSL</b> is enabled and <b>Intra-VPC Plaintext Access</b> is disabled for the ROMA Connect instance. In other cases, set this parameter to <b>No</b> .
SSL Username/ Application Key	This parameter is mandatory only if <b>Enable SSL</b> is set to <b>Yes</b> . Set this parameter to the key recorded in <a href="#">Step 1: Prepare Data Sources</a> .
SSL Password/ Application Secret	This parameter is mandatory only if <b>Enable SSL</b> is set to <b>Yes</b> . Set this parameter to the secret recorded in <a href="#">Step 1: Prepare Data Sources</a> .

- d. After setting the parameters for the MQS data source, click **Check Connectivity** to test the data source connection.
  - If the test result is **Data source connected successfully**, click **Create**.

- If the test result is **Failed to connect to the data source**, check and modify the connection parameters, and click **Recheck** until the connection is successful.
- e. Click **Create**.

### Step 3: Create a Data Integration Task

After configuring the API and MQS data sources, create a data integration task from the source to the destination.

1. In the navigation pane on the left, choose **Fast Data Integration > Task Management** and click **Create Common Task**.
2. On the **Create Task** page, set the parameters of the data integration task.
  - a. Configure basic information about the task.

**Table 2-3** Basic task information

Parameter	Description
Task Name	Enter a task name. It is recommended that you enter a name based on naming rules to facilitate search.
Description	Enter a brief description of the task.
Integration Mode	Select <b>Scheduled</b> as the mode used for data integration. When API is used as the data source at the source, only scheduled tasks are supported.
Tag	Add a tag to classify tasks for quick search. You do not need to set this parameter.

- b. Configure a schedule for the task.  
For simplicity, retain the default settings for the task schedule.
  - c. Configure the API data source at the source.

**Figure 2-2** Configuring source information

The screenshot shows the 'Source Information' configuration page. It contains several fields with red asterisks indicating required fields:

- Instance:** A dropdown menu with 'romafdi\_roma-lz5a' selected.
- Integration Application:** A dropdown menu with 'app-doc' selected.
- Data Source Type:** A dropdown menu with 'API' selected.
- Data Source Name:** A dropdown menu with 'API-doc' selected. To the right is a 'Create Data Source' link.
- Paging:** A toggle switch that is turned off. Text next to it says 'If Paging is disabled, ROMA Connect obtains all data that meets'.
- Incremental Migration:** A toggle switch that is turned off.
- Request Parameters:** A section with three tabs: 'Params' (selected), 'Headers', and 'Body'. Below the tabs is a table with one row containing the word 'Key'. Below the table is an 'Add' button with a plus sign.
- Parse:** A dropdown menu with 'No' selected and a help icon.

**Table 2-4** Configuration information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application created in <a href="#">Step 2: Connect to Data Sources</a> .
Source Data Type	Select <b>API</b> .
Data Source Name	Select the API data source that you configured in <a href="#">Step 2: Connect to Data Sources</a> .
Paging	You do not need to enable this option.
Incremental Migration	You do not need to enable this option.
Request Parameters	The API data source does not have request parameters. Therefore, you do not need to set this parameter.
Parse	Set this parameter to <b>No</b> .

- d. Configure the MQS data source at the destination.

**Figure 2-3** Configuring destination information

### Destination Information

\* Instance

\* Integration Application

\* Data Source Type

\* Data Source Name  [Create Data Source](#)

\* Topic Name

Key

**Table 2-5** Configuration information at the destination

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application created in <a href="#">Step 2: Connect to Data Sources</a> .
Data Source Type	Select <b>MQS</b> .
Data Source Name	Select the MQS data source that you configured in <a href="#">Step 2: Connect to Data Sources</a> .
Topic Name	Select the topic recorded in <a href="#">Step 1: Prepare Data Sources</a> .
Key	You do not need to set this parameter.

- e. Retain the default settings for **Post-Integration Operation**.
3. Click **Create**, and then click **Back to Task List**. The created data integration task is displayed in the task list.

### Step 4: Start the Data Integration Task

After a data integration task is created, **Task Status** will be displayed as **Stopped**.

You can start the task by using either of the following methods:

- **Creating schedule:** The task is periodically executed according to the schedule created during task creation.  
Select the task you want to start, click **Start** above the task list to start the task schedule, and wait until the task is executed as scheduled.

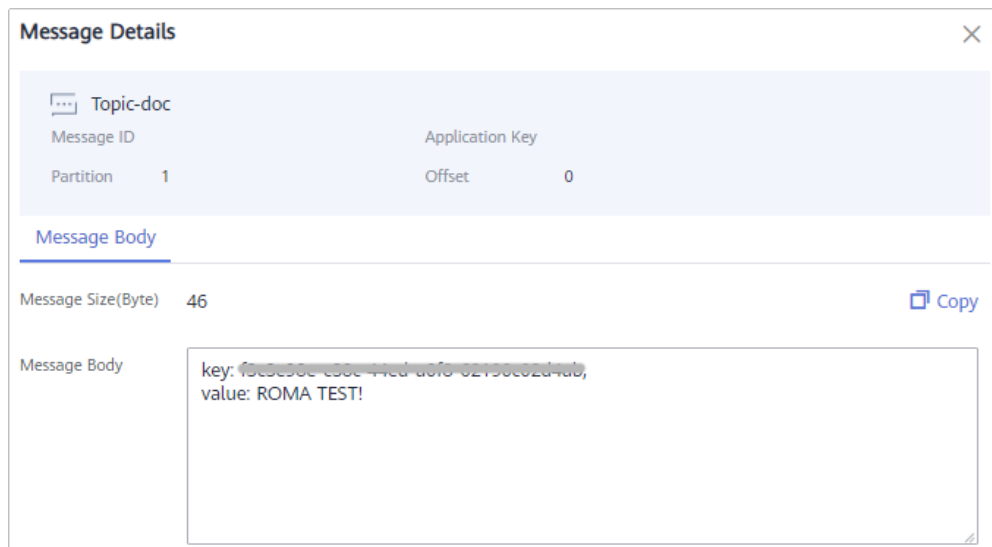
- **Manual scheduling:** The task is manually executed only once. Before manually executing a task, ensure that **Task Status** is **Stopped**.  
Click **Start Scheduling** on the right of the data integration task you want to start. When **Running Status** of the task is displayed as **Successful**, the task is successfully executed.

To execute the data integration task immediately and view the task result, select the manual scheduling method to integrate data from the API data source to the MQS data source.

### Step 5: View the Data Integration Result

1. In the navigation pane on the left, choose **Message Queue Service > Message Query**.
2. Click **Advanced Search** to expand advanced search options.
3. Enter the search criteria and click **Search** to query the messages sent by the client.
  - Select the message topic created in **Step 1: Prepare Data Sources** for **Topic Name**.
  - Select **Creation time** for **Search By** and select the time segment for the client to send messages to ROMA Connect.
4. Click **Message Body** next to the message record and check whether the message content is the same as the response message of the API.

Figure 2-4 Viewing messages



## 2.3 Using ROMA Connect for Service Integration

### Overview

ROMA Connect can encapsulate APIs, data sources, and custom functions into standard RESTful APIs and expose them to external systems. ROMA Connect provides complete lifecycle management and multi-layer security protection for APIs.

This chapter provides a configuration example to help you get familiar with the service integration process.

The steps for integrating services with ROMA Connect are as follows:

- [Step 1: Prepare a Backend Service](#)
- [Step 2: Create an API Group](#)
- [Step 3: Create an API](#)
- [Step 4: Debug the API](#)
- [Step 5: Publish the API](#)
- [Step 6: Call the API](#)

## Prerequisites

Before performing this operation, complete the following preparations:

- An EIP has been bound to the ROMA Connect instance, and the network where the local PC is located can communicate with the EIP network.
- The [Postman](#) tool has been downloaded and installed on your local PC.

## Step 1: Prepare a Backend Service

You need to prepare an API that can be called as a backend service. In this example, you can customize a function API that does not require any authentication or contain any request parameters as the backend service.

1. Create an integration application.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
  - b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
  - c. In the dialog box displayed, set **Name** and **Description**, and click **OK**.
2. Create a function API.
  - a. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Create**.
  - b. On the **Create Backend** page, set backend parameters and click **Create**.

**Figure 2-5** Creating a backend

The screenshot shows a web form for creating a backend. It contains the following elements:

- Name:** A text input field containing "backend-test". Below it is a tooltip: "Enter a string of 3 to 100 characters starting with a letter. Only letters, digits, hyphens".
- Integration Application:** A dropdown menu showing "app-doc" and a "Create Integration Application" button.
- Backend Request Method:** A dropdown menu showing "GET".
- Backend Request Path:** A text input field with a placeholder: "Enter a request path, for example, /getUserInfo/userId." Below it is a tooltip: "Enter a path starting with a slash (/). Only letters, digits, hyphens (-), underscores (\_),".
- Backend Security Authentication:** Two radio buttons: "Signature key" and "None". The "None" button is selected. Below it is a red warning message: "No security authentication. Access will be granted to all users. (Not recommended)".
- Description:** A large text area with a placeholder: "Enter a backend description." and a character count "0/1,000".
- Advanced Settings:** A link with a downward arrow.

**Table 2-6** Backend configuration

Parameter	Description
Name	Enter a backend name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in <b>1</b> .
Backend Request Method	Select <b>GET</b> as the request method of the backend.
Backend Request Path	Enter the request path of the backend in the <i>/serviceName/interfaceName</i> format.
Backend Security Authentication	Select <b>None</b> , which indicates that authentication is not required when the API is called.
Description	Enter a brief description of the backend.
Advanced Settings	You do not need to configure advanced settings.

After the backend is created, the online IDE page is automatically displayed.

- c. Use a reference script provided by ROMA Connect to implement a simple function backend.  
In the upper left corner of the online IDE page, choose **File > New Function Backend > Hello World Example**. In the dialog box displayed, click **Yes**. The built-in script **Hello world!** is displayed in the editing box.
- d. Change **Hello world!** to **ROMA TEST!** so that the response **ROMA TEST!** will be returned when the backend is called. Then, click **Save** in the upper right corner of the page.  

```
function execute(data){
  return "ROMA TEST!"
}
```
- e. Click **Test** in the upper right corner of the page, and then click **Test** in the test bar.  
In the **Execution Result** area, check whether the response is **ROMA TEST!** defined in 2.d.
- f. After the backend test is complete, click **Deploy** in the upper right corner of the page. In the dialog box displayed, click **Yes** to deploy the function backend.
- g. After the backend deployment is complete, click **Publish** in the upper right corner of the page.
- h. In the pane displayed, set required parameters and click **Publish** to create a frontend function API for the backend and publish the API in the environment.

**Table 2-7** Parameters for publishing a backend

Parameter	Description
Group	Select the API group to which the frontend API belongs. If there is a default API group, select the default API group. If there is no default API group, click <b>Create API Group</b> . In the <b>Create API Group</b> dialog box, click <b>OK</b> to create an API group.
Environment	Select the default environment <b>RELEASE</b> .
Frontend Security Authentication	Select the security authentication mode used by the frontend API. In this example, select <b>None</b> .
Frontend Request Protocol	Enter the request protocol used to access the backend API. In this example, retain the default option <b>HTTPS</b> .
Timeout (ms)	Timeout interval for the frontend API to call a backend service. Retain the default value <b>60000</b> .
Advanced Settings	You do not need to configure advanced settings.

- i. After the function API is published, the API details page is displayed. In the **Request Definition** area on the **API Calling** tab page, obtain and



record the values of **Subdomain Name**, **Protocol**, **Method**, and **Path** of the function API.

## Step 2: Create an API Group

Each open API belongs to an API group. Before creating an API, create an API group.

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **API Groups** tab page, click **Create**.
3. In the **Create API Group** dialog box, enter the group-related parameters and click **OK**.

**Figure 2-6** Creating an API group

**Table 2-8** Parameters for creating an API group

Parameter	Description
Name	Enter an API group name. It is recommended that you enter a name based on naming rules to facilitate search.
Type	Select <b>Integration Application</b> as the API group type.
Integration Application	Select the integration application created in <b>Step 1: Prepare a Backend Service</b> . If no integration application is available, click <b>Create Integration Application</b> on the right to create an integration application.
Description	Enter a brief description of the API group.

### Step 3: Create an API

Create an API on ROMA Connect and associate it with the backend service.

1. On the **APIs** tab page, click **Create API**.
2. On the **Create API** page, set the parameters of the API.
  - a. Configure basic information and click **Next**.

**Table 2-9** Basic API information

Parameter	Description
Name	Enter an API name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in <b>Step 1: Prepare a Backend Service</b> . If no integration application is available, click <b>Create Integration Application</b> on the right to create an integration application.
API Group	Select the API group created in <b>Step 2: Create an API Group</b> .
Security Authentication	Security authentication mode used for the API. In this example, select <b>None</b> .
Two-factor Authentication	You do not need to enable this option.
Tag Name	You do not need to set this parameter.
Description	You do not need to enter a description.

- a.
  - b. Define API request information, which will be exposed to external users. Then, click **Next**.

**Figure 2-7** Defining API request information

The screenshot shows a configuration form titled "Define API Request". It contains the following elements:

- Domain Name:** bcf81609929c417aae926c64f322df0a.apic.cn-north-1.huaweicloudapis.com
- Protocol:** Three buttons: HTTP, **HTTPS** (selected), and HTTP&HTTPS. Below it, text says "WebSocket is supported for HTTP and HTTPS."
- \* Path:** A text input field with the placeholder "Enter an API request path." Below it, text says "Enclose parameters in braces, for example, /getUserInfo/{userid}. The following s".
- Matching:** Two buttons: **Exact match** (selected) and Prefix match. Below it, text says "API requests will be forwarded to the specified path."
- \* Method:** A dropdown menu showing "GET".
- CORS:** A toggle switch that is currently turned off. Below it is a link: "Learn more about CORS".
- Input Parameters:** A dropdown menu with a downward arrow.

**Table 2-10** API request configuration

Parameter	Description
Protocol	Select <b>HTTPS</b> as the request protocol used to access the API.
Path	Enter the request path of the API in the <code>/serviceName/interfaceName</code> format.
Matching	Select <b>Exact match</b> as the matching mode of the request path.
Method	Select <b>GET</b> as the request method of the API.
CORS	Specifies whether CORS is supported. You do not need to enable this option.
Input Parameters	Request parameters of the API. You do not need to add request parameters.

- c. Configure backend service information of the API, that is, the access information of the API functioning as the backend service.

If you have prepared a function API by referring to [Step 1: Prepare a Backend Service](#), configure the access information of that API.

Then, click **Next**.

**Figure 2-8** Configuring the access to the backend service

**Basic Information**

Protocol

Method

Load Balance Channel Do not configure Configure now

A load balance channel enables a backend to directly

\* Backend Address

The IP address of the backend to which the APIC will

\* Path

Enclose parameters in braces, for example, /getUserIn

\* Timeout (ms)

Backend Authentication

Specify a custom authorizer for backend access.

---

Backend Parameters (?) v

Constant Parameters (?) v

System Parameters v

**Table 2-11** Backend service access parameters

Parameter	Description
Backend Type	Select <b>HTTP/HTTPS</b> as a backend service.
Protocol	Select the request protocol of the function API recorded in <a href="#">Step 1: Prepare a Backend Service</a> .
Method	Select the request method of the function API recorded in <a href="#">Step 1: Prepare a Backend Service</a> .
Load Balance Channel	Select <b>Do not configure</b> .

Parameter	Description
Backend Address	Enter the access address of the backend service in the <i>Host:Port</i> format. <i>Host</i> indicates the IP address or domain name for accessing the backend service. If no port is specified, port 80 is used for HTTP by default, and port 443 is used for HTTPS by default. Enter the subdomain name of the function API recorded in <a href="#">Step 1: Prepare a Backend Service</a> .
Path	Enter the request path of the backend service in the <i>/serviceName/interfaceName</i> format. Enter the request path of the function API recorded in <a href="#">Step 1: Prepare a Backend Service</a> .
Timeout (ms)	Retain the default value <b>5000</b> as the timeout interval for ROMA Connect to call the backend service.
Two-way Authentication	Determine whether to enable two-way authentication between ROMA Connect and backend services. You do not need to enable this option.
Backend Authentication	Determine whether the backend service authenticates the calling requests sent by ROMA Connect. You do not need to enable this option.
Backend Parameters	In this example, the backend service does not contain request parameters. You do not need to set this parameter.
Constant Parameters	In this example, the backend service does not contain request parameters. You do not need to set constant parameters.
System Parameters	In this example, the backend service does not contain request parameters. You do not need to set system parameters.

- d. Configure the response example returned by the API, which will be displayed in the API calling information. After the configuration is complete, click **Finish**.

**Figure 2-9** Configuring the API response examples

**Table 2-12** Parameters for configuring API response examples

Parameter	Description
Example Success Response	Set the success response example based on the response of the backend service. In this example, set this parameter to <b>ROMA TEST!</b> .
Example Failure Response	Example of a failure response returned by the system when an API fails to be called. You do not need to set this parameter.

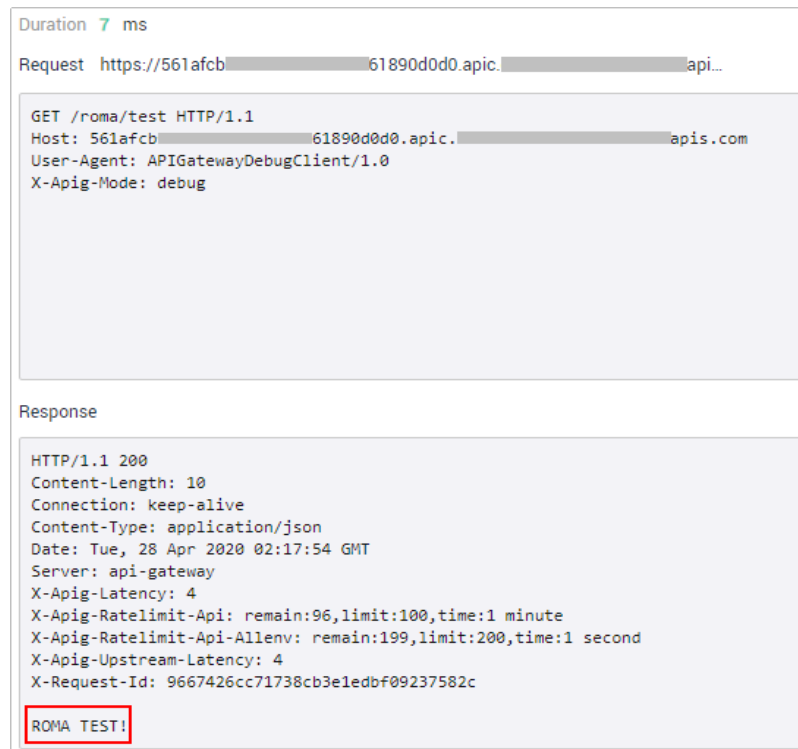
3. After the API is created, click **Back to API List**. The created API is displayed in the API list.

## Step 4: Debug the API

After you create an API, debug it to ensure that the API functions properly.

1. On the **APIs** tab page, choose **More > Debug** in the **Operation** column of the API.
2. Click **Send Request** on the API debugging page to test the API calling.
3. On the right of the page, view the request and response sent by the API.  
If the service is successfully called, a success response body is returned. In this example, **ROMA TEST!** is returned.

**Figure 2-10** Debugging an API

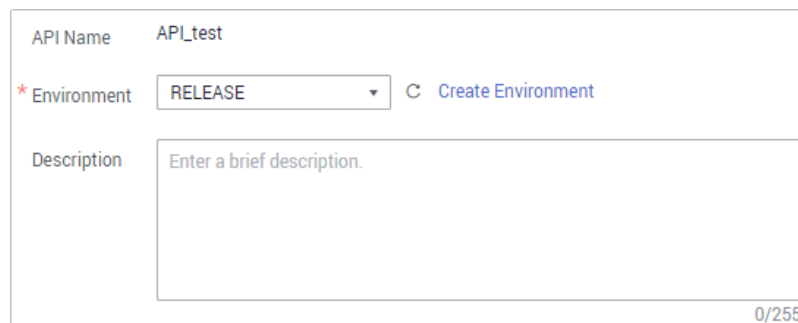


## Step 5: Publish the API

After the API function test is successful, publish the API to the environment. An API can be called by external users only after it is published to the environment.

1. On the **APIs** tab page, click **Publish** in the **Operation** column of the API.
2. On the **Publish API** page, set **Environment** to **RELEASE** and click **Publish**.

**Figure 2-11** Publishing the API



3. Click **Back to API List**. In the API list, the running environment of the API changes to **RELEASE**.

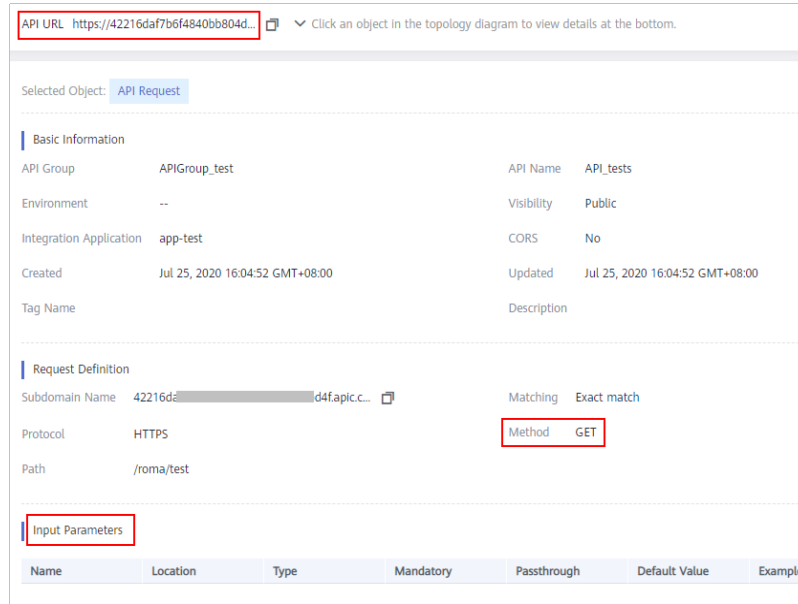
## Step 6: Call the API

In this example, Postman will be used to call APIs.

1. Obtain request information of the API.

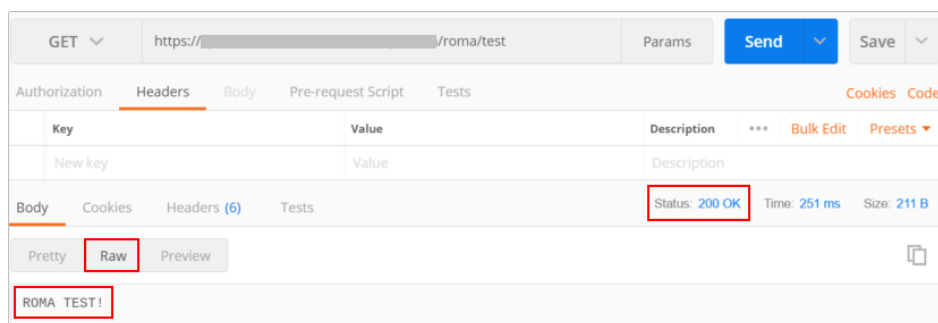
- a. On the **APIs** tab page, click the API name to access the API details page.
- b. On the **API Calling** tab page, select **API Request** in the topology. View the API calling information, including the URL, subdomain name, request method, request path, and request parameters, in the lower part of the page.

**Figure 2-12** API request information



2. Call the exposed API.  
Use Postman to call the API obtained in 1.  
After the API is successfully called, check whether the response is the same as the test result in **Step 4: Debug the API**.

**Figure 2-13** API calling response



## 2.4 Using ROMA Connect for Message Integration

### Overview

ROMA Connect provides secure and standard message channels to implement integration with different message systems.



This chapter provides a configuration example of interworking with ROMA Connect using Kafka command-line tool to help you get familiar with the message integration process.

The steps for integrating messages with ROMA Connect are as follows:

- [Step 1: Create a Message Topic](#)
- [Step 2: Send and Receive Messages to and from a Topic](#)

## Prerequisites

Before performing this operation, complete the following preparations:

- An EIP has been bound to the ROMA Connect instance.
- The [Java JDK](#) has been downloaded and installed on your local PC, and related environment variables have been configured.
- The open-source Kafka command-line tool of the corresponding version has been downloaded based on the Kafka version of the ROMA Connect instance. You can view the Kafka version in the **MQS Information** area on the **Instance Information** page of the ROMA Connect console.
  - [Kafka command-line tool 1.1.0](#)
  - [Kafka command-line tool 2.3.0](#)

### NOTE

The Kafka server version of ROMA Connect can be 1.1.0 or 2.3.0. Therefore, you must use the Kafka command-line tool of the same version as the server to avoid unpredictable problems.

- If **MQS SASL\_SSL** has been enabled for the ROMA Connect instance, ensure that the has been downloaded. To download the client certificate, log in to the ROMA Connect console, choose **Message Queue Service > Topic Management**, and click **Download SSL Certificate**.

## Step 1: Create a Message Topic

A topic is a channel for message transmission between the message client and ROMA Connect. The client sends messages to and receives messages from ROMA Connect through the topic.

1. Create an integration application.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
  - b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
  - c. In the dialog box displayed, set **Name** and **Description**, and click **OK**.
2. Create a message topic.
  - a. In the navigation pane on the left, choose **Message Queue Service > Topic Management**. On the page displayed, click **Create Topic** in the upper right corner.
  - b. In the **Create Topic** dialog box, set topic parameters and click **OK**.

**Figure 2-14** Creating a topic

The screenshot shows a dialog box for creating a topic. It includes the following fields and controls:

- Topic Name:** A text input field with the placeholder "Enter a topic name."
- Integration Application:** A dropdown menu.
- Permission:** Three buttons: "Publish/Subscribe" (highlighted in blue), "Publish", and "Subscribe".
- Partitions:** A numeric spinner set to 3, with a value range of 1 to 50.
- Replicas:** A numeric spinner set to 3, with a value range of 1 to 3.
- Aging Time (h):** A numeric spinner set to 72, with a value range of 1 to 168.
- Synchronous Replication:** A toggle switch that is currently turned off. Below it is a note: "The acks parameter must be set to -1 on the producer client so that multiple replicas can be synchronously replicated."
- Synchronous Flushing:** A toggle switch that is currently turned off.
- Tag:** A button with a plus icon and the text "Add".
- Sensitive Fields:** A button with a plus icon and the text "Add".
- Description:** A button with a pencil icon.
- Buttons:** "OK" (red) and "Cancel" (grey) buttons at the bottom right.

**Table 2-13** Topic parameters

Parameter	Description
Topic Name	Enter a topic name. It is recommended that you enter a name based on naming rules to facilitate search.
Integration Application	Select the integration application created in <a href="#">1</a> .
Permission	Select the topic permission for the integration application to which the topic belongs. In this example, select <b>Publish/Subscribe</b> , which indicates that the topic can be used to produce and consume messages.
Partitions	The appropriate number of partitions can improve the concurrent performance of message creation and retrieval. Retain the default value <b>3</b> .
Replicas	ROMA Connect automatically backs up data on each replica. If one replica is faulty, data will still be available. The reliability increases with the number of replicas of a topic. Retain the default value <b>3</b> .

Parameter	Description
Aging Period (h)	After the aging time expires, messages stored in the topic are deleted. Retain the default value <b>72</b> .
Synchronous Replication	When a client creates a message to a topic, ROMA Connect determines whether to replicate the message to all replicas and then returns a response to the client. You do not need to enable this option.
Synchronous Flushing	Specifies whether each message created by a client to the topic is immediately written to the disk. You do not need to enable this option.
Tag	A tag can be added for a topic to facilitate topic search. You do not need to add a tag.
Sensitive Fields	If a message sent to a topic contains sensitive fields, the sensitive fields will be shielded. You do not need to add sensitive fields.
Description	Enter a brief description of the topic.

## Step 2: Send and Receive Messages to and from a Topic

Use the Kafka command-line tool on the local PC to send messages to and receive messages from topics in CLI mode.

The operations for sending and receiving messages to and from a topic vary depending on whether SASL\_SSL is enabled for the ROMA Connect instance. If SASL\_SSL is enabled, messages sent and received by clients to and from topics are encrypted before transmission.

1. Decompress the Kafka command-line tool and client certificates.

Locate the downloaded Kafka command-line tool and client certificate files on your local PC and decompress them.

The following uses Windows as an example. Assume that the Kafka command-line tool path is **D:\kafka\_2.11-1.1.0** and the client certificate file path is **D:\cert**.

2. (Optional) Modify the **kafka-run-class.bat** file in the Kafka command-line tool. This step is required only if the Kafka command-line tool 1.1.0 is used. Otherwise, skip this step.

Find the **kafka-run-class.bat** file in the **D:\kafka\_2.11-1.1.0\bin\windows** directory and add double quotation marks (") to **%CLASSPATH%** in the following script line:

```
set COMMAND=%JAVA% %KAFKA_HEAP_OPTS% %KAFKA_JVM_PERFORMANCE_OPTS%  
%KAFKA_JMX_OPTS% %KAFKA_LOG4J_OPTS% -cp "%CLASSPATH%" %KAFKA_OPTS% %*
```

3. (Optional) Modify the configuration file of the Kafka command-line tool. This step is performed only if SASL\_SSL is enabled for the ROMA Connect instance. Otherwise, skip this step.

Locate the **consumer.properties** and **producer.properties** files in **D:\kafka\_2.11-1.1.0\config** and add the following content to the files separately:

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
username="*****" \
password="*****";
sasl.mechanism=PLAIN
security.protocol=SASL_SSL
ssl.truststore.location=D:/cert/client.truststore.jks
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
```

In the preceding information:

- The values of **username** and **password** are the key and secret of the integration application to which the topic belongs. You can click the integration application name on the **Integration Applications** page of the ROMA Connect console to view and obtain the key and secret.
- The value of **ssl.truststore.location** is the path for storing the client certificate obtained in **1**.

#### NOTICE

The certificate path in Windows must contain slashes (/).

4. On the command line window, run the following command to go to the directory where the Kafka command-line tool is stored:

```
d:
cd kafka_2.11-1.1.0\bin\windows
```

5. Produce messages to ROMA Connect.

- a. Run the following command to create a connection with the topic for producing messages:

If SASL\_SSL is not enabled for the ROMA Connect instance, run the following command:

```
kafka-console-producer.bat --broker-list IP:9094,IP:9095,IP:9096 --topic TopicName
```

If SASL\_SSL is enabled for the ROMA Connect instance, run the following command:

```
kafka-console-producer.bat --broker-list IP:9095,IP:9096,IP:9097 --topic TopicName --
producer.config ../../config/producer.properties
```

In the preceding information:

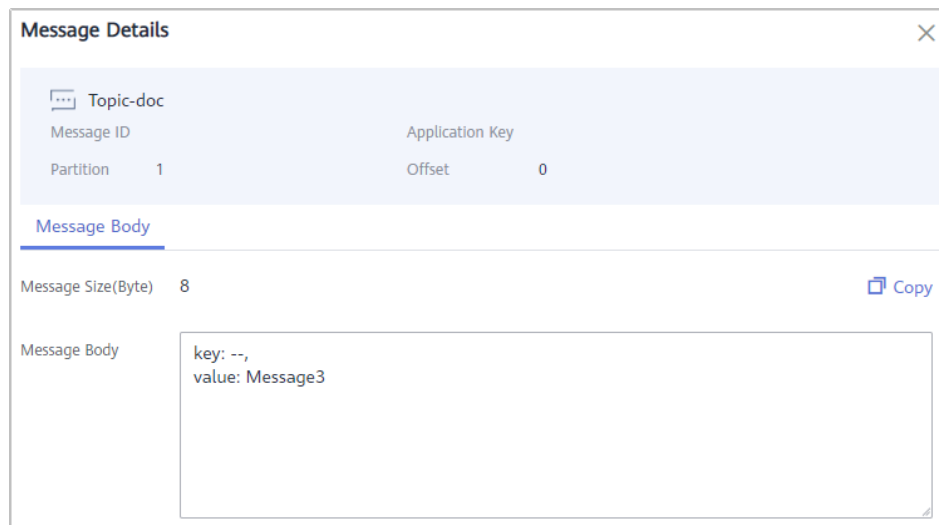
- **IP** indicates the MQS connection address of ROMA Connect. You can obtain the address by viewing **EIP** on the **Instance Information** page of the ROMA Connect console.
  - **TopicName** indicates the name of the topic created in **Step 1: Create a Message Topic**.
- b. Enter the message contents and send the messages to the topic.
 

```
>Message1
>Message2
>Message3
```

In the preceding information, **Message1**, **Message2**, and **Message3** indicate the actual message contents sent to the topic. Each row indicates a message.
  - c. On the ROMA Connect console, choose **Message Queue Service** > **Message Query**.
  - d. Click **Advanced Search** to expand advanced search options.

- e. Enter search criteria and click **Search** to query the messages sent by the client.
  - Select the message topic created in [Step 1: Create a Message Topic](#) for **Topic Name**.
  - Select **Creation time** for **Search By** and select the time segment for the client to send messages to ROMA Connect.
- f. Click **Message Body** next to the message record and check whether the message content is the same as that sent in [5.b](#).

**Figure 2-15** Querying messages



6. Consume messages from ROMA Connect.
  - a. Run the following command to create a connection with the topic for reading messages:
 

If SASL\_SSL is not enabled for the ROMA Connect instance, run the following command:

```
kafka-console-consumer.bat --bootstrap-server IP:9094,IP:9095,IP:9096 --topic TopicName --from-beginning
```

If SASL\_SSL is enabled for the ROMA Connect instance, run the following command:

```
kafka-console-consumer.bat --bootstrap-server IP:9095,IP:9096,IP:9097 --topic TopicName --from-beginning --consumer.config ../../config/consumer.properties
```

In the preceding information:

    - **IP** indicates the MQS connection address of ROMA Connect. You can obtain the address by viewing **EIP** on the **Instance Information** page of the ROMA Connect console.
    - **TopicName** indicates the name of the topic created in [Step 1: Create a Message Topic](#).
  - b. After the command is executed, the system is continuously connected to the topic and reads messages. To stop reading messages, press **Ctrl+C**, enter **Y**, and press **Enter**.

## 2.5 Using ROMA Connect for Device Integration

### Overview

ROMA Connect uses the standard Message Queue Telemetry Transport (MQTT) protocol to connect devices and implement quick and easy device management on the cloud.

This chapter provides a configuration example to help you get familiarized with the device integration process.

The steps for integrating devices with ROMA Connect are as follows:

- [Step 1: Define a Product](#)
- [Step 2: Register a Device](#)
- [Step 3: Configure Data Forwarding Rules](#)
- [Step 4: Debug the Device](#)

### Prerequisites

Before performing this operation, complete the following preparations:

- An EIP has been bound to the ROMA Connect instance.
- The [MQTT.fx client](#) has been downloaded and installed on your local PC.
- A topic has been created for message integration in the ROMA Connect instance. For details, see [Creating a Message Topic](#).

### Step 1: Define a Product

A product is a device model. You can define a product to determine the function attributes of a device. Each device belongs to a product.

1. Create an integration application.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
  - b. In the navigation pane on the left, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
  - c. In the dialog box displayed, set **Name** and **Description**, and click **OK**.
2. Create a product.
  - a. In the navigation pane on the left, choose **LINK > Product Management**. On the page displayed, click **Create Product** in the upper right corner.
  - b. In the **Create Product** dialog box, set product parameters and click **OK**.

**Table 2-14** Parameters for creating a product

Parameter	Description
Product Type	Select <b>Common</b> as the product type.

Parameter	Description
Protocol Type	Select <b>MQTT</b> as the protocol type.
Integration Application	Select the integration application created in <b>1</b> .
Product Template	You do not need to set this parameter.
Product Name	Enter a product name, which can be user-defined.
Manufacturer Name	Enter a manufacturer name of the device, which can be user-defined.
Manufacturer ID	Enter a manufacturer ID of the device, which can be user-defined.
Product Model	Enter a product model of the device, which can be user-defined.
Device Type	Select <b>Default Type</b> as the type of the access device.
Model Version	Enter a device model version, which can be user-defined.
Description	Enter a brief description of the product.

3. Add a thing model service for the product.
  - a. On the **Product Management** page, click the **Products** tab and click the name of the product created in **2** to access the product details page.
  - b. On the **Thing Model** tab page, click **Create**.
  - c. In the **Create Thing Model Service** dialog box, set thing model service parameters and click **OK**.

Assume that the device is a temperature sensor. Create a temperature service **Temperature** for the product.

**Table 2-15** Parameters for creating a thing model service

Parameter	Description
Name	Enter <b>Temperature</b> as the name of the thing model service.
Status	This parameter specifies whether to enable the thing model service. Retain the default value, that is, the thing model service is enabled.
Description	Enter a brief description of the thing model service.

4. Add an attribute for the thing model service.
  - a. In the left pane of the **Thing Model** tab page, select the thing model service added in **3**.

- b. On the **Attributes** tab page, click **Create**.
- c. In the **Create Attribute** dialog box, set attribute parameters and click **OK**.

Assume that the device reports only the attribute **temperature** whose data type is **String**.

**Table 2-16** Parameters for creating an attribute


Parameter	Description
Name	Enter <b>temperature</b> as the attribute name.
Data Type	Select <b>String</b> as the data type of the attribute parameter reported by the device.
Mandatory	This parameter specifies whether the attribute must be reported by a device. Retain the default value, that is, the attribute must be reported by a device.
Description	Enter a brief description of the attribute.
Data Length	Enter <b>32</b> as the maximum length of the attribute value.
Enumerated Value	You do not need to set this parameter.

- 5. Add a command for the thing model service.
  - a. On the **Commands** tab page, click **Create**.
  - b. In the **Create Command** dialog box, set command parameters and click **OK**.

Assume that a device has the **SET\_STATUS** command for setting the status, the command field is **status**, the data type is **String**, and the enumerated values are **on** and **off**.

**Table 2-17** Parameters for creating a command

Parameter	Description
Name	Enter <b>SET_STATUS</b> as the command name.
Description	Enter a brief description of the command.

- c. Locate the created command in the command list and click  on the left of the command name to expand the command fields.
- d. Click **Create Field**.
- e. In the **Create Delivery Command Field** dialog box, set field parameters and click **OK**.



**Table 2-18** Parameters for creating a command field

Parameter	Description
Name	Enter <b>status</b> as the command field name.
Data Type	Select <b>String</b> as the data type of the command field.
Mandatory	This parameter specifies whether the field must be carried in a delivered command. Retain the default value.
Description	Enter a brief description of the command field.
Data Length	Enter <b>32</b> as the maximum length of the command field value.
Enumerated Value	Set this parameter to <b>on,off</b> , indicating that the device can be enabled or disabled.

## Step 2: Register a Device

After a device is registered with ROMA Connect, a unique ID and key are allocated to the device so that the device can connect to ROMA Connect.

1. In the navigation pane on the left, choose **LINK > Device Management**. On the page displayed, click **Create Device** in the upper right corner.
2. In the **Create Device** dialog box, set device parameters and click **OK**.

**Table 2-19** Parameters for creating a device

Parameter	Description
Integration Application	Select the integration application created in <a href="#">1</a> .
Associated Product	Select the product created in <a href="#">Step 1: Define a Product</a> .
Device ID	Enter a device ID, which is the unique identifier of the device and can be user-defined.
Device Name	Enter a device name, which can be user-defined.
Password	Enter the access password of the device. If you do not specify this parameter, the system automatically generates a value.
Confirm Password	Enter the confirm password of the device. If you do not specify this parameter, the system automatically generates a value.

Parameter	Description
Status	This parameter specifies whether to enable the device. The device can connect to ROMA Connect only after being enabled. Retain the default value, that is, the device is enabled.
Device Tag	You do not need to set this parameter.
Description	Enter a brief description of the device.

3. On the **Device Management** page, click the name of the created device to access the device details page. On the **Topics** tab page, view the topic information about the device.

### Step 3: Configure Data Forwarding Rules

ROMA Connect does not directly store data reported by devices. You need to configure data forwarding rules to forward device data to other services for storage. The following section uses ROMA Connect MQS as an example to describe how to forward data to the destination.

1. In the navigation pane on the left, choose **LINK > Rule Engine**. On the page displayed, click **Create Rule** in the upper right corner.
2. In the **Create Rule** dialog box, set rule parameters and click **OK**.

**Figure 2-16** Creating a rule

**Table 2-20** Rule parameters

Parameter	Description
Application Name	Select the integration application created in <b>1</b> .
Name	Enter a rule name, which can be user-defined.

Parameter	Description
Description	Enter a brief description of the rule.
Status	This parameter specifies whether to enable the rule. Retain the default value, that is, the rule is enabled.

- In the rule list, click the name of the rule created in 2 to access the rule details page.
- In the **Source** area, click **Create Source**, set related parameters and click **Save**.

**Figure 2-17** Creating a source

**Table 2-21** Source parameters

Parameter	Description
Product Name	Select the product created in <a href="#">Step 1: Define a Product</a> .
Device Name	Select the device created in <a href="#">Step 2: Register a Device</a> .
Topic Name	Select the topic to which the device sends messages. The topic name is in the format of <i>{Product ID}/out/{Device ID}</i> .
Topic Level	Select a topic level. The value is automatically adapted based on the value of <b>Device Name</b> . If you do not specify <b>Device Name</b> , <b>Topic Level</b> is <b>Product</b> . If you select a specific device from the <b>Device Name</b> drop-down list box, <b>Topic Level</b> is <b>Device</b> .
Base64 Encoding	Specifies whether Base64 encoding is performed on the forwarded device data. You do not need to enable this option.
Include Device Data	Specifies whether the forwarded data includes device data. To facilitate subsequent identification of the device to which the forwarded data belongs, enable this option.

- In the **Destination** area, click **Create Destination**, set related parameters and click **Save**.

**Figure 2-18** Creating a destination

**Table 2-22** Destination parameters

Parameter	Description
Destination	Select <b>ROMA MQS</b> to integrate messages forwarded to ROMA Connect MQS.
Connection Address	Select the MQS intranet address of the current ROMA Connect instance. You can view the address on the <b>Instance Information</b> page of the ROMA Connect console.
Topic Name	Select the topic created for message integration in <a href="#">Prerequisites</a> .
Username	This parameter is available only if <b>MQS SASL_SSL</b> is enabled for the ROMA Connect instance. Enter the key of the integration application to which the topic defined in <b>Topic Name</b> belongs.
Password	This parameter is available only if <b>MQS SASL_SSL</b> is enabled for the ROMA Connect instance. Enter the secret of the integration application to which the topic defined in <b>Topic Name</b> belongs.


## Step 4: Debug the Device

A device can access ROMA Connect by integrating the MQTT.fx client or SDK. Use the MQTT.fx client to simulate a device to connect and send messages to ROMA Connect.

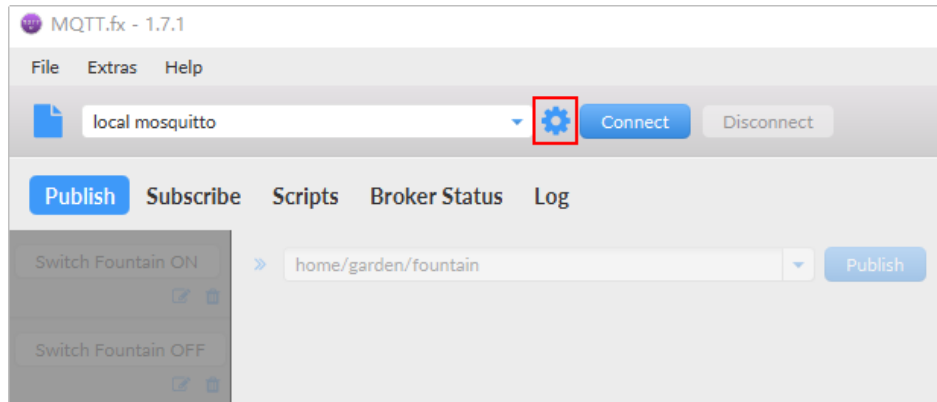
1. Obtain device access information.

In the device list on the **Device Management** page, view and record the values of **MQTT Connection Address**, **Client ID**, **Username**, and **Password** of the device created in [Step 2: Register a Device](#).

**Figure 2-19** Device access information

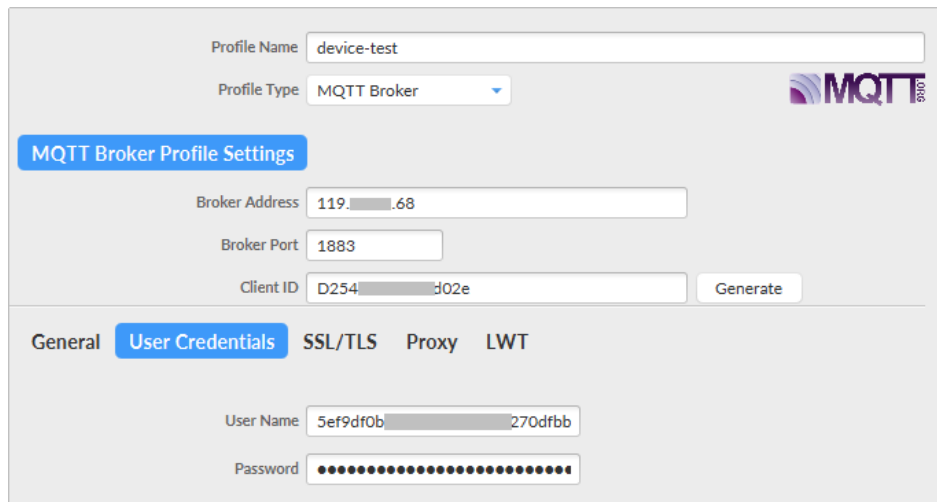
- Open the MQTT.fx client installed on your local PC and click  on the client.

**Figure 2-20** MQTT.fx client



- On the page for editing connection attributes, enter the connection attribute configuration information and click **OK**.

**Figure 2-21** Configuring connection attributes



**Table 2-23** Connection attribute configuration

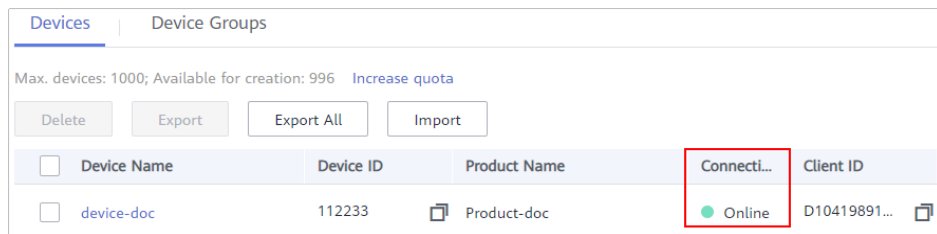
Parameter	Description
Profile Name	Enter a name for the connection configuration.
Profile Type	Select <b>MQTT Broker</b> .
Broker Address	Enter the IP address in <b>MQTT Connection Address</b> obtained in <a href="#">1</a> .
Broker Port	Enter the port number in <b>MQTT Connection Address</b> obtained in <a href="#">1</a> .
Client ID	Enter the client ID obtained in <a href="#">1</a> .

Parameter	Description
User Credentials - User Name	Enter the username obtained in <b>1</b> .
User Credentials - Password	Enter the password obtained in <b>1</b> .

4. Click **Connect** to connect to ROMA Connect.

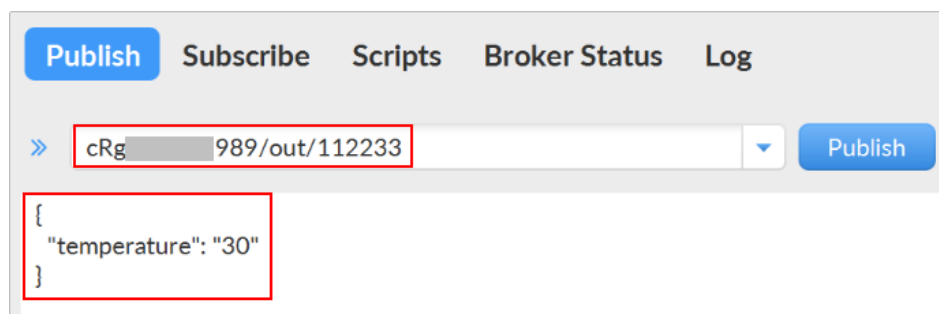
If the connection is successful, the connection status in the upper right corner of the MQTT.fx client will be displayed in green. On the ROMA Connect console, choose **LINK > Device Management**. On the page displayed, **Status** of the device will be displayed as **Online**.

**Figure 2-22** Device online



5. Enable the device to send data to ROMA Connect.
  - a. On the MQTT.fx client, click the **Publish** tab, enter the device topic and message content, and click **Publish** to send a message to ROMA Connect.
    - Enter the topic of the message sent by the device in the text box on the left of the **Publish** button. The topic name is in the format of *{Product ID}/out/{Device ID}*. You can choose **LINK > Device Management** on the ROMA Connect console to view the topic.
    - Enter the message content in JSON format in the message text box.

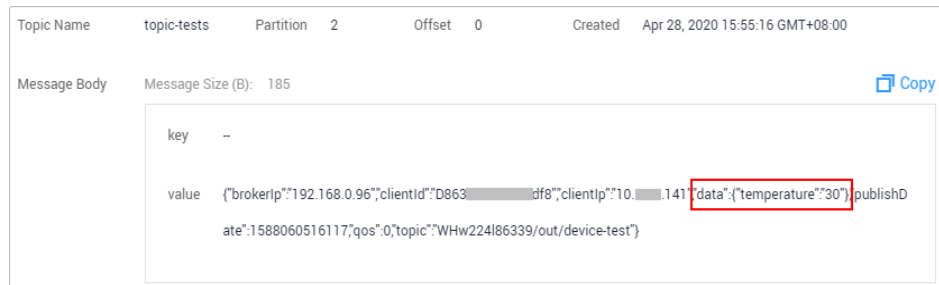
**Figure 2-23** Enabling a device to send a message



- b. On the ROMA Connect console, choose **Message Queue Service > Message Query**.
- c. Click **Advanced Search** to expand advanced search options.
- d. Enter search criteria and click **Search** to query the messages sent by the client.

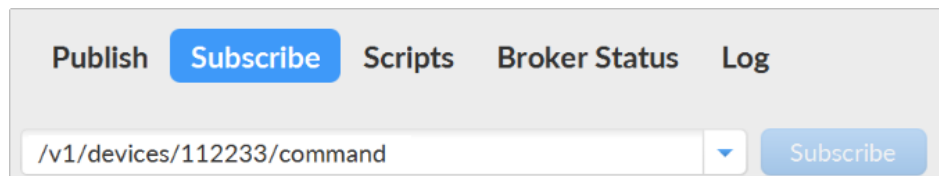
- Select the message topic created in **Step 3: Configure Data Forwarding Rules** for **Topic Name**.
  - Select **Creation time** for **Search By** and select the time segment for the device to send messages to ROMA Connect.
- e. Click **Message Body** and check whether the value of the **data** field in the message reported by the device is the same as that sent by the MQTT.fx client.

**Figure 2-24** Viewing the message body



6. Enable the device to receive a command from ROMA Connect.
- a. On the MQTT.fx client, click the **Subscribe** tab, enter the topic (in the format of `/v1/devices/{Device ID}/command`) of the command delivery request, and click **Subscribe**.  
You can choose **LINK > Device Management** on the ROMA Connect console to view the topic.

**Figure 2-25** Subscribing to device messages



- b. On the ROMA Connect instance console, choose **LINK > Device Management**. In the **Operation** column of the device, choose **More > Deliver Command**.
- c. In the **Deliver Command** dialog box, set the command to be delivered and click **OK**.

**Figure 2-26** Delivering a command

**Table 2-24** Parameters for delivering a command

Parameter	Description
Thing Model Service	Select the thing model service added in 3. In this example, select <b>Temperature</b> .
Synchronous Communication	Retain the default value <b>Disable</b> .
Command	Select the command added in 5. In this example, select <b>SET_STATUS</b> .
status	This parameter varies according to the command field added in 5. In this example, the <b>status</b> field is displayed and set to <b>on</b> , indicating that the device is enabled.

- d. On the **Subscribe** tab page of the MQTT.fx client, you can view the command delivered by ROMA Connect to the device. Check whether the values of **cmd** and **paras** in the command are the same as those set on the ROMA Connect console.

**Figure 2-27** Command received by the device

```

/v1/devices/112233/command
1
18-02-2021 23:37:29.85049119 QoS 0
{"mid":38613,"serviceId":"Temperature","deviceId":"D10419891038188tNu0m",
"cmd":"SET_STATUS","paras":{"status":"on"},"msgType":"cloudReq"}
    
```



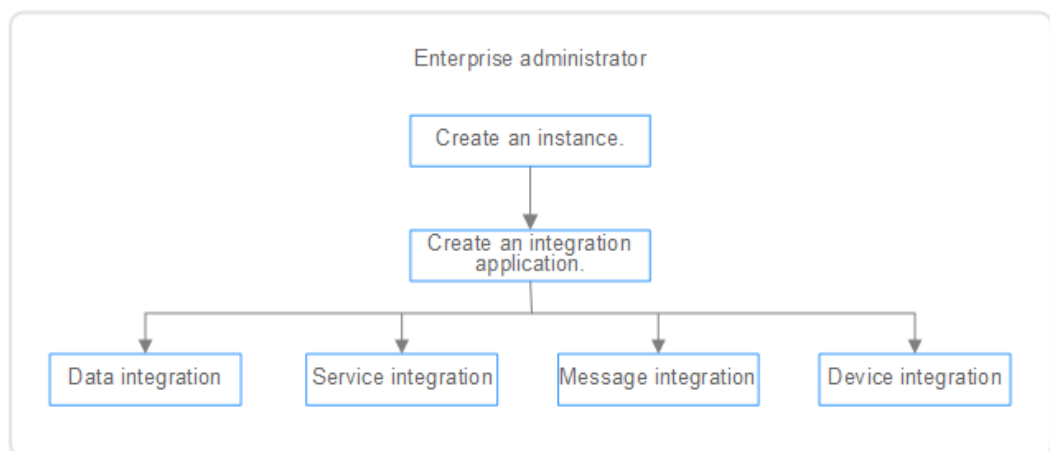
# 3 Getting Started

## Process for Using ROMA Connect

ROMA Connect is a full-stack integration platform that connects applications and data to provide data, API, message, and device integration capabilities. It applies to common enterprise system integration scenarios.

The following figure shows the process for using ROMA Connect.

**Figure 3-1** Process for using ROMA Connect



1. **Create an instance.**  
On the ROMA Connect console, create a ROMA instance and select an instance edition based on service requirements.
2. **Create an integration application.**  
Under the instance, create an integration application. All resources in an instance must belong to an integration application.
3. Select ROMA Connect functions based on service scenarios.
  - **Data integration:** Connects data sources at the source and destination. A data task can be used to integrate data from the source to the destination.

- **Service integration:** Encapsulates existing backend services, data sources, and custom functions into RESTful APIs and exposes them to external systems.
- **Message integration:** Creates message topics through which different systems interconnect with each other to send and receive messages.
- **Device integration:** Defines device models and registers devices in the cloud. Devices are connected to the cloud through SDKs to send and receive messages.

# 4 Instances

---

[Creating a ROMA Connect Instance](#)

[Managing Instances](#)

[Instance Configuration Parameters](#)

## 4.1 Creating a ROMA Connect Instance

### 4.1.1 Preparing Required Resources

#### Overview

Before creating a ROMA Connect instance, you need to ensure availability of resources, including a virtual private cloud (VPC), subnet, and security group. Each ROMA Connect instance is deployed in a VPC and bound to specific subnets and security groups. In this way, ROMA Connect provides an isolated virtual network environment and security protection policies that can be easily configured and managed by users.

#### Required Resources

The following table lists the resources required by a ROMA Connect instance.

**Table 4-1** ROMA Connect resources

Resource	Requirement	Operations
VPC and subnet	<p>Different ROMA Connect instances can use the same or different VPCs and subnets based on site requirements. Note the following when creating a VPC and subnet:</p> <ul style="list-style-type: none"> <li>• The created VPC and ROMA Connect must be in the same region.</li> <li>• Retain the default settings unless otherwise specified.</li> </ul>	<p>For details about how to create a VPC and subnet, see "Creating a VPC and Subnet" in the <i>Virtual Private Cloud User Guide</i>.</p>
Security group	<p>Different ROMA Connect instances can use the same security group or different security groups. Note the following when creating a security group:</p> <ul style="list-style-type: none"> <li>• Set <b>Template</b> to <b>Custom</b>.</li> <li>• After a security group is created, retain the default inbound and outbound traffic rules.</li> <li>• To use ROMA Connect, you must add the security group rules described in <a href="#">Table 4-2</a>. Other rules can be added based on site requirements.</li> </ul>	<p>For details about how to create a security group, see "Creating a Security Group" and "Adding a Security Group Rule" in the <i>Virtual Private Cloud User Guide</i>.</p>
(Optional) EIP	<p>If you want to access ROMA Connect through a public network, apply for an EIP.</p>	<p>For details about how to apply for an EIP, see "Assigning an EIP" in the <i>Elastic IP User Guide</i>.</p>

**Table 4-2** Security group rules

Direction	Protocol	Port	Source IP Address	Description
Inbound	TCP	80	0.0.0.0/0	Access APIC through a public network (without SSL encryption).
Inbound	TCP	443	0.0.0.0/0	Access APIC through a public network (with SSL encryption).

Direction	Protocol	Port	Source IP Address	Description
Inbound	TCP	1883	0.0.0.0/0	Access device integration through a public network (without SSL encryption).
Inbound	TCP	7443	0.0.0.0/0	Access device integration through a public network (using RESTful APIs).
Inbound	TCP	8443	0.0.0.0/0	Access device integration through a public network (with SSL encryption).
Inbound	TCP	9092	0.0.0.0/0	Access MQS through an intranet (without SASL authentication).
Inbound	TCP	9093	0.0.0.0/0	Access MQS through an intranet (with SASL authentication).
Inbound	TCP	9094	0.0.0.0/0	Access MQS through a public network (without SASL authentication).
Inbound	TCP	9095	0.0.0.0/0	Access MQS through a public network (with or without SASL authentication).
Inbound	TCP	9096	0.0.0.0/0	Access MQS through a public network (with or without SASL authentication).
Inbound	TCP	9097	0.0.0.0/0	Access MQS through a public network (with SASL authentication).
Inbound	TCP	9292	0.0.0.0/0	Access MQS through a public network (using RESTful APIs).

## 4.1.2 Creating an Instance

### Overview

Before using ROMA Connect, you need to create a ROMA Connect instance. A ROMA Connect instance is an independent resource space. All operations are performed in an instance. Resources of different instances are isolated from each other. You can use one or more ROMA Connect instances based on service requirements.

## Prerequisites

- A VPC is available, and the subnet and security group have been configured. (For details on how to create a VPC, subnet, and security group, see [Preparing Required Resources](#).)
- The available quota of ROMA Connect instances is not **0**. (If the available quota is **0**, delete unnecessary instances.)

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **Create ROMA Connect Instance**.
2. On the **Create ROMA Connect Instance** page, enter configuration information about the instance. Then, click **Next**.

**Table 4-3** ROMA Connect instance parameters

Parameter	Configuration
Edition	<p>Select the capacity specifications of the instance. The edition cannot be changed after the instance is created.</p> <ul style="list-style-type: none"> <li>• <b>Basic</b> (for small enterprises): This edition supports the integration of 5 to 10 systems and 25 connections.</li> <li>• <b>Professional</b> (for SMEs): This edition supports the integration of 10 to 20 systems and 80 connections.</li> <li>• <b>Enterprise</b> (for medium-sized or large enterprises): This edition supports the integration of 20 to 30 systems and 200 connections.</li> <li>• <b>Platinum</b> (for large enterprises): This edition supports the integration of more than 30 systems and 800 connections.</li> </ul> <p><b>NOTE</b> The number of connections and the number of systems are only used as a reference for selecting ROMA Connect instance edition. They do not limit the number of resources (such as integration tasks, APIs, and message topics) in an instance and are not directly associated with the resource quota of the instance.</p> <ul style="list-style-type: none"> <li>• The number of systems refers to the number of users' service systems interconnecting with the ROMA Connect instance. Multiple connections can be set up between a service system and a ROMA Connect instance.</li> <li>• A connection refers to an interaction between a service system and ROMA Connect. The number of connections varies depending on the functional module in ROMA Connect to connect. The mappings between the number of resources and the number of connections are as follows: <ul style="list-style-type: none"> <li>• FDI: One data integration task occupies one connection.</li> <li>• APIC: 10 hosting APIs (not using custom backends as the backend services) occupy one connection. Two function APIs or data APIs occupy one connection.</li> <li>• MQS: Three topics occupy one connection.</li> <li>• LINK: 1000 devices occupy one connection.</li> </ul> </li> </ul>
CPU Architecture	<p>Select <b>x86</b> or <b>Kunpeng</b> based on the CPU architecture supported by the current environment.</p>
AZ	<p>Select the availability zone (AZ) where the instance resides. Different AZs are physically isolated but can communicate with each other through an internal network.</p> <ul style="list-style-type: none"> <li>• To enhance ROMA Connect availability, create instances in different AZs.</li> <li>• To shorten network latency, create instances in the same AZ.</li> </ul>
Instance Name	<p>Enter the instance name as planned.</p>

Parameter	Configuration
Enterprise Project	Associate the instance with an enterprise project. Available for enterprise accounts.
Description	Enter a brief description of the instance.
VPC	Select the VPC and subnet associated with the instance. The VPC and subnet must have been created in <a href="#">Preparing Required Resources</a> .
Security Group	Select the security group associated with the instance. The security group must have been created in <a href="#">Preparing Required Resources</a> .
Public Access	Determine whether to enable the function of accessing ROMA Connect from a public network based on service requirements.
EIP	This parameter is available only if <b>Public Access</b> is enabled. Select the EIP associated with the instance. The EIP must be applied for in advance.
Kafka Version	Select the Kafka server version used by the instance. Currently, both 1.1.0 and 2.3.0 are supported. Select a version based on the site requirements. Version 2.3.0 is recommended.
MQS Capacity Threshold Policy	Select a processing policy that will be used when the number of messages in MQS reaches the capacity threshold (95% of the storage space). <ul style="list-style-type: none"> <li>• <b>Delete messages:</b> The oldest messages (10% of the total number of messages) are automatically deleted to free up storage space for new requests. This policy has no adverse impact on message creation. It applies when message retrieval is fast and there are no accumulated messages.</li> <li>• <b>Stop creation:</b> New message creation requests will be rejected until a message is deleted according to the configured aging time. After the available storage space is greater than 5%, new message creation requests can be received. This policy applies when message retrieval processing is slow or inventory messages need to be repeatedly retrieved and cannot be deleted in advance.</li> </ul>
MQS SASL_SSL	Determine whether to enable SASL authentication and SSL encryption for message transmission. It is recommended that you enable this parameter to improve data transmission security.



Parameter	Configuration
Intra-VPC Plaintext Access	<p>This parameter is available only if <b>MQS SASL_SSL</b> is enabled.</p> <p>Specify whether plaintext access is used in the VPC.</p> <p>If intra-VPC plaintext access is enabled, you cannot use SASL to connect to message integration topics in the VPC. In this case, no certificate is required for connecting to topics.</p>

3. On the **Confirm** page, click **Submit** to create an instance.

The instance creation takes about 18 to 25 minutes. After the instance is created, the instance status is **Available** on the **Instances** page.

If the ROMA Connect instance fails to be created, delete the failed instance and create a new one. If the creation still fails, contact technical support.

## 4.2 Managing Instances


### Overview





After an instance is created, you can view and edit the instance configuration information on the console, including basic information, configuration parameters, and data dictionaries, as well as import and export assets.

### Viewing Details of an Instance

On the **Instance Information** page of the console, click the **Basic Information** tab to view and edit basic information about the instance.

**Table 4-4** Basic information


Content	Description
Instance	<p>Basic information about the instance, including the instance name, instance ID, instance edition, availability zone, and creation time.</p> <ul style="list-style-type: none"> <li>• You can modify <b>Instance Name</b> and <b>Description</b> as required.</li> <li>• Click  next to <b>Instance ID</b> to copy the instance ID.</li> </ul>
VPC	VPC associated with the instance. Click the VPC name to view the VPC configuration.
Subnet	Subnet associated with the instance. Click the subnet name to view the subnet configuration.
Security Group	Security group associated with the instance. You can click the security group name to view the security group configuration or click <b>Edit</b> to bind a new one.

Content	Description
Routes	Private network segment. After a ROMA Connect instance is created, it can communicate with the VPC subnet segment specified during instance creation by default. If other private network segments need to communicate with the instance, you can configure routes by setting this parameter.
Message Storage Space	Information about the MQS message storage space of the instance, including the storage class and space usage (percentage).
MQS Information	Basic configuration about MQS, including the bandwidth, maximum number of partitions, message tracing, MQS SASL_SSL, intra-VPC plaintext access, RESTful API, and capacity threshold policy. <ul style="list-style-type: none"> <li>You can modify <b>MQS Capacity Threshold Policy</b> as required.</li> <li>You can click  next to <b>RESTful API</b> to copy the API access address.</li> </ul>
Connection Addresses	Connection addresses of the instance, including the addresses for connecting to ROMA Connect, LINK, APIC, and MQS. Click  next to the connection address to copy the connection address.
Public Access	EIP bound to the instance. <ul style="list-style-type: none"> <li>If an EIP has been bound to the instance, you can click  on the right of the EIP to copy the address information.</li> <li>If an EIP has been bound to the instance, you can click <b>Unbind</b> on the right of the address to unbind the EIP from the instance.</li> <li>If an EIP has not been bound to the instance, you can click  on the right of the address to bind an EIP to the instance.</li> </ul>

## Modifying Configuration Parameters

On the **Instance Information** page of the ROMA Connect console, click the **Configuration Parameters** tab and view the configuration parameters of the instance. You can also change the values of **Current Value**.

**Table 4-5** Configuration parameters

Content	Description
Parameter Name	Name of a parameter. You can move the cursor to  before a parameter name to view its description.
Default Value	Default value of a parameter.
Value Range	Value range of a parameter.
Current Value	Current value of a parameter.
Updated	Time when a parameter was last updated. If the parameter has never been modified, this parameter is left blank.
Operation	Click <b>Edit</b> to change the value of <b>Current Value</b> .


## Creating a Data Dictionary

A data dictionary is used to open up key values in a system database to users. DEVICE\_TYPE3 is the default data dictionary. You can also customize a data dictionary. To create a data dictionary, perform the following steps:

1. On the **Instance Information** page of the console, click the **Data Dictionaries** tab.
2. Click **Create Data Dictionary**.
3. In the **Create Data Dictionary** dialog box, set dictionary parameters and click **OK**.

**Table 4-6** Parameters for creating a data dictionary

Parameter	Description
Dictionary Code	Enter the unique ID of a dictionary.
Dictionary Name	Enter a dictionary name.
Extended Field 1	These parameters are reserved for the dictionary and can be left empty.
Extended Field 2	
Description	Enter a brief description of the dictionary.

4. After the data dictionary is created, click  next to the dictionary name to expand the dictionary item list.
5. Click **Create Dictionary Item**.
6. In the **Create Dictionary Item** dialog box, set dictionary item parameters and click **OK**.

**Table 4-7** Parameters for creating a dictionary item

Parameter	Description
Item Code	Enter the unique ID of a dictionary item.
Item Name	Enter the dictionary item name.
Extended Field 1	These parameters are reserved for the dictionary item and can be left empty.
Extended Field 2	
Description	Enter a brief description of the dictionary item.

## Importing and Exporting Assets

ROMA Connect allows you to export application and task assets of an instance as files to a local PC or import local asset files to ROMA Connect to migrate assets in batches.

### NOTE

- Ensure that the instance configurations during import and export are the same. Otherwise, the import may fail. For example, if app\_route is not enabled on the exported instance but is enabled on the instance to be imported, the asset import may fail when the imported asset package contains APIs with conflict paths.
- Request throttling policies, access control policies, and signature keys of APIs cannot be imported or exported.
- Importing assets
  - a. On the **Instances** page of the console, click **Import Asset** in the upper right corner.
  - b. In the **Import Asset** dialog box, select **Append** or **Overwrite** based on the site requirements.
    - The data source IDs of different instances must be different. If the data source IDs are the same, select **Overwrite**.
    - The data integration task names in different instances must be different. If the data integration task names are the same, enter a prefix in the text box.
  - c. In the dialog box displayed, select a local asset file and import it.
  - d. After the import is complete, click **Close**. You can view the imported asset in the instance.

### NOTE

- The custom authorizer name is globally unique. If a custom authorizer with the same name already exists in the instance when the asset is imported, random characters will be added to the end of the imported custom authorizer name to avoid duplicate names.
- If the following error information is displayed during asset import, contact technical support to upgrade the instance:  

```
"CipherEntry":{"reason":{"resourceType":"cipherEntries"},"resourceId":"\\","resourceName":null,"errorCode":"APIG.9999","errorMsg":"System error"}}}
```

- Exporting assets
  - a. On the **Instances** page of the console, click **Export Asset** in the upper right corner.
  - b. Export assets.
    - Exporting specified assets: Select the application or task assets to be exported and click **Export Selected** to export the asset files to the local PC.
    - Exporting all assets: Click **Export All** to export all asset files to the local PC.

If you need to export all API asset information when exporting application assets, select **Export all assets in applications**. All API asset information including self-created and authorized APIs will be exported. If you do not select this parameter, information about self-owned and self-authorized APIs will be exported.
  - c. After the export is complete, click **Close**. You can view the exported assets in the local PC.

## 4.3 Instance Configuration Parameters

The following table lists the instance configuration parameters of ROMA Connect.

**Table 4-8** Instance configuration parameters

Parameter	Description
ratelimit_api_limits	Default request throttling value applied to all APIs. If no request throttling policy is bound to an API, the total number of times the API can be called is determined by this parameter value. If a request throttling policy has been bound to an API, the total number of times the API can be called is determined by the bound policy.
request_body_size	Maximum size of the body allowed in an API request.
backend_timeout	Maximum timeout duration for ROMA Connect to send a request to a backend service.

Parameter	Description
app_token	<p>Whether to enable app_token authentication. After this function is enabled, the obtained access token can be used in API requests for authentication during API calling.</p> <ul style="list-style-type: none"> <li>• app_token_expire_time indicates the validity period of the access token. Before the access token expires, you must obtain a new access token.</li> <li>• refresh_token_expire_time indicates the validity period of the refresh token. A refresh token is used to obtain a new access token.</li> <li>• app_token_uri indicates the URI used for obtaining an access token.</li> <li>• app_token_key indicates the encryption key of the access token.</li> </ul>
app_basic	<p>Whether to enable app_basic authentication. After this function is enabled, you can add the <b>Authorization</b> parameter to an API request to carry the key and secret of an integration application (or the AppKey and AppSecret of a client) for authentication during API calling.</p>
app_secret	<p>Whether to enable app_secret authentication. After this function is enabled, you can add the <b>X-HW-ID</b> and <b>X-HW-AppKey</b> parameters to an API request to carry the key and secret of an integration application (or the AppKey and AppSecret of a client) for authentication during API calling.</p> <p>If the app_secret authentication mode is used, the app_api_key authentication mode must be disabled.</p>
app_route	<p>Whether to support IP address access. After this function is enabled, APIs in non-DEFAULT groups can be called by using IP addresses.</p>
cascade	<p>Whether to enable the API cascading function. After this function is enabled, APIs can be called across instances.</p> <ul style="list-style-type: none"> <li>• cascade_auth_key indicates the encryption key used for authentication between APIs in the cascading relationship.</li> <li>• cascade_instance_ids indicates the ID list of cascading instances. Only instances specified by this parameter can establish the cascading relationship with the current instance.</li> </ul>
default_group_hide	<p>Whether to hide the DEFAULT group. After this function is enabled, the DEFAULT group is hidden.</p>

Parameter	Description
livedata_config	<p>Custom backend configuration.</p> <ul style="list-style-type: none"> <li>• sandbox_max_memory indicates the maximum memory required for executing a script of a function backend.</li> <li>• sandbox_max_memory indicates the maximum CPU time required for executing a script of a function backend.</li> <li>• livedata_env indicates the running mode of a custom backend. If this parameter is set to <b>online</b>, the custom backend will return responses in the online format.</li> <li>• gw_address_protocol indicates the default request protocol transferred through the <b>DICT:gw_rest_float_addr</b> field in a function script.</li> <li>• procedure_async indicates whether the stored procedure is executed in asynchronous mode.</li> <li>• dataapi_return_type indicates the response format of a data backend.</li> </ul>
backend_client_certificate	<p>Whether to enable two-way authentication for a backend. After this function is enabled, you can configure the two-way authentication for a backend when configuring the backend information during API creation.</p>
ssl_ciphers	<p>HTTPS cipher suite that can be configured. You can select the cipher suites as required.</p>
real_ip_from_xff	<p>Whether to use the IP addresses in the <b>X-Forwarded-For</b> header as the criterion for access control and request throttling.</p> <p>xff_index: Sequence number of an IP address in the <b>X-Forwarded-For</b> header. The value can be positive, negative, or 0.</p> <ul style="list-style-type: none"> <li>• If the value is 0 or positive, obtain the IP address of the corresponding index in the <b>X-Forwarded-For</b> header.</li> <li>• If the value is negative, obtain the IP address of the indicated reverse sequence in the <b>X-Forwarded-For</b> header.</li> </ul> <p>For example, assume that the <b>X-Forwarded-For</b> header of a request received by API gateway contains three IP addresses: IP1, IP2, and IP3. If the value of xff_index is <b>0</b>, IP1 is obtained. If the value of xff_index is <b>1</b>, IP2 is obtained. If the value of xff_index is <b>-1</b>, IP3 is obtained. If the value of xff_index is <b>-2</b>, IP2 is obtained.</p>

# 5 Integration Application Management

[Creating an Integration Application](#)

[Configuring Integration Application Authorization](#)

## 5.1 Creating an Integration Application

### Overview

ROMA Connect uses integration applications to isolate resources of different users in the same instance. Resources (such as data sources, APIs, topics, and products) created in a ROMA Connect instance must belong to an integration application. By default, IAM users can view and manage only the integration applications and resources created by themselves and a primary account can view and manage all resources created by IAM users under it.

To share resources created by yourself to other IAM users, you can authorize the integration applications to other IAM users. For details, see [Configuring Integration Application Authorization](#).

### Creating an Integration Application

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **Integration Applications**. In the upper right corner of the page, click **Create Integration Application**.
3. In the dialog box displayed, enter the application information and click **OK**.

Parameter	Description
Name	Set an application name as required.
Description	Enter the description of the application. This parameter is optional.

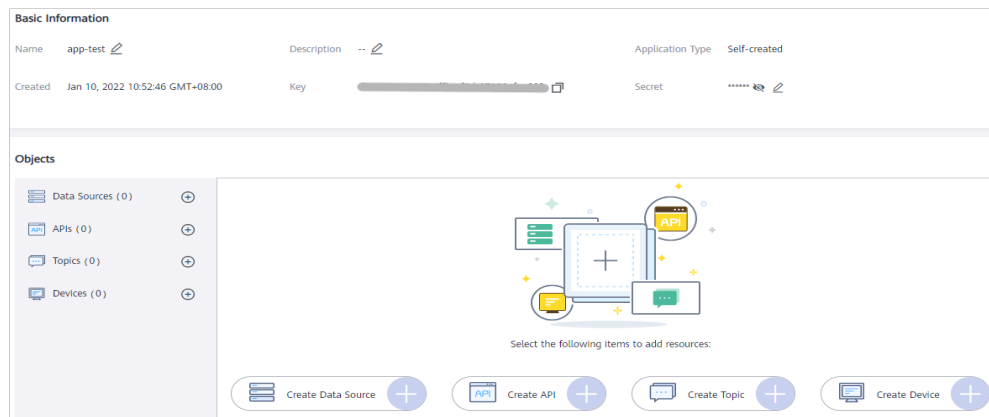


Parameter	Description
Key	Enter the key of the integration application. If this parameter is not specified, the system automatically generates a key.
Secret	Enter the secret of the integration application. If this parameter is not specified, the system automatically generates a secret.


## Viewing and Editing an Integration Application

On the **Integration Applications** page of the ROMA Connect console, click the integration application name to view its details.

**Figure 5-1** Viewing application details



**Table 5-1** Integration application information

Content	Description
Basic Information	<p>Basic parameters of the application, including <b>Name</b>, <b>Description</b>, <b>Integration Application Type</b>, <b>Created</b>, <b>Key</b>, and <b>Secret</b>. <b>Key</b> and <b>Secret</b> can be used to authorize the access to application resources.</p> <ul style="list-style-type: none"> <li>You can modify <b>Name</b> and <b>Description</b> as needed.</li> <li>You can click  on the right of <b>Secret</b> to view secret information.</li> <li>You can reset the secret when required. After the secret is reset, the old secret will automatically become invalid.</li> <li>For accounts, the integration applications are all self-created applications. For IAM users, the integration applications created by themselves are called self-created applications, and the integration applications created and authorized by other IAM users are called authorized applications.</li> </ul>

Content	Description
Objects	You can view, create, edit, and delete resources in an integration application, including data sources, APIs, topics, and devices. For details, see <a href="#">data source</a> , <a href="#">API</a> , <a href="#">topic</a> , and <a href="#">device</a> usage guides.

## 5.2 Configuring Integration Application Authorization

### Overview


ROMA Connect provides strict permissions management for user resources. By default, IAM users can view and manage only self-created integration applications and resources under the same instance, but not integration applications and resources created by other IAM users. With integration application authorization, IAM users can share applications and resources with other IAM users under the same account.

### Configuring Integration Application Authorization

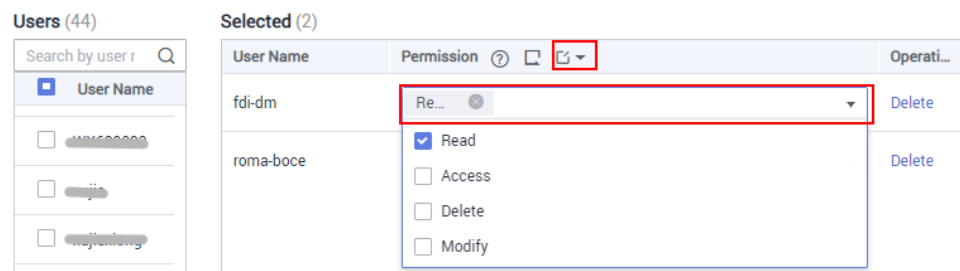
1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Integration Applications**. On the page displayed, locate the target integration application and click **Authorize User**.

#### NOTE

A primary account can grant permissions for integration applications created by all IAM users, but an IAM user can grant permissions only for self-created integration applications.

3. In the **Authorize User** dialog box displayed, select the IAM user to be authorized in the **Users** pane and grant permissions to the selected user in the **Selected** pane. You can grant permissions to each user separately or click  on the right of **Permission** to grant permissions to all selected users in batches.


**Figure 5-2** Granting permissions for applications



**Table 5-2** User permissions for applications

Permission	FDI	APIC	MQS	LINK
Read	View data sources of applications and use the data sources to create data integration tasks.	View, debug, and export APIs of applications.	View topics of applications.	View devices, products, and rules of applications, and export devices and products for debugging.
Modify	Create and edit data sources under an application.	Create, edit, release, take APIs offline, and import APIs under an application.	Create a topic under an application.	Create and edit devices, products, and rules under an application, import devices and products, and reset device and product passwords.
Delete	Delete data sources of an application.	Delete APIs of an application.	Delete topics of an application.	Delete devices, products, and rules of an application, as well as product attributes, device topics, rule data source, and rule data destination.
Access	N/A	Configure authorization , access control, traffic control, and signature key binding for APIs of applications.	Configure application and user permissions for topics of applications.	N/A

Permission	FDI	APIC	MQS	LINK
Admin	Permission of the application administrator.			

4. Click **OK**.  
In the integration application list, click  on the left of an integration application to view the IAM users who have been granted permissions, and the permission scope.
5. Log in to the ROMA Connect console as the IAM user authorized in **3** and access the corresponding ROMA Connect instance to check whether the user has the management permissions on the integration application and resources.

# 6 Data Integration Guide

---

[Usage Introduction](#)

[Connecting to Data Sources](#)

[Creating a Data Integration Task](#)

[Creating a Composite Data Integration Task](#)

[Starting or Stopping a Data Integration Task](#)

[Managing a Data Integration Task](#)

## 6.1 Usage Introduction

### Function Description

FDI is a data integration component of ROMA Connect and supports data integration and conversion between multiple data sources. ROMA Connect has the following advantages for data integration:

- **Access of multiple data source types**

By default, ROMA Connect supports connections of multiple types of data sources, such as relational databases, big data storage, semi-structured storage, and message systems. For details about the supported data source types, see [Data Sources Supported by Data Integration Task](#).

If the default data source types provided by ROMA Connect cannot meet your data integration requirements, you can customize data sources. For details on how to customize a data source, see [Connecting to a Custom Data Source](#).

- **Flexible integration modes**

ROMA Connect supports the following integration modes:

- **Scheduled:** ROMA Connect periodically obtains data from the source based on a task schedule and then integrates the data to the destination.
- **Real-time:** ROMA Connect integrates data generated at the source to the destination in real time.

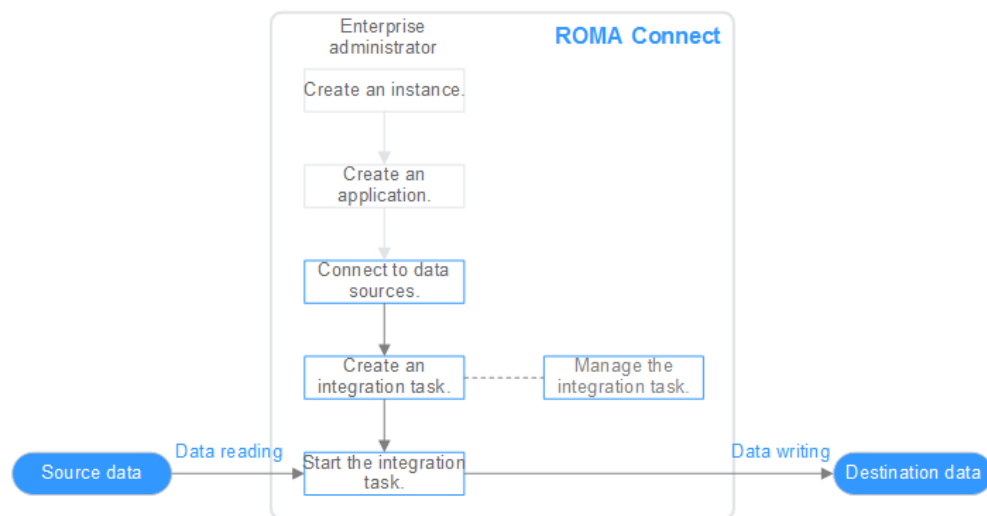
For details about the data source types supported in the two integration modes, see [Connecting to Data Sources](#).

- Custom mapping rules**  
 When converting data fields from the source to destination, you can customize mapping rules. For example, you can replicate one data column in source data to multiple data columns and then integrate these columns to the destination.
- Data integration between different network environments**  
 FDI allows data sources at the source and destination to come from different network environments that are not interconnected. For example, if the data source at the source comes from an on-premises data center and the data source at the destination comes from a VPC on the cloud, FDI can access both the data sources, implementing data integration between different network environments.
- Resumable transmission of real-time tasks**  
 After a fault on the source or destination is rectified or a task is manually restarted, FDI automatically resumes data collection from the last interrupted position to prevent data loss.

## Process Flow

The following figure shows how data integration is performed using ROMA Connect.

**Figure 6-1** Using ROMA Connect for data integration



1. You have **created a ROMA Connect instance and an integration application.**
2. **Access data sources.**  
 Access data sources at the source and destination to ensure that data can be read from the source and written to the destination.
3. **Create an integration task.**  
 A data integration task defines detailed rules for data integration from the source to destination. The rules include data source types on both the source and destination, mapping rules of data fields, and filtering conditions for data

integration. ROMA Connect allows you to create the following data integration tasks:

- **Common Data Integration Task:** Supports all default data source types and two integration modes: scheduled and real-time. For database data sources, only one data table at the source can be integrated to one data table at the destination each time.
- **Composite Data Integration Task:** Uses the Change Data Capture (CDC) to implement real-time and incremental synchronization of data from the source to the destination. Multiple data tables at the source can be integrated to multiple data tables at the destination. Currently, the following relational databases are supported: Oracle, MySQL, and SQL Server. For details, see [CDC Configurations](#).

4. **Start the integration task.**

- After a scheduled task is started, ROMA Connect integrates data on a scheduled basis. During the first execution, all source data that meets the conditions is integrated to the destination. Then, full data that meets the conditions or only incremental data will be integrated based on the task configuration.
- After a real-time task is started, ROMA Connect continuously detects data changes at the source. During the first execution, all source data that meets the conditions is integrated to the destination. Subsequently, only new data will be integrated to the destination each time.

## 6.2 Connecting to Data Sources

Before creating a data integration task, you need to connect data sources at the source and destination to ROMA Connect so that data can be read from the source and written to the destination.

The configuration for data source connection varies depending on data source types. After a data source is connected, click the data source name to view the data source details, which include relevant task information.

 **NOTE**

- ROMA Connect data integration is applicable to data type conversion and integration between heterogeneous data sources. DRS is recommended for data migration and synchronization between mainstream databases. In scenarios where data such as relational database data, big data, and text is migrated to the data lake, DataArts Studio is recommended.
- If the destination data source is FI Hive or MRS Hive, the source data source can only be API, MQS, or SAP.

**Table 6-1** Configurations for data source connections

Data Source Connection	Integration Mode at the Source	Integration Mode at the Destination
<a href="#">Connecting to an API Data Source</a>	Real-time	Scheduled and real-time

<b>Data Source Connection</b>	<b>Integration Mode at the Source</b>	<b>Integration Mode at the Destination</b>
<a href="#">Connecting to an ActiveMQ Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an ArtemisMQ Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a DB2 Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a DIS Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a DWS Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a GaussDB 100 Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a GaussDB 200 Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an FTP Data Source</a>	Real-time	Real-time
<a href="#">Connecting to an FI HDFS Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an FI Hive Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an FI Kafka Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an HL7 Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a HANA Data Source</a>	Real-time	-
<a href="#">Connecting to a HIVE Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an LDAP Data Source</a>	Real-time	-
<a href="#">Connecting to an IBM MQ Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a Kafka Data Source</a>	Real-time	Scheduled and real-time



<b>Data Source Connection</b>	<b>Integration Mode at the Source</b>	<b>Integration Mode at the Destination</b>
<a href="#">Connecting to a MySQL Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a MongoDB Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an MRS Hive Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an MRS HDFS Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an MRS HBase Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an MRS Kafka Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an OBS Data Source</a>	Real-time	Real-time
<a href="#">Connecting to an Oracle Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a PostgreSQL Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a Redis Data Source</a>	-	Scheduled and real-time
<a href="#">Connecting to a RabbitMQ Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to an SAP Data Source</a>	Real-time	-
<a href="#">Connecting to an SNMP Data Source</a>	Real-time	-
<a href="#">Connecting to a SQL Server Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a TaurusDB Data Source</a>	Real-time	Scheduled and real-time
<a href="#">Connecting to a WebSocket Data Source</a>	Real-time	-
<a href="#">Connecting to a Custom Data Source</a>	Scheduled and real-time	Scheduled and real-time

## 6.3 Creating a Data Integration Task

### 6.3.1 Configuring Basic Information

#### Overview

You can convert data among different data sources by creating a data integration task on ROMA Connect. Based on your configured data integration task, ROMA Connect determines how to integrate data in a specific data table at the source to a data table at the destination.

A data integration task includes basic information, schedule (optional), source information, destination information, mapping information, fault information storage (optional), and post-integration operation (optional). This topic describes how to configure basic information and create a schedule for a data integration task.

#### Prerequisites

- The source and destination data sources have been connected to ROMA Connect. (For details, see [Connecting to Data Sources](#).)
- ROMA Connect has the permission to write data to the destination.
- If you need to configure data storage for abnormal synchronization is abnormal, ensure that the OBS data source has been connected. For details, see [Connecting to an OBS Data Source](#).

#### Configuring Basic Information

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Fast Data Integration > Task Management**. On the page displayed, click **Create Task**.
3. On the **Create Task** page, configure basic task information.

**Table 6-2** Basic task information

Parameter	Description
Task Name	Enter a task name. It is recommended that you enter a name based on naming rules to facilitate search.
Description	Enter a brief description of the task.

Parameter	Description
Integration Mode	<p>Select <b>Scheduled</b> as the mode used for data integration.</p> <ul style="list-style-type: none"> <li>• <b>Scheduled:</b> A data integration task is executed according to the schedule to integrate data from the source to the destination.</li> <li>• <b>Real-Time:</b> The data integration task continuously detects updates to the data at the source and integrates updates to the destination in real time.</li> </ul> <p>The data integration mode varies depending on the data source. For details, see <a href="#">Table 6-1</a>.</p>
Tag	<p>Add a tag to classify tasks for quick search. You can select an existing tag for association. If no tag is available, click <b>Add Tag</b> to add a new tag. The new tag is saved when the task is saved. You can search for the tag when creating a task.</p>

4. Proceed to the next step.
  - If **Integration Mode** is set to **Scheduled**, [create a schedule](#).
  - If **Integration Mode** is set to **Real-Time**, [configure source information](#).

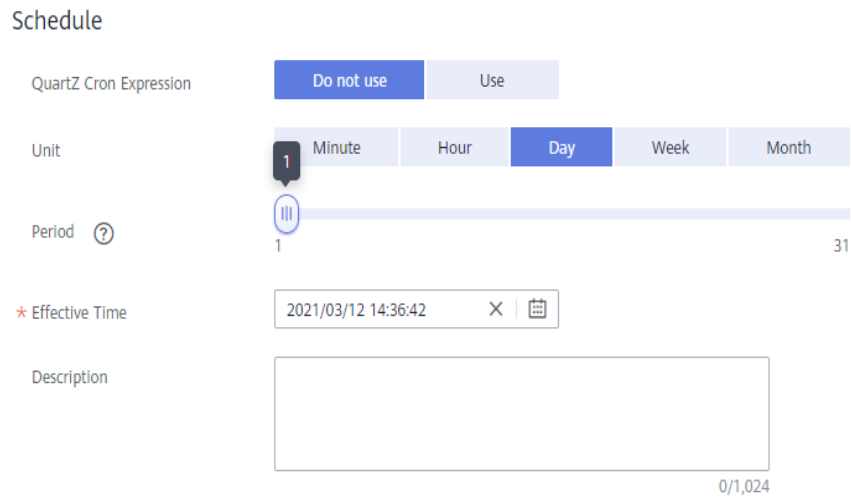
## (Optional) Creating a Schedule

If **Integration Mode** of a task is set to **Scheduled**, you need to configure schedule information for the task.

### NOTE

- If you need to modify a schedule after a data integration task is created, stop the task first.
  - If a large amount of data needs to be synchronized and the task execution interval is short, the next task scheduling time may arrive before the previous task scheduling is complete. In this case, ROMA Connect suspends new scheduling and waits until the previous scheduling is complete.
1. On the **Create Task** page, configure schedule information. ROMA Connect provides two methods for configuring a schedule:
    - **GUI configuration**  
Currently, only the simple periodic configuration is supported, for example, every few minutes, hours, or days.

**Figure 6-2** GUI configuration



**Table 6-3** Simple GUI configuration

Parameter	Description
Quartz Cron Expression	Select <b>Do not use</b> .
Unit	Set the unit of the task execution period. The value can be <b>Minute, Hour, Day, Week, or Month</b> .
Period	Set the task execution cycle. The value range varies depending on the value of <b>Unit</b> . For example, if <b>Unit</b> is set to <b>Day</b> and <b>Period</b> is set to <b>1</b> , the data integration task is executed once a day.
Effective Time	Specify the time at which you want the task to be executed. The task will be executed in the next scheduling period. For example, if <b>Effective Time</b> is set to 2020-2-2 02:02:02 and the scheduling period is one day, the task is executed for the first time at 02:02:02 on February 3, 2020.
Description	Enter the description of the schedule.

– **Using Quartz Cron expression**

The Quartz Cron expression supports flexible schedules. For example, a task can be executed every 15 minutes from 01:00 to 04:00 every day based on the settings of the Quartz Cron expression. Such a schedule cannot be implemented through simple GUI configuration, but using the Quartz Cron expression.

0 0/15 1-4 \* \* ?

**Figure 6-3** Quartz Cron expression

Schedule

Quartz Cron Expression Do not use **Use**

\* Expression Schedule ?

Second	Minute	Hour	Day	Month	Week
0	0/15	1-4	*	*	?

\* Effective Time  × 📅

Description

0/1,024

**Table 6-4** Using Quartz Cron expression

Parameter	Description
Quartz Cron Expression	Select <b>Use</b> .
Expression Schedule	Specify the Quartz Cron expression for the schedule. Currently, ROMA Connect supports only minute-level schedules. The second in the expression is fixed to <b>0</b> . For example, if a task needs to be executed every 15 minutes from 01:00 to 04:00 every day, the Quartz Cron expression corresponding to the schedule is as follows: 0 0/15 1-4 * * ?
Effective Time	Specify the time at which you want the task to be executed. The task will be executed in the subsequent scheduling period. For example, if <b>Effective Time</b> is set to 2020-2-2 02:02:02 and <b>Quartz Cron Expression</b> is set to <b>0 0/15 1-4 * * ?</b> , the task is executed for the first time at 02:15:00 on February 3, 2020.
Description	Enter the description of the schedule.

2. After configuring the task schedule, proceed with [Configuring Source Information](#).

## 6.3.2 Configuring Source Information

### Overview

This topic describes how to configure source information for a data integration task. Based on the source information, ROMA Connect integrates data, including

the data source type, data format, and data range. The source information configuration varies depending on data source types.

Data Source Types Supported by Scheduled Integration Tasks		Data Source Types Supported by Real-Time Integration Tasks
<ul style="list-style-type: none"> <li>• <a href="#">API</a></li> <li>• <a href="#">DB2</a></li> <li>• <a href="#">DWS</a></li> <li>• <a href="#">FTP</a></li> <li>• <a href="#">FI HDFS</a></li> <li>• <a href="#">FI Hive</a></li> <li>• <a href="#">GaussDB 100</a></li> <li>• <a href="#">GaussDB 200</a></li> <li>• <a href="#">HANA</a></li> <li>• <a href="#">Hive</a></li> <li>• <a href="#">LDAP</a></li> <li>• <a href="#">MySQL</a></li> <li>• <a href="#">MongoDB</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">MRS Hive</a></li> <li>• <a href="#">MRS HDFS</a></li> <li>• <a href="#">MRS HBase</a></li> <li>• <a href="#">OBS</a></li> <li>• <a href="#">Oracle</a></li> <li>• <a href="#">PostgreSQL</a></li> <li>• <a href="#">SAP</a></li> <li>• <a href="#">SNMP</a></li> <li>• <a href="#">SQL Server</a></li> <li>• <a href="#">TaurusDB</a></li> <li>• <a href="#">Custom</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">ActiveMQ</a></li> <li>• <a href="#">ArtemisMQ</a></li> <li>• <a href="#">DIS</a></li> <li>• <a href="#">FI Kafka</a></li> <li>• <a href="#">HL7</a></li> <li>• <a href="#">IBM MQ</a></li> <li>• <a href="#">Kafka</a></li> <li>• <a href="#">MRS Kafka</a></li> <li>• <a href="#">RabbitMQ</a></li> <li>• <a href="#">WebSocket</a></li> <li>• <a href="#">Custom</a></li> </ul>

## API

If **Integration Mode** is set to **Scheduled**, you can select API as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-5** API information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the API data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>API</b> .
Data Source Name	Select the API data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Paging	<p>This parameter specifies whether data is returned on multiple pages when ROMA Connect sends a request to the API data source to obtain data. Multiple data records can be returned for one API request.</p> <ul style="list-style-type: none"> <li>• If <b>Paging</b> is enabled, all data that meets the conditions is displayed on multiple pages based on the fixed number of records on each page. Each time an integration task is executed, ROMA Connect sends multiple API requests to obtain all data. That is, each API request is sent to obtain data on one page.</li> <li>• If <b>Paging</b> is disabled, ROMA Connect obtains all data that meets the conditions through one API request.</li> </ul>
Page Number Field	<p>This parameter is mandatory only if <b>Paging</b> is enabled. Enter a page number field defined in the API data source, for example, <b>pageNo</b>. This parameter is carried when ROMA Connect sends an API request to the source to specify the number of the page from which data is to be obtained. <b>Value</b> indicates whether the page number starts from 0 or 1. Set <b>Value</b> based on the original definition of the API.</p> <p>The page number field must be configured in <b>Params</b> or <b>Body of Request Parameters</b>.</p>
Page Size Field	<p>This parameter is mandatory only if <b>Paging</b> is enabled. Enter a page size field defined in the API data source, for example, <b>pageSize</b>. This parameter is carried when ROMA Connect sends an API request to the source to specify the maximum number of records on each page. Set the number of records on each page based on the original definition of the API.</p>
Maximum Number of Pages	<p>This parameter is mandatory only if <b>Paging</b> is enabled. This parameter specifies the maximum number of pages that can be queried in each scheduled task, for example, 10. If the number of pages exceeds the specified value, the task is stopped. The value <b>0</b> indicates that no restriction applies.</p>
Pagination End	<p>This parameter is mandatory only if <b>Paging</b> is enabled. Select the method to stop obtaining source data in pagination mode.</p> <ul style="list-style-type: none"> <li>• <b>Empty page list:</b> If no data record is returned, ROMA Connect stops obtaining source data.</li> <li>• <b>Number of records:</b> ROMA Connect compares the calculation result between the number of requested pages and the page size with the total number of records to determine whether to stop obtaining source data.</li> </ul>

Parameter	Description
<p>Pagination End Field Path</p>	<p>This parameter is mandatory only if <b>Paging</b> is enabled. Enter the path of the field in an API response, which is used to determine the end of pagination. In the API response, elements in different layers are separated by periods (.). For example, if element c in the {"a":{"b":{"c":"xxx"}}} response is the pagination end field, the pagination end field path is set to <b>a.b.c</b>.</p> <ul style="list-style-type: none"> <li>• If <b>Pagination End</b> is set to <b>Empty page list</b>, set this parameter to the root path of the list field.</li> <li>• If <b>Pagination End</b> is set to <b>Number of records</b>, set this parameter to the path of the total number of records.</li> </ul>
<p>Incremental Migration</p>	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
<p>Start Time Field</p>	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Enter the start time field originally defined in the API data source, for example, <b>startTime</b>. This parameter is carried when ROMA Connect sends an API request to the source, indicating that data following the specified field will be obtained.</p> <p>The start time field and end time field must be both entered in <b>Params</b> or <b>Body</b> of the <b>Request Parameters</b>.</p>
<p>End Time Field</p>	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Enter the end time field originally defined in the API data source, for example, <b>endTime</b>. This parameter is carried when ROMA Connect sends an API request to the source, indicating that data before the specified value will be obtained.</p>
<p>Time Zone</p>	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the API data source so that ROMA Connect can identify the data timestamp.</p>



Parameter	Description
<p>Timestamp Initial Value</p>	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p> <p>Assume that <b>Start Time Field</b> is <b>startTime</b>, <b>End Time Field</b> is <b>endTime</b>, <b>Timestamp Initial Value</b> is <b>2020-11-01 12:00:00</b>, <b>Compensation Period</b> is <b>0</b>, and <b>Period Settings</b> is <b>Default</b> for incremental collection. If the first scheduling time of the task is 2020-11-01 13:00:00, the data collected for the first time is that the value of <b>startTime</b> is greater than or equal to 2020-11-01 12:00:00 and the value of <b>endTime</b> is less than or equal to 2020-11-01 13:00:00. For subsequent collection, the data collected each time is that the value of <b>startTime</b> is greater than or equal to the time when the task is successfully executed last time and the value of <b>endTime</b> is less than or equal to the time. Execution time of the current task.</p>
<p>Compensation Period (ms)</p>	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
<p>Time Format</p>	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select a timestamp format, for example, <i>yyyy-MM-dd</i>.</p>

Parameter	Description
Period Settings	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the mode used for setting the time range for subsequent data integration after an incremental migration task is executed for the first time.</p> <ul style="list-style-type: none"> <li>• <b>Default:</b> Data generated between the previous scheduling and current scheduling is integrated. When ROMA Connect obtains data from the source, it uses the triggering time of the two tasks as the start time and end time, respectively.</li> <li>• <b>Custom:</b> The start time and end time are determined based on the configured period rules. This mode applies to common periodic tasks, for example, tasks executed once a day, a week, or a month.</li> </ul>
Start Time Offset (Days)	<p>This parameter is mandatory only if <b>Period Settings</b> is set to <b>Default</b>.</p> <p>Set the number of days before the start time of data collection.</p> <p>If data generated at the source changes in real time, such as alarm data, you can collect the data by setting this parameter.</p> <p>Start time of data collection = Data source system time – Start time offset</p>
Time Interval	<p>This parameter is mandatory only if <b>Period Settings</b> is set to <b>Custom</b>.</p> <p>Select the time granularity. The value must be the same as the unit configured in the <b>task schedule</b> so that the new data can be overwritten. For example, if <b>Unit</b> is set to <b>Day</b> in a task schedule, set this parameter to <b>Day</b>, indicating that data is obtained once a day.</p>
Period	<p>This parameter is mandatory only if <b>Period Settings</b> is set to <b>Custom</b>.</p> <p>Select the time period for obtaining source data. For example, if the task is executed once a day, <b>Time Interval</b> is set to <b>Day</b>, and <b>Period</b> is set to <b>Previous period</b>, data of the previous day is incrementally integrated once. If <b>Period</b> is set to <b>Current period</b>, data of the current day is incrementally integrated once.</p>
Right Periodic Boundary	<p>This parameter is mandatory only if <b>Period Settings</b> is set to <b>Custom</b>.</p> <p>This parameter specifies whether the end time is included in the time range for obtaining source data.</p> <ul style="list-style-type: none"> <li>• <b>Closed interval:</b> The end time is included.</li> <li>• <b>Open interval:</b> The end time is not included.</li> </ul>

Parameter	Description
Request Parameters	Construct the parameter definition of the API request, for example, the page number and page size fields must be carried in <b>Params</b> or <b>Body</b> . Set this parameter based on the definition of the API data source.
Parse	<p>If <b>Paging</b> is enabled, <b>Parse</b> is set to <b>Yes</b> by default and cannot be changed.</p> <p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Response Type	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Select the format that will be used for the response of an API request. The value can be <b>JSON</b> or <b>XML</b> . Ensure that the format is the same as the actual response format of the API.
Data Root Field	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON or XML format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a> .
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies each underlying key-value data element that is obtained from the source in JSON or XML format and needs to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the response.</li> <li>• <b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

#### Description on Metadata Parsing Path Configuration

- Data in JSON or XML format does not contain arrays:  
For example, in the following JSON data (similar to XML data), the complete path of element a is **a**, the complete path of element b is **a.b**,

the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON or XML format contains arrays:

For example, in the following JSON data (similar to XML data), the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.

- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The preceding data in JSON or XML format that contains arrays is used as an example. The following describes the configuration when the destination is API:

- In the example of pagination configuration, **pageNo** and **pageSize** are the pagination parameters of the API and need to be added to the **Request Parameters**.

**Figure 6-4** API pagination configuration example

\* Page Number Field ?  Value

\* Page Size Field ?  Records

Maximum Number of Pages  ?

Pagination End

Pagination End Field Path

\* Incremental Migration

Request Parameters

Key	Value	Operation
<input type="text" value="pageNo"/>	<input type="text"/>	Delete
<input type="text" value="pageSize"/>	<input type="text"/>	Delete

- In the example of incremental migration configuration, **startTime** and **endTime** are the time parameters of the API and need to be added to the **Request Parameters**.

**Figure 6-5** API incremental migration configuration example

The screenshot displays the configuration interface for an API incremental migration. It includes several input fields and a table for request parameters.

- Start Time Field:** startTime
- End Time Field:** endTime
- Time Zone:** GMT+08:00
- Timestamp Initial Value:** 2020/08/01 00:00:00
- Compensation Period (ms):** 0
- Time Format:** yyyy-MM-dd
- Period Settings:** Default

The **Request Parameters** section is expanded to show the **Params** tab, which contains the following table:

Key	Value
startTime	
endTime	

- In the example of metadata configuration, **Data Root Field** is set to **a**.

**Figure 6-6** API metadata configuration example

The screenshot displays the configuration interface for API metadata. It includes several input fields and a table for metadata.

- Parse:** Yes
- Response Type:** JSON
- Data Root Field:** a

The **Metadata** section contains the following table:

Alias	Type	Parsing Path
c	Integer	b[[]].c
d	String	b[[]].d

2. After configuring the source information, proceed with **Configuring Destination Information**.

## ActiveMQ

### [Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select ActiveMQ as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-6** ActiveMQ information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the ActiveMQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>ActiveMQ</b> .
Data Source Name	Select the ActiveMQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
Destination Type	Select the message transfer model of the ActiveMQ data source. The value can be <b>Topic</b> or <b>Queue</b> .
Destination Name	Enter the name of a topic or queue from which to obtain data. Ensure that the topic or queue already exists.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"><li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li><li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li></ul>
Data Root Field	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a> .
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination. <ul style="list-style-type: none"><li>• <b>Alias</b>: user-defined metadata name.</li><li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li><li>• <b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li></ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [
      {
        "c": "xx",
        "d": "xx"
      },
      {
        "c": "yy",
        "d": "yy"
      }
    ]
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:



- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The preceding JSON data that does not contain arrays is used as an example. The following describes the configuration when the source is ActiveMQ:

**Figure 6-7** ActiveMQ configuration example

* Destination Type	Topic		
* Destination Name	doctest		
* Parse	Yes		
Data Root Field	a		
* Metadata	Alias	Type	Parsing Path
	c	Integer	b.c
	d	String	b.d

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## ArtemisMQ

[Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select ArtemisMQ as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-7** ArtemisMQ information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the ArtemisMQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Data Source Type	Select <b>ArtemisMQ</b> .
Data Source Name	Select the ArtemisMQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
Destination Type	Select the message transfer model of the ArtemisMQ data source. The value can be <b>Topic</b> or <b>Queue</b> .
Destination Name	Enter the name of a topic or queue from which to obtain data. Ensure that the topic or queue already exists.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"> <li>If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Data Root Field	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a> .
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination. <ul style="list-style-type: none"> <li><b>Alias</b>: user-defined metadata name.</li> <li><b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li><b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d

are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.

- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The configuration when the source is ArtemisMQ is similar to that when the source is ActiveMQ. For details, see [ActiveMQ configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## DB2

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select DB2 as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-8** DB2 information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the DB2 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>DB2</b> .
Data Source Name	Select the DB2 data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the DB2 data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the DB2 data source so that ROMA Connect can identify the data timestamp.</p>

Parameter	Description
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE type as the timestamp, which can be used to check whether a data row meets incremental integration conditions. If the entered values of <b>Timestamp</b> field and <b>Timestamp Initial Value</b> are incomplete, full integration is used by default.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>
Extended Metadata	<p>This parameter is mandatory if the value of a field in the database is in JSON format and the underlying key-value data elements in the JSON field value need to be collected.</p> <ul style="list-style-type: none"> <li>● <b>Field Name:</b> Select the name of the data field whose sub-elements need to be collected in the table.</li> <li>● <b>Type:</b> data type of the data element to be collected in the JSON field value.</li> <li>● <b>Parsing Path:</b> complete path of the data element in the JSON field value. For details, see <a href="#">Description on Extended Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Extended Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [
      {
        "c": "xx",
        "d": "xx"
      },
      {
        "c": "yy",
        "d": "yy"
      }
    ]
  }
}
```

#### NOTE

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is DB2 is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## DWS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select DWS as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-9** DWS information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the DWS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>DWS</b> .
Data Source Name	Select the DWS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the DWS data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>



Parameter	Description
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the DWS data source so that ROMA Connect can identify the data timestamp.</p>
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>

Parameter	Description
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is DWS is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## DIS

[Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select DIS as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-10** DIS information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the DIS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>DIS</b> .
Data Source Name	Select the DIS data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Parse	<p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Data Root Field	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</p>
Data Type	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>Select the format of data obtained from the DIS data source. The data format must be the same as the actual data format stored in DIS.</p>
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>
Time Zone	<p>Select the time zone used by the DIS data source so that ROMA Connect can identify the data timestamp.</p>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
```

```

    "c": "xx",
    "d": "xx"
  }
}

```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```

{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}

```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The configuration when the source is DIS is similar to that when the source is ActiveMQ. For details, see [ActiveMQ configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## FTP

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select FTP as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-11** FTP information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the FTP data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FTP</b> .
Data Source Name	Select the FTP data source that you configured in <a href="#">Connecting to Data Sources</a> .
File Path	Enter the path of the folder to be accessed on the FTP server, for example, <b>/data/FDI</b> .
File Name	Enter the name of the data file from which data is to be obtained. If you do not specify this parameter, data of all files is obtained.
File Name Encoding	Select the encoding mode of the data file name.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"><li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li><li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li></ul>
Maximum File Size	Enter the maximum size of the file to be obtained. If the size of a file exceeds the value you specify here, the file cannot be obtained.
File Content Encoding	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Select the encoding format of the data file content.

Parameter	Description
File Separator	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Enter the field separator for the data file to distinguish different fields in each row of data.
Space Format Character	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Enter the space characters defined in the data file so that ROMA Connect can identify the space characters in the file content. For example, if the space format is defined as a period (.), it is considered as a space when the period (.) is read during data obtaining.
Skip File Header	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies whether to skip the file header in the data file. The file header is the first line or several lines at the beginning of a file, which helps identify and distinguish the file content.
Skipped Header Lines	This parameter is mandatory only if <b>Skip File Header</b> is set to <b>Yes</b> . Enter the number of lines in the file header in a data file so that ROMA Connect can skip the header lines and identify the start line of the data in the file.
Migrated Data Quantity	Set the amount of data to be obtained. <ul style="list-style-type: none"> <li>• If <b>Parse</b> is set to <b>Yes</b>, this parameter indicates the number of data rows to be parsed.</li> <li>• If <b>Parse</b> is set to <b>No</b>, this parameter indicates the number of files to be read.</li> </ul>
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each data field in a data file obtained from the source to be integrated to the destination. Metadata must be filled according to the sequence of the fields in the file. <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following describes the configuration when the source is FTP. The **id**, **name**, and **info** fields are obtained from the FTP data source and need to be integrated to the destination.

**Figure 6-8** FTP configuration example

The screenshot shows the FTP configuration interface with the following settings:

- File Path:** /data/test
- File Name:** sourcedata.csv
- File Name Encoding:** UTF-8
- Parse:** Yes
- Maximum File Size:** A slider is set to 115MB, with a range from 0MB to 200MB.
- File Content Encoding:** UTF-8
- File Separator:** ,
- Space Format Character:** Enter a space format character.
- Skip File Header:** No
- Migrated Data Quantity:** 1,500,000
- Metadata:**

Alias	Type
id	Integer
name	String
info	String

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## FI HDFS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select FI HDFS as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-12** FI HDFS information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the FI HDFS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FI HDFS</b> .

Parameter	Description
Data Source Name	Select the FI HDFS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Separator	This parameter specifies the user-defined separator.
Storage Subpath	This parameter specifies the subpath for storing data.
Storage Block Size (M)	Select the storage block size. The value can be <b>64</b> or <b>128</b> .
Storage Type	The text file type is supported.
Parse	<p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies each data field in a data file obtained from the source to be integrated to the destination. Metadata must be filled according to the sequence of the fields in the file.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The configuration when the source is FI HDFS is similar to that when the source is MRS HDFS. For details, see [MRS HDFS configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## FI Hive

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select FI Hive as the data source type at the source.

1. On the **Create Task** page, configure source information.



**Table 6-13** FI Hive information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the FI Hive data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FI Hive</b> .
Data Source Name	Select the FI Hive data source that you configured in <a href="#">Connecting to Data Sources</a> .
Database Name	Select the FI Hive database name that you configured.
Table	Select the table in the FI Hive database. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.
Incremental Migration	This parameter specifies whether only data generated in a specific period is integrated.  For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.
Timestamp	This parameter is mandatory only if <b>Incremental Migration</b> is enabled.  Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions. If the entered values of <b>Timestamp</b> field and <b>Timestamp Initial Value</b> are incomplete, full integration is used by default.
Time Zone	This parameter is mandatory only if <b>Incremental Migration</b> is enabled.  Select the time zone used by the FI Hive data source so that ROMA Connect can identify the data timestamp.
Timestamp Initial Value	This parameter is mandatory only if <b>Incremental Migration</b> is enabled.  This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.

Parameter	Description
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

The configuration when the source is FI Hive is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## FI Kafka

### [Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select FI Kafka as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-14** FI Kafka information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the FI Kafka data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FI Kafka</b> .

Parameter	Description
Data Source Name	Select the FI Kafka data source that you configured in <a href="#">Connecting to Data Sources</a> .
Topic Name	Enter a topic name that is created in the FI Kafka service and starts with T_.
Parse	<p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Data Root Field	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</p>
Data Type	Select the format of data obtained from the FI Kafka data source. The data format must be the same as the actual data format stored in FI Kafka. If <b>Parse</b> is set to <b>Yes</b> , you can only select <b>JSON</b> . If <b>Parse</b> is set to <b>No</b> , you can select either <b>JSON</b> or <b>Binary file</b> .
Offset	Select whether to integrate the earliest message data or the latest message data.
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>
Time Zone	Select the time zone used by the FI Kafka data source so that ROMA Connect can identify the data timestamp.

**Description on Metadata Parsing Path Configuration**

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [
      {
        "c": "xx",
        "d": "xx"
      },
      {
        "c": "yy",
        "d": "yy"
      }
    ]
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The configuration when the source is FI Kafka is similar to that when the source is Kafka. For details, see [Kafka configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## GaussDB 100

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select GaussDB 100 as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-15** Source GaussDB 100 information

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the GaussDB 100 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>GaussDB 100</b> .
Data Source Name	Select the GaussDB 100 data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Schema	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select a schema.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the GaussDB 100 data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>

Parameter	Description
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the GaussDB 100 data source so that ROMA Connect can identify the data timestamp.</p>
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is GaussDB 100 is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## GaussDB 200

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select GaussDB 200 as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-16** Source GaussDB 200 information

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the GaussDB 200 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>GaussDB 200</b> .
Data Source Name	Select the GaussDB 200 data source that you configured in <a href="#">Connecting to Data Sources</a> .



Parameter	Description
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the GaussDB 200 data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the GaussDB 200 data source so that ROMA Connect can identify the data timestamp.</p>

Parameter	Description
Timestamp	This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled. Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.
Timestamp Initial Value	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.
Compensation Period (ms)	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here. For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b> , the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).
Filter	This parameter is mandatory only if <b>Insert SQL</b> is disabled. Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination. For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is GaussDB 200 is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## HANA

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select HANA as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-17** HANA information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the HANA data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>HANA</b> .
Data Source Name	Select the HANA data source that you configured in <a href="#">Connecting to Data Sources</a> .
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>

Parameter	Description
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the HANA data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the HANA data source so that ROMA Connect can identify the data timestamp.</p>
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>

Parameter	Description
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is HANA is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## HL7

[Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select HL7 as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-18** HL7 information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the HL7 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>HL7</b> .
Data Source Name	Select the HL7 data source that you configured in <a href="#">Connecting to Data Sources</a> .
Encoding Format	Select the encoding mode of data files in the HL7 data source. The value can be <b>UTF-8</b> or <b>GBK</b> .
Metadata	<p>This parameter specifies the data fields in HL7 messages obtained from the source to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata.</li> <li>• <b>Parsing Path:</b> location of metadata in HL7 messages. For details, see the metadata path configuration description in the subsequent section.</li> </ul>

### Description on Metadata Path Configuration

```
MSH|^~\&|hl7Integration|hl7Integration||||ADT^A01|||2.3|
EVN|A01|20191212155644
PID||PATID1234^5^M11||FN^Patrick^^MR||19700101|1||xx Street^^NY^^Ox4DP|||||
NK1|1|FN^John^^MR|Father||999-9999
NK1|2|MN^Georgie^^MSS|Mother||999-9999
```

The metadata parsing path of HL7 messages must be set based on the Terser syntax specifications. In the preceding example HL7 message, each line represents an information segment. Each information segment starts with three uppercase letters, which are paragraph symbols of the information segment and are used to indicate content of the information segment. Information segments are separated by separators.

- |: field separator, which is used to divide information segments into different fields. Each field in an information segment is numbered starting from 1 (excluding paragraph symbols). The rest may be deduced by analogy.
- ^: component separator, which divides the field content into different components. In the fields that are divided into components, the position of a component is identified by a number, starting from 1. The rest may be deduced by analogy.
- ~: subcomponent separator, which is used to divide a component into subcomponents.

For example, in the PID information segment, if the field position of 19700101 is 7, the parsing path is **/PID-7**. If the field position of xx Street is 11 and the component position is 1, the parsing path is **/PID-11-1**.

For the information segments with the same paragraph symbol in the HL7 message, the repeated paragraph symbol is identified by adding a number enclosed by brackets after the paragraph symbol. In repeated paragraph symbols, the first is (0), the second is (1), and so on.

For example, in the NK1 information segment, "Father" is located in the first NK1 information segment, and a field location is 3, a parsing path of the NK1 information segment is **NK1(0)-3**. Similarly, the parsing path of Mother is **NK1(1)-3**.

Reading the 19700101 and xx Street fields in the preceding HL7 message is used as an example. The following describes the configuration when the source is HL7:

**Figure 6-9** HL7 configuration example

The screenshot shows a configuration interface for HL7. At the top, there is a dropdown menu for 'Encoding Format' set to 'UTF-8'. Below it is a 'Metadata' section with a table. The table has three columns: 'Alias', 'Type', and 'Parsing Path'. There are two rows of metadata: one for 'ID' with type 'Integer' and parsing path '/PID-7', and one for 'Address' with type 'String' and parsing path '/PID-11-1'. An 'Add' button is located at the bottom of the table.

Alias	Type	Parsing Path
ID	Integer	/PID-7
Address	String	/PID-11-1

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## Hive

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select Hive as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-19** Hive information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the Hive data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>Hive</b> .
Data Source Name	Select the Hive data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Database Name	Select the database in the Hive data source. <b>NOTE</b> You need to use a self-built database instead of the default database of Hive.
Table	Select the data table in the Hive data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.
Incremental Migration	This parameter specifies whether only data generated in a specific period is integrated. For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.
Timestamp	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.
Time Zone	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. Select the time zone used by the Hive data source so that ROMA Connect can identify the data timestamp.
Timestamp Initial Value	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.
Compensation Period (ms)	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here. For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b> , the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).



Parameter	Description
Filter	<p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is Hive is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## IBM MQ

[Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select IBM MQ as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-20** IBM MQ information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the IBM MQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>IBM MQ</b> .
Data Source Name	Select the IBM MQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
Destination Type	Select the message transfer model of the IBM MQ data source. The value can be <b>Topic</b> or <b>Queue</b> .

Parameter	Description
Destination Name	Enter the name of a topic or queue from which to obtain data. Ensure that the topic or queue already exists.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"> <li>If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Data Root Field	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a> .
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination. <ul style="list-style-type: none"> <li><b>Alias</b>: user-defined metadata name.</li> <li><b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li><b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:
- For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.
- The configuration when the source is IBM MQ is similar to that when the source is ActiveMQ. For details, see [ActiveMQ configuration example](#).
2. After configuring the source information, proceed with [Configuring Destination Information](#).

## Kafka

### [Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select Kafka as the data source type at the source. ROMA Connect MQS is a Kafka data source.

1. On the **Create Task** page, configure source information.

**Table 6-21** Kafka information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the Kafka data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>Kafka</b> .
Data Source Name	Select the Kafka data source that you configured in <a href="#">Connecting to Data Sources</a> .
Topic Name	Select the name of the topic whose data is to be obtained.
Parse	<p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Data Root Field	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path in Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</p>
Data Type	Select the format of data obtained from the Kafka data source. The data format must be the same as the actual data format stored in Kafka. If <b>Parse</b> is set to <b>Yes</b> , you can only select <b>JSON</b> . If <b>Parse</b> is set to <b>No</b> , you can select either <b>JSON</b> or <b>Binary file</b> .
Offset	Select whether to integrate the earliest message data or the latest message data.

Parameter	Description
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>. This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path:</b> path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>
Time Zone	Select the time zone used by the Kafka data source so that ROMA Connect can identify the data timestamp.

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The preceding JSON data that does not contain arrays is used as an example. The following describes the configuration when the source is Kafka:

**Figure 6-10** Kafka configuration example

The screenshot shows a configuration form for Kafka with the following fields and values:

- \* Topic Name:** dropdown menu with "topic-doc" selected.
- \* Parse:** dropdown menu with "Yes" selected.
- Data Root Field:** text input field containing "a".
- \* Data Type:** dropdown menu with "JSON" selected.
- Offset:** radio buttons for "Earliest" and "Latest", with "Latest" selected.
- \* Metadata:** a table with columns "Alias", "Type", and "Parsing Path".
 

Alias	Type	Parsing Path
c	Integer	b.c
d	String	b.d
- \* Time Zone:** dropdown menu with "GMT+08:00" selected.

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## LDAP

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select LDAP as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-22** LDAP information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the LDAP data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>LDAP</b> .
Data Source Name	Select the LDAP data source that you configured in <a href="#">Connecting to Data Sources</a> .
DN	This parameter specifies the distinguished name (DN) in the character string for connecting to the server, including Domain Component (DC), Common Name (CN), User ID (UID), and Organizational Unit (OU). For example, if <b>DN</b> is set to <b>CN=test,OU=roma,DC=domainname</b> , it indicates the test object in the roma organization of the domainname domain.
Filter Criteria	Filter fields based on the LDAP syntax. For example, <code>( (uid=user.1*)&amp;(createTimestamp&gt;={{begin}})(createTimestamp&lt;={{end}}))</code> indicates that objects whose UID starts with user.1 within a specified period are queried. <ul style="list-style-type: none"> <li>• <code>{{begin}}</code> indicates the time when the task was executed last time.</li> <li>• <code>{{end}}</code> indicates the time when the task is executed this time.</li> </ul>
Time Format	Select a time format for the filter criterion.
Timestamp Initial Value	This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.

Parameter	Description
Metadata	<p>This parameter specifies each data field in a data file obtained from the source to be integrated to the destination. Metadata must be filled according to the sequence of the fields in the file.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following describes the configuration when the source is LDAP. The **id**, **name**, and **info** fields are obtained from the LDAP data source and need to be integrated to the destination.

**Figure 6-11** LDAP configuration example

The screenshot shows a configuration interface for LDAP. It includes the following fields and a table:

- DN:** CN=test,OU=roma,DC=domainname
- Filter:** ((uid=user:1\*)&(createTimestamp>=\${b
- Filter Conditional Time Format:** Please select time format (dropdown menu)
- Timestamp Initial Value:** 2020/08/01 00:00:00 (with clear and calendar icons)
- Metadata Table:**

Alias	Type
id	Integer
name	String
info	String

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## MySQL

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select MySQL as the data source type at the source.

1. On the **Create Task** page, configure source information.



**Table 6-23** MySQL information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the MySQL data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MySQL</b> .
Data Source Name	Select the MySQL data source that you configured in <a href="#">Connecting to Data Sources</a> .
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the MySQL data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>

Parameter	Description
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the MySQL data source so that ROMA Connect can identify the data timestamp.</p>
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE type as the timestamp, which can be used to check whether a data line meets data integration conditions. If the entered values of <b>Timestamp</b> field and <b>Timestamp Initial Value</b> are incomplete, full integration is used by default.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>

Parameter	Description
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>
Extended Metadata	<p>This parameter is mandatory if the value of a field in the database is in JSON format and the underlying key-value data elements in the JSON field value need to be collected.</p> <ul style="list-style-type: none"> <li>• <b>Field Name:</b> Select the name of the data field whose sub-elements need to be collected in the table.</li> <li>• <b>Type:</b> data type of the data element to be collected in the JSON field value.</li> <li>• <b>Parsing Path:</b> complete path of the data element in the JSON field value. For details, see <a href="#">Description on Extended Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Extended Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
```

```

    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}

```

**NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The preceding JSON data that contains arrays is used as an example. The following describes the configuration when the destination is MySQL:

- In the example of incremental migration configuration, the data table must contain a field of the DATE, TIME, or TIMESTAMP type as the timestamp field.

**Figure 6-12** MySQL incremental migration configuration example

- In the example of extended metadata configuration, the child elements **c** and **d** are obtained from the **desc** field in the data table.

**Figure 6-13** MySQL extended metadata configuration example

Filter	⊕ Add Condition		
metaData	Field Name	Type	Parsing Path
	desc	integer	a.b.c
	desc	string	a.b.d
	⊕ Add		

2. After configuring the source information, proceed with **Configuring Destination Information**.

## MongoDB

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select MongoDB as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-24** MongoDB information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the MongoDB data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MongoDB</b> .
Data Source Name	Select the MongoDB data source that you configured in <a href="#">Connecting to Data Sources</a> .
Source Set	Select the data set to be obtained from the MongoDB data source. (The data set is equivalent to a data table in a relational database.) Then, click <b>Select Fields in Set</b> and select only the column fields that you want to integrate and synchronize.
Incremental Migration	This parameter specifies whether only data generated in a specific period is integrated.  For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.
Timestamp	This parameter is mandatory only if <b>Incremental Migration</b> is enabled.  Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.
Time Zone	This parameter is mandatory only if <b>Incremental Migration</b> is enabled.  Select the time zone used by the MongoDB data source so that ROMA Connect can identify the data timestamp.

Parameter	Description
Timestamp Initial Value	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.
Compensation Period (ms)	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here. For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b> , the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is MongoDB is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## MRS Hive

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select MRS Hive as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-25** MRS Hive information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.

Parameter	Description
Integration Application	Select the integration application to which the MRS Hive data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS Hive</b> .
Data Source Name	Select the MRS Hive data source that you configured in <a href="#">Connecting to Data Sources</a> .
Database Name	Select the database in the MRS Hive data source. <b>NOTE</b> You need to use a self-built database instead of the default database of MRS Hive.
Table	Select the data table from which data is to be obtained in the MRS Hive data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.
Incremental Migration	This parameter specifies whether only data generated in a specific period is integrated. For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.
Timestamp	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.
Time Zone	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. Select the time zone used by the MRS Hive data source so that ROMA Connect can identify the data timestamp.
Timestamp Initial Value	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.

Parameter	Description
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is MRS Hive is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## MRS HDFS

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select MRS HDFS as the data source type at the source.

1. On the **Create Task** page, configure source information.



**Table 6-26** MRS HDFS information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the MRS HDFS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS HDFS</b> .
Data Source Name	Select the MRS HDFS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Separator	Enter the field separator for the text data in the MRS HDFS data source. The separator is used to distinguish different fields in each row of data.
Storage Subpath	Enter the path of the data to be integrated in the hdfs:///hacluster directory of MRS HDFS.
Storage Block Size (M)	Select the size of data to be read each time ROMA Connect obtains data from the MRS HDFS data source.
Storage Type	Select the storage type of data in the MRS HDFS data source. The value must be the same as the actual data storage type of the MRS HDFS data source. Currently, only <b>Text file</b> is available.
Parse	<p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies each data field in text data obtained from the source to be integrated to the destination. Metadata must be filled according to the sequence of the fields in the file.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following describes the configuration when the source is MRS HDFS. The **id**, **name**, and **info** fields are obtained from the MRS HDFS data source and need to be integrated to the destination.

**Figure 6-14** MRS HDFS configuration example

* Separator	<input type="text" value=","/>								
* Storage Subpath	<input type="text" value="/data/test"/>								
* Storage Block Size (M)	<input type="text" value="64"/>								
* Storage Type	<input type="text" value="Text file"/>								
* Parse	<input type="text" value="Yes"/>								
* Metadata <span>?</span>	<table border="1"> <thead> <tr> <th>Alias <span>?</span></th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>Integer</td> </tr> <tr> <td>name</td> <td>String</td> </tr> <tr> <td>info</td> <td>String</td> </tr> </tbody> </table>	Alias <span>?</span>	Type	id	Integer	name	String	info	String
Alias <span>?</span>	Type								
id	Integer								
name	String								
info	String								

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## MRS HBase

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select MRS HBase as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-27** MRS HBase information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the MRS HBase data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS HBase</b> .
Data Source Name	Select the MRS HBase data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Table	Select the data table in the MRS HBase data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## MRS Kafka

### [Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select MRS Kafka as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-28** MRS Kafka information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the MRS Kafka data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS Kafka</b> .
Data Source Name	Select the MRS Kafka data source that you configured in <a href="#">Connecting to Data Sources</a> .
Topic Name	Enter a topic name that is created in the MRS Kafka service and starts with T_.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>

Parameter	Description
Data Root Field	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Path Configuration</a> .
Data Type	Select the format of data obtained from the MRS Kafka data source. The data format must be the same as the actual data format stored in MRS Kafka. If <b>Parse</b> is set to <b>Yes</b> , you can only select <b>JSON</b> . If <b>Parse</b> is set to <b>No</b> , you can select either <b>JSON</b> or <b>Binary file</b> .
Offset	Select whether to integrate the earliest message data or the latest message data.
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination. <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path</b>: path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>
Time Zone	Select the time zone used by the Kafka data source so that ROMA Connect can identify the data timestamp.

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.
- The configuration when the source is MRS Kafka is similar to that when the source is Kafka. For details, see [Kafka configuration example](#).
2. After configuring the source information, proceed with [Configuring Destination Information](#).

## OBS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select OBS as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-29** OBS information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the OBS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>OBS</b> .
Data Source Name	Select the OBS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Path	Enter the name of the object from which data is to be obtained in the OBS data source. The value of <b>Path</b> cannot end with a slash (/).
File Name Prefix	Enter a file name prefix. This parameter is used together with <b>Time Format</b> to filter the data files to be integrated.
Time Format	Select the time format to be used in the file name. This parameter is used together with <b>File Name Prefix</b> to filter the data files to be integrated.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the files based on the default sequence of OBS. By default, the file with the latest time is parsed first and then integrated into the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
File Type	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Select the format of the data file obtained from the OBS data source. The value can be <b>txt</b> , <b>csv</b> , or <b>zip</b> .
Field Separator	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Enter the field separator for the data file to distinguish different fields in each row of data.

Parameter	Description
Encoding Format	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . Select the encoding mode of data files obtained from the OBS data source. The value can be <b>UTF-8</b> or <b>GBK</b> .
Skip Title	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies whether to skip the title lines in the data file. The title is the first line or several lines at the beginning of a file, which helps identify and distinguish the file content.
Title Lines	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> and <b>Skip Title</b> is set to <b>Yes</b> . Enter the number of rows in the title information in the data file so that ROMA Connect can identify the start row of the data in the file.
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each data field in a data file obtained from the source to be integrated to the destination. Metadata must be filled according to the sequence of the fields in the file. <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following describes the configuration when the source is OBS. The **id**, **name**, and **info** fields are obtained from the OBS data source and need to be integrated to the destination.

**Figure 6-15** OBS configuration example

The screenshot shows the OBS configuration interface with the following settings:

- Path:** data/test
- File Name Prefix:** roma
- Time Format:** yyyy-MM-dd
- Parse:** Yes
- File Type:** txt (Note: Parses txt files that comply with the csv format.)
- Field Separator:** ,
- Encoding Format:** UTF-8
- Skip Title:** No
- Metadata:** Text Mode

Alias	Type
id	Double
name	String
info	String

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## Oracle

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select Oracle as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-30** Oracle information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the Oracle data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .



Parameter	Description
Data Source Type	Select <b>Oracle</b> .
Data Source Name	Select the Oracle data source that you configured in <a href="#">Connecting to Data Sources</a> .
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a <b>SELECT</b> statement and contain the <b>WHERE</b> condition. The <b>INSERT</b>, <b>UPDATE</b>, <b>DELETE</b>, and <b>DROP</b> statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the Oracle data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>

Parameter	Description
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the Oracle data source so that ROMA Connect can identify the data timestamp.</p>
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

Parameter	Description
Extended Metadata	<p>This parameter is mandatory if the value of a field in the database is in JSON format and the underlying key-value data elements in the JSON field value need to be collected.</p> <ul style="list-style-type: none"> <li>• <b>Field Name:</b> Select the name of the data field whose sub-elements need to be collected in the table.</li> <li>• <b>Type:</b> data type of the data element to be collected in the JSON field value.</li> <li>• <b>Parsing Path:</b> complete path of the data element in the JSON field value. For details, see <a href="#">Description on Extended Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Extended Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    },
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

 **NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is Oracle is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## PostgreSQL

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select PostgreSQL as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-31** PostgreSQL information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the PostgreSQL data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>PostgreSQL</b> .
Data Source Name	Select the PostgreSQL data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a <b>SELECT</b> statement and contain the <b>WHERE</b> condition. The <b>INSERT</b>, <b>UPDATE</b>, <b>DELETE</b>, and <b>DROP</b> statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the PostgreSQL data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the PostgreSQL data source so that ROMA Connect can identify the data timestamp.</p>

Parameter	Description
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>
Extended Metadata	<p>This parameter is mandatory if the value of a field in the database is in JSON format and the underlying key-value data elements in the JSON field value need to be collected.</p> <ul style="list-style-type: none"> <li>● <b>Field Name:</b> Select the name of the data field whose sub-elements need to be collected in the table.</li> <li>● <b>Type:</b> data type of the data element to be collected in the JSON field value.</li> <li>● <b>Parsing Path:</b> complete path of the data element in the JSON field value. For details, see <a href="#">Description on Extended Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Extended Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [
      {
        "c": "xx",
        "d": "xx"
      },
      {
        "c": "yy",
        "d": "yy"
      }
    ]
  }
}
```

#### NOTE

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is PostgreSQL is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## RabbitMQ

[Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select RabbitMQ as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-32** RabbitMQ information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the RabbitMQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>RabbitMQ</b> .
Data Source Name	Select the RabbitMQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
New Queue Creation	This parameter specifies whether to create a queue in the RabbitMQ data source. <ul style="list-style-type: none"> <li>• If this parameter is set to <b>Yes</b>, a new queue is created and data is obtained from the new queue.</li> <li>• If this parameter is set to <b>No</b>, data is obtained from the existing queue.</li> </ul>
Exchange Mode	This parameter is mandatory only if <b>New Queue Creation</b> is set to <b>Yes</b> . Select a routing mode for the exchange in RabbitMQ to forward messages to the new queue. <ul style="list-style-type: none"> <li>• <b>Direct</b>: If the routing key in a message fully matches the queue, the message will be forwarded to the queue.</li> <li>• <b>Topic</b>: If the routing key in a message approximately matches the queue, the message will be forwarded to the queue.</li> <li>• <b>Fanout</b>: All messages will be forwarded to the queue.</li> <li>• <b>Headers</b>: If the Headers attribute of a message fully matches the queue, the message will be forwarded to the queue.</li> </ul>
Exchange Name	This parameter is mandatory only if <b>New Queue Creation</b> is set to <b>Yes</b> . Enter the exchange name of the new queue in RabbitMQ.



Parameter	Description
Routing Key	<p>This parameter is mandatory only if <b>Exchange Mode</b> is set to <b>Direct</b> or <b>Topic</b>.</p> <p>Enter the routing key of the new queue. RabbitMQ uses the routing key as a condition and forwards messages that meet the condition to the new queue.</p>
Message Parameters	<p>This parameter is mandatory only if <b>Exchange Mode</b> is set to <b>Headers</b>.</p> <p>Enter the Headers key-value pair of the new queue. RabbitMQ uses Headers as a condition and forwards messages that meet the condition to the new queue.</p>
Queue Name	<p>Enter the name of the message queue whose data is to be obtained.</p> <ul style="list-style-type: none"> <li>• If <b>New Queue Creation</b> is set to <b>Yes</b>, enter a new queue name.</li> <li>• If <b>New Queue Creation</b> is set to <b>No</b>, enter the name of an existing queue in the RabbitMQ data source.</li> </ul>
Automatic Deletion	<p>This parameter specifies whether a queue will be automatically deleted if no client is connected.</p>
Persistence	<p>This parameter specifies whether messages in a queue are stored permanently.</p>
Parse	<p>This parameter specifies whether ROMA Connect parses the obtained source data.</p> <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>.</p> <p>This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path</b>: full path of metadata. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [
      {
        "c": "xx",
        "d": "xx"
      },
      {
        "c": "yy",
        "d": "yy"
      }
    ]
  }
}
```

The preceding JSON data that does not contain arrays is used as an example. The following describes the configuration when the source is RabbitMQ:

**Figure 6-16** RabbitMQ configuration example

The screenshot shows a configuration form for a new queue. The settings are as follows:

- New Queue Creation:** No
- Queue Name:** doctest
- Automatic Deletion:** Yes
- Persistency:** Yes
- Parse:** Yes
- Metadata:**

Alias	Type	Parsing Path
c	Integer	a.b.c
d	String	a.b.d

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## SAP

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select SAP as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-33** SAP information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the SAP data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>SAP</b> .
Data Source Name	Select the SAP data source that you configured in <a href="#">Connecting to Data Sources</a> .
Available Function	<p>This parameter specifies the functions that are opened by the SAP data source for remote invoking.</p> <p>Click <b>Query</b>. In the <b>Available Functions</b> dialog box, enter the function name and group name to query and obtain the list of available functions of the SAP data source. In the <b>Function List</b> area, select the function you want to use and click <b>OK</b>.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• The available functions can be queried on ROMA Connect only if the function that allows remote access has been enabled in the SAP data source.</li> <li>• The function name and group name support fuzzy search by prefix.</li> </ul>
Function Name	Enter the name of the SAP function selected in <b>Available Functions</b> .

Parameter	Description
Request Parameters	<p>The request parameters of the selected source function are automatically displayed. Enter the value of each request parameter based on the site requirements. The source function returns the corresponding output parameter based on the request parameter.</p> <p>If the request parameter is a table-type parameter, only the first row of the parameter table is displayed. You can click <b>Add</b> on the right of the parameter to add parameters in other rows of the parameter table. For details, see <b>Description on Configuration of Table-type Request Parameters</b>.</p>
Metadata	<p>The output parameters (metadata) of the selected source function are automatically displayed. The metadata alias is automatically generated based on the parsing path. You can edit the metadata alias based on the site requirements. You can also delete unnecessary metadata and retain only the metadata that needs to be integrated to the destination.</p>

**Description on Configuration of Table-type Request Parameters:**

The following request parameter structure is used as an example. [] indicates the table-type data, and {} indicates a row of data in the table. There are three rows of data in a parameter table, and each row of data contains two parameters.

The key of parameter b is **a[i].b**, and the key of parameter c is **a[i].c**. i indicates the row number of the parameter in the parameter table. The value 0 indicates the first row, the value 1 indicates the second row, and so on.

```
a: [
  {b: xx, c: xx},
  {b: yy, c: yy},
  {b: zz, c: zz},
]
```

After you select an available function, only the parameters in the first row of the a data table are displayed, that is, **a[0].b** and **a[0].c**. If you need to add the parameters of a row in the data table, add the parameters row by row in sequence. Ensure that the parameters of the previous row have been added. You are not allowed to add the parameters of only row 1 and row 3 without adding the parameters of row 2.

For example, in the preceding example, if you want to add the parameter in the second row, add **a[1].b** or **a[1].c** after **a[0].c**. To add the parameter in the third row, add at least one of parameters **a[1].b** and **a[1].c** in the second row after **a[0].c**, and then add parameter **a[2].b** or **a[2].c** in the third row.

**NOTICE**

The same parameter in the same row of data cannot be added repeatedly. Otherwise, the data integration task fails to be executed.

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## SNMP

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select SNMP as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-34** SNMP information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the SNMP data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>SNMP</b> .
Data Source Name	Select the SNMP data source that you configured in <a href="#">Connecting to Data Sources</a> .
Data Mode	Select a data mode from the SNMP data source. <ul style="list-style-type: none"> <li>• If you select <b>Row</b> for <b>Data Mode</b>, SNMP data is parsed by row.</li> <li>• If you select <b>Column</b> for <b>Data Mode</b>, SNMP data is parsed by column.</li> </ul>
Object Identifier	This parameter is mandatory only if <b>Data Mode</b> is set to <b>Column</b> . Enter the OID of the object to be obtained from the SNMP message. Separate multiple OIDs with commas (,).
Metadata	This parameter specifies the object (data field) to be integrated to the destination from the SNMP message data obtained from the source. <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata.</li> <li>• <b>Parsing Path</b>: OID of metadata in the SNMP message. This parameter is mandatory only if <b>Data Mode</b> is set to <b>Row</b>.</li> </ul>

The following describes the configuration when the destination is SNMP:

- When **Data Model** is set to **Row**, **user** indicates obtaining system contact information and **location** indicates obtaining server location information.

**Figure 6-17** Configuration example when the data model is Row

* Data Mode	Row		
* Metadata	Alias ?	Type	Parsing Path ?
	user	String	.1.3.6.1.2.1.1.4.0
	location	String	.1.3.6.1.2.1.1.6.0

- When **Data Model** is set to **Column**, the **info** column contains the system contact information and server location information.

* Data Mode	Column		
* Object Identifier	.1.3.6.1.2.1.1.4.0,.1.3.6.1.2.1.1.6.0		
* Metadata	Alias ?	Type	
	info	String	

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## SQL Server

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select SQL Server as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-35** SQL Server information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the SQL Server data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>SQL Server</b> .
Data Source Name	Select the SQL Server data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter needs to be set only when <b>Insert SQL</b> is disabled.</p> <p>Select the data table in the SQL Server data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the SQL Server data source so that ROMA Connect can identify the data timestamp.</p>

Parameter	Description
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE, TIME, or TIMESTAMP type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be integrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>
Extended Metadata	<p>This parameter is mandatory if the value of a field in the database is in JSON format and the underlying key-value data elements in the JSON field value need to be collected.</p> <ul style="list-style-type: none"> <li>• <b>Field Name:</b> Select the name of the data field whose sub-elements need to be collected in the table.</li> <li>• <b>Type:</b> data type of the data element to be collected in the JSON field value.</li> <li>• <b>Parsing Path:</b> complete path of the data element in the JSON field value. For details, see <a href="#">Description on Extended Metadata Parsing Path Configuration</a>.</li> </ul>



### Description on Extended Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [
      {
        "c": "xx",
        "d": "xx"
      },
      {
        "c": "yy",
        "d": "yy"
      }
    ]
  }
}
```

#### NOTE

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The configuration when the source is SQL Server is similar to that when the source is MySQL. For details, see [MySQL configuration example](#).

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## TaurusDB

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select TaurusDB as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-36** TaurusDB information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the TaurusDB data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>TaurusDB</b> .
Data Source Name	Select the TaurusDB data source that you configured in <a href="#">Connecting to Data Sources</a> .
Insert SQL	<p>This parameter specifies whether SQL statements are used to obtain source data.</p> <ul style="list-style-type: none"> <li>• If <b>Insert SQL</b> is enabled, ROMA Connect obtains source data based on the entered SQL statements.</li> <li>• If <b>Insert SQL</b> is disabled, ROMA Connect obtains source data based on the conditions configured on the GUI.</li> </ul> <p>If <b>Insert SQL</b> is enabled, you need to enter an SQL statement for querying data. The statement must be a SELECT statement and contain the WHERE condition. The INSERT, UPDATE, DELETE, and DROP statements cannot be used. Click <b>Check SQL</b> to check the validity of the statement.</p> <p>For example, if you enter the statement <b>SELECT col01, col02 FROM table01 WHERE col02 IN('A', 'B', 'C') ORDER BY col01</b>, ROMA Connect selects the col01 and col02 columns from table01, filters data rows whose values are A, B, or C in the col02 column, and sorts the data rows in sequence based on the values in the col01 column.</p>
Table	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Select the data table from which data is to be obtained in the TaurusDB data source. Then, click <b>Select Table Field</b> and select only the column fields that you want to integrate and synchronize.</p>
Field Sorting	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>This parameter specifies how the data to be integrated is sorted by field. To sort the integrated data, select a reference field and select <b>Ascending</b> or <b>Descending</b>.</p>

Parameter	Description
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>
Time Zone	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>Select the time zone used by the TaurusDB data source so that ROMA Connect can identify the data timestamp.</p>
Timestamp	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled and <b>Insert SQL</b> is disabled.</p> <p>Select a field of the DATE type in the data table as the timestamp of source data to determine whether the data meets the incremental integration conditions.</p>
Timestamp Initial Value	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be migrated.</p>
Compensation Period (ms)	<p>This parameter is mandatory only if <b>Incremental Migration</b> is enabled.</p> <p>This parameter specifies the period of time (in milliseconds) which will be used to compensate for any delay in data generation at the source when ROMA Connect queries incremental data. The end time for obtaining data is the current system time minus the value you specify here.</p> <p>For example, if the end time of the previous incremental migration task is 15:05, the current scheduled task is triggered at 17:00, and <b>Compensation Period (ms)</b> is set to <b>100</b>, the time range of data to be integrated in the current incremental migration task is 15:05 to (17:00 - 100 ms).</p>
Filter	<p>This parameter is mandatory only if <b>Insert SQL</b> is disabled.</p> <p>Add filter criteria for data to be integrated. Only the source data that meets the filter criteria will be integrated to the destination.</p> <p>For example, the condition "and   col02   equal   A" means that only the data rows whose values are A in the col02 column will be integrated.</p>

Parameter	Description
Extended Metadata	<p>This parameter is mandatory if the value of a field in the database is in JSON format and the underlying key-value data elements in the JSON field value need to be collected.</p> <ul style="list-style-type: none"> <li>• <b>Field Name:</b> Select the name of the data field whose sub-elements need to be collected in the table.</li> <li>• <b>Type:</b> data type of the data element to be collected in the JSON field value.</li> <li>• <b>Parsing Path:</b> complete path of the data element in the JSON field value. For details, see <a href="#">Description on Extended Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Extended Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    },
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

**NOTE**

- Incremental migration does not support synchronization of physical deletion operations on source data tables to the destination. Logical deletion is recommended at the source.
- For data changes at the source, the timestamp of data rows needs to be updated synchronously. ROMA Connect compares the timestamp with the task execution time to identify the data to be incrementally migrated.

The preceding JSON data that contains arrays is used as an example. The following describes the configuration when the destination is TaurusDB:

- In the example of incremental migration configuration, the data table must contain a field of the DATE type as the timestamp field.

**Figure 6-18** TaurusDB incremental migration configuration example

- In the example of extended metadata configuration, the child elements **c** and **d** are obtained from the **desc** field in the data table.

**Figure 6-19** TaurusDB extended metadata configuration example

Field Name	Type	Parsing Path
desc	integer	a.b.c
desc	string	a.b.d

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## WebSocket

### [Back to Overview](#)

If **Integration Mode** is set to **Real-Time**, you can select WebSocket as the data source type at the source.

1. On the **Create Task** page, configure source information.

**Table 6-37** WebSocket information at the source

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the WebSocket data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>WebSocket</b> .
Data Source Name	Select the WebSocket data source that you configured in <a href="#">Connecting to Data Sources</a> .
Data Root Field	This parameter specifies the path of the upper-layer common fields among all metadata in the data obtained from the source in JSON format. <b>Data Root Field</b> and <b>Parsing Path</b> in <b>Metadata</b> form a complete metadata path. For details, see <a href="#">Description on Metadata Path Configuration</a> .
Heartbeat Mechanism	This parameter specifies whether the heartbeat mechanism is enabled between ROMA Connect and the WebSocket data source to ensure connection validity.
Heartbeat Message	This parameter is mandatory only if <b>Heartbeat Mechanism</b> is set to <b>Yes</b> . This parameter specifies the message content carried in heartbeat packets sent by ROMA Connect to the WebSocket data source.
Heartbeat Sending Interval	This parameter is mandatory only if <b>Heartbeat Mechanism</b> is set to <b>Yes</b> . This parameter specifies the interval for the ROMA Connect to send heartbeat packets to the WebSocket data source.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>• If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>

Parameter	Description
Metadata	<p>This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b>. This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path:</b> path of the metadata, which does not contain the data root field. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.
  - **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b.c**, and **Parsing Path** of element d must be set to **b.d**.
  - **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **c**, and **Parsing Path** of element d must be set to **d**.
- Data in JSON format contains arrays:
- For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c

and d are underlying data elements, that is, the data to be integrated to the destination.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  }
}
```

In this scenario, three configuration solutions are available for **Data Root Field** and **Parsing Path**:

- **Data Root Field** is not specified.  
**Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.
- **Data Root Field** is set to **a**.  
**Parsing Path** starts from the underlying path of element a. **Parsing Path** of metadata c must be set to **b[i].c**, and **Parsing Path** of element d must be set to **b[i].d**.
- **Data Root Field** is set to **a.b**.  
**Parsing Path** starts from the underlying path of element b. **Parsing Path** of metadata c must be set to **[i].c**, and **Parsing Path** of element d must be set to **[i].d**.

The preceding JSON data that does not contain arrays is used as an example. The following describes the configuration when the source is WebSocket:

**Figure 6-20** WebSocket configuration example

Data Root Field <span>?</span>	<input type="text" value="a"/>		
* Heartbeat Mechanism	<input type="text" value="Yes"/>		
* Heartbeat Message	<input type="text" value="request"/>		
* Heartbeat Sending Interval	<input type="text" value="1"/>		
* Parse <span>?</span>	<input type="text" value="Yes"/>		
* Metadata	Alias <span>?</span>	Type	Parsing Path <span>?</span>
	c	Integer	b.c
	d	String	b.d

2. After configuring the source information, proceed with [Configuring Destination Information](#).

## Custom

[Back to Overview](#)



If **Integration Mode** is set to **Scheduled**, you can select a custom data source at the source. ROMA Connect communicates with custom data sources through standard RESTful APIs.

1. On the **Create Task** page, configure source information.

**Table 6-38** Custom data source information

Parameter	Description
Instance	Select the ROMA Connect instance that is being used.
Integration Application	Select the integration application to which the custom data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select a custom data source type.
Data Source Name	Select the custom data source that you configured in <a href="#">Connecting to Data Sources</a> .
Paging	<p>Specifies whether data is returned on multiple pages when ROMA Connect sends a request to the custom data source to obtain data. Multiple data records can be returned for one API request.</p> <ul style="list-style-type: none"> <li>• If <b>Paging</b> is enabled, all data that meets the conditions is displayed on multiple pages based on a fixed number of records on each page. ROMA Connect obtains data on one page each time and obtains all data by sending multiple requests.</li> <li>• If <b>Paging</b> is disabled, ROMA Connect obtains all data that meets the conditions through one API request.</li> </ul>
Start Page	This parameter is mandatory only if <b>Paging</b> is enabled. Enter the start page number for paging. The value can be 0 or 1. Set this parameter based on the original definition of the API.
Page Size	This parameter is mandatory only if <b>Paging</b> is enabled. Enter the maximum number of data records on each page. Set this parameter based on the original definition of the API.
Incremental Migration	<p>This parameter specifies whether only data generated in a specific period is integrated.</p> <p>For the first scheduling, the data between the initial timestamp and the current scheduling time is collected. For subsequent scheduling, the data between the last successful collection time and the current time is collected.</p>

Parameter	Description
Time Zone	Select the time zone used by the custom data source so that ROMA Connect can identify the data timestamp.
Start Time	This parameter is mandatory only if <b>Incremental Migration</b> is enabled. This parameter specifies the time at which data is to be migrated for the first time. That is, only the data generated after this time point will be integrated.
Parse	This parameter specifies whether ROMA Connect parses the obtained source data. <ul style="list-style-type: none"> <li>If you select <b>Yes</b>, ROMA Connect parses the obtained source data based on the configured parsing rules and then integrates the data to the destination.</li> <li>If you select <b>No</b>, ROMA Connect transparently transmits the obtained source data and integrates the data to the destination.</li> </ul>
Metadata	This parameter is mandatory only if <b>Parse</b> is set to <b>Yes</b> . This parameter specifies each underlying key-value data element that is obtained from the source in JSON format and needs to be integrated to the destination. <ul style="list-style-type: none"> <li><b>Alias</b>: user-defined metadata name.</li> <li><b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the response.</li> <li><b>Parsing Path</b>: full path of metadata. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>
<p>In addition to the preceding parameters, different custom data sources define different reader parameters. Set the parameters based on the original definition specifications of the connector. You can locate the connector used by the custom data source on the <b>Assets</b> page of the ROMA Connect console and view the reader parameter definition of the connector.</p>	

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, and the complete path of element d is **a.b.d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
```

```

    "c": "xx",
    "d": "xx"
  }
}

```

- Data in JSON format contains arrays:

For example, in the following JSON data, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements, that is, the data to be integrated to the destination.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```

{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}

```

The preceding JSON data that does not contain arrays is used as an example. The following figure describes the configuration when the source is a custom data source. In the example, the value of **key** is the reader parameter defined in the connector.

**Figure 6-21** Custom data source configuration example

Alias	Type	Parsing Path
c	Integer	a.b.c
d	String	a.b.d

2. After configuring the source information, proceed with **Configuring Destination Information**.

## 6.3.3 Configuring Destination Information

### Overview

This topic describes how to configure destination information for a data integration task. Based on the destination information (including the data source and data storage), ROMA Connect writes data to the destination. The destination information configuration varies depending on data source types.

 **NOTE**

During data migration, if a primary key conflict occurs at the destination, data is automatically updated based on the primary key.

Data Source Types Supported by Both Real-Time and Scheduled Integration Tasks		Data Source Types Supported by Only Real-Time Tasks
<ul style="list-style-type: none"> <li>• <a href="#">API</a></li> <li>• <a href="#">ActiveMQ</a></li> <li>• <a href="#">ArtemisMQ</a></li> <li>• <a href="#">DB2</a></li> <li>• <a href="#">DIS</a></li> <li>• <a href="#">DWS</a></li> <li>• <a href="#">FI HDFS</a></li> <li>• <a href="#">FI Hive</a></li> <li>• <a href="#">FI Kafka</a></li> <li>• <a href="#">GaussDB 100</a></li> <li>• <a href="#">GaussDB 200</a></li> <li>• <a href="#">HL7</a></li> <li>• <a href="#">HANA</a></li> <li>• <a href="#">IBM MQ</a></li> <li>• <a href="#">Kafka</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">MySQL</a></li> <li>• <a href="#">MongoDB</a></li> <li>• <a href="#">MRS Hive</a></li> <li>• <a href="#">MRS HDFS</a></li> <li>• <a href="#">MRS HBase</a></li> <li>• <a href="#">MRS Kafka</a></li> <li>• <a href="#">Oracle</a></li> <li>• <a href="#">PostgreSQL</a></li> <li>• <a href="#">Redis</a></li> <li>• <a href="#">RabbitMQ</a></li> <li>• <a href="#">SQL Server</a></li> <li>• <a href="#">TaurusDB</a></li> <li>• <a href="#">Custom Data Sources</a></li> </ul>	<ul style="list-style-type: none"> <li>• <a href="#">FTP</a></li> <li>• <a href="#">OBS</a></li> </ul>

### API

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select API as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-39** API information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the API data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>API</b> .
Data Source Name	Select the API data source that you configured in <a href="#">Connecting to Data Sources</a> .
Request Parameters	<p>Construct the parameter definition of an API request. For example, the data to be integrated to the destination must be defined in <b>Body</b>. Set this parameter based on the definition of the API data source.</p> <ul style="list-style-type: none"> <li>• <b>Params</b>: parameters defined after the question mark (?) in the request URL. Only fixed values can be transferred. The method for setting <b>Params</b> is similar to that for setting <b>Body</b> in <a href="#">form-data</a> format.</li> <li>• <b>Headers</b>: message headers of RESTful requests. Only fixed values can be transferred. The method for setting <b>Headers</b> is similar to that for setting <b>Body</b> in <a href="#">form-data</a> format.</li> <li>• <b>Body</b>: bottom-layer parameters in the body of RESTful requests. This parameter and <b>Data Root Field</b> constitute the complete body of a request sent to the destination API. Source data is transferred to the destination API through parameters defined in <b>Body</b>. <b>Body</b> supports two modes: <b>form-data</b> and <b>raw</b>. For details, see <a href="#">Description on Body Parameter Configuration</a>.</li> </ul>
Data Root Field	<p>This parameter specifies the path of upper-layer common fields in all parameters in the body sent to the destination in JSON format. <b>Data Root Field</b> and <b>Body</b> in <b>Request Parameters</b> form the request body sent to the destination API.</p> <p>For example, if the body parameter is {"c":"xx","d":"xx"} and <b>Data Root Field</b> is set to <b>a.b</b>, the encapsulated request data is {"a":{"b":{"c":"xx","d":"xx"}}}.</p>

**Description on Body Parameter Configuration**

- form-data mode:

Set **Key** to the parameter name defined by the API data source and leave **Value** unspecified. The key will be used as the destination field name in mapping information to map and transfer the value of the source field.

**Figure 6-22** form-data mode

Key	Value	Operation
c		Delete
d		Delete

– Raw mode:

The raw mode supports the JSON, Array, and nested JSON formats. Enter an example body sent to the destination API in JSON format. ROMA Connect replaces the parameter values in the example based on the mapping configuration, and finally transfers the source data to the destination. The following is an example body in raw mode:

▪ JSON format:

```
{
  "id": 1,
  "name": "name1"
}
```

Enter the body content in JSON format , leave **Data Root Field** empty, and set the field names in **Mapping Information**.

Mapping Information

\* Mapping Rule

Text Mode

Ensure that the field type of the source matches that of the destination. Otherwise, an error may occur during type conversion.

Destination Field	Source Field
id	id
name	name

▪ Array format:

```
{
  "record": [
    {
      "id": 1,
      "name": ""
    }
  ]
}
```

```
}  
]  
}
```

Set **Data Root Field** to the JSONArray object name, for example, **record**. Enter field names in mapping information.

The screenshot shows the 'Request Parameters' section with the 'Body' tab selected. The 'form-data' radio button is selected, and the 'raw' option is chosen. A red box highlights the JSON body content: 

```
{  
  "record": [  
    {  
      "id": 1,  
      "name": ""  
    }  
  ]  
}
```

Below the body, the 'Data Root Field' is set to 'record' in a text input field, also highlighted with a red box.

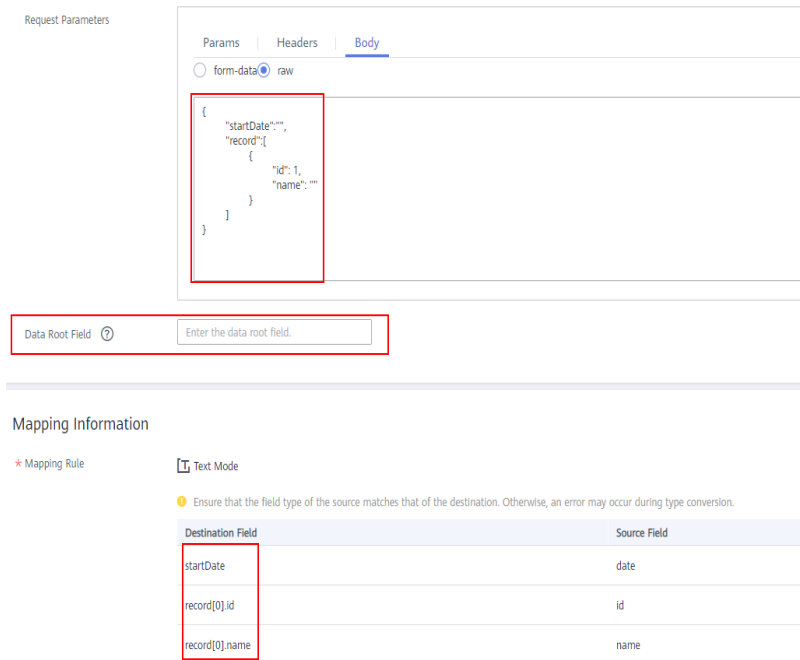
The 'Mapping Information' section is visible below, showing a table with 'Destination Field' and 'Source Field' columns. The 'id' and 'name' fields are mapped, with 'id' and 'name' in the destination field column and 'id' and 'name' in the source field column. A red box highlights the 'id' and 'name' rows in the destination field column.

Destination Field	Source Field
id	id
name	name

■ **Nested JSON format:**

```
{  
  "startDate": "",  
  "record": [  
    {  
      "id": 1,  
      "name": ""  
    }  
  ]  
}
```

Leave **Data Root Field** blank. In mapping information, set the json fields to the field names and set the jsonArray fields to specific paths, for example, **record[0].id**.



2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## ActiveMQ

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select ActiveMQ as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-40** ActiveMQ information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the ActiveMQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>ActiveMQ</b> .
Data Source Name	Select the ActiveMQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
Destination Type	Select the message transfer model of the ActiveMQ data source. The value can be <b>Topic</b> or <b>Queue</b> .



Parameter	Description
Destination Name	Enter the name of a topic or queue to which to send data. Ensure that the topic or queue already exists.
Metadata	<p>Define each underlying key-value data element to be written to the destination in JSON format. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path:</b> full path of metadata. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, the complete path of element d is **a.b.d**, and elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    },
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

The preceding JSON data that does not contain arrays is used as an example. The following describes the configuration when the destination is ActiveMQ:

**Figure 6-23** ActiveMQ configuration example

The screenshot shows a configuration form for ActiveMQ. It includes a dropdown for 'Destination Type' set to 'Topic', a text input for 'Destination Name' with the value 'doctest', and a table for 'Metadata'. The table has four columns: 'Alias', 'Type', 'Parsing Path', and 'Operation'. There are two rows of metadata: one with Alias 'c', Type 'Integer', and Parsing Path 'a.b.c'; the other with Alias 'd', Type 'String', and Parsing Path 'a.b.d'. Each row has 'Edit' and 'Delete' links. An 'Add' button is at the bottom.

Alias	Type	Parsing Path	Operation
c	Integer	a.b.c	Edit   Delete
d	String	a.b.d	Edit   Delete

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## ArtemisMQ

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select ArtemisMQ as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-41** ArtemisMQ information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the ArtemisMQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>ArtemisMQ</b> .
Data Source Name	Select the ArtemisMQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
Destination Type	Select the message transfer model of the ArtemisMQ data source. The value can be <b>Topic</b> or <b>Queue</b> .
Destination Name	Enter the name of a topic or queue to which to send data. Ensure that the topic or queue already exists.

Parameter	Description
Metadata	<p>Define each underlying key-value data element to be written to the destination in JSON format. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path:</b> full path of metadata. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, the complete path of element d is **a.b.d**, and elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    }],
    {
      "c": "yy",
      "d": "yy"
    }
  }
}
```

The configuration when the destination is ArtemisMQ is similar to that when the destination is ActiveMQ. For details, see [ActiveMQ configuration example](#).

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## DB2

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select DB2 as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-42** DB2 information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the DB2 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>DB2</b> .
Data Source Name	Select the DB2 data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select the data table to which data will be written. Then, click <b>Select Table Field</b> and select only the column fields that you want to write.
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## DIS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select DIS as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-43** DIS information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the DIS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>DIS</b> .
Data Source Name	Select the DIS data source that you configured in <a href="#">Connecting to Data Sources</a> .

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## DWS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select DWS as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-44** DWS information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the DWS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Data Source Type	Select <b>DWS</b> .
Data Source Name	Select the DWS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select the data table to which data will be written. Then, click <b>Select Table Field</b> and select only the table fields that you want to write.
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.
Clear Table	This parameter specifies whether to clear the destination table each time when a schedule starts.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## FTP

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select FTP as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-45** FTP information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the FTP data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Data Source Type	Select <b>FTP</b> .
Data Source Name	Select the FTP data source that you configured in <a href="#">Connecting to Data Sources</a> .
File Path	Enter the path of the folder to be accessed on the FTP server, for example, <b>/data/FDI</b> .
File Name Prefix	Enter the prefix of the file name. This parameter is used together with <b>File Name Extension</b> to define the name of the file to be written to the FTP data source.
File Name Extension	Select a timestamp format for the file name extension. This parameter is used together with <b>File Name Prefix</b> to define the data file to be written to the FTP data source.
File Content Type	Select the content type of the data file to be written. Currently, <b>Text file</b> and <b>Binary file</b> are supported.
File Type	Select the format of the data file to be written. If <b>File Content Type</b> is set to <b>Text file</b> , <b>CSV</b> and <b>TXT</b> are available. If <b>File Content Type</b> is set to <b>Binary file</b> , <b>XLS</b> and <b>XLSX</b> are available.
File Name Encoding	Select the encoding mode of the data file name.
File Content Encoding	This parameter is mandatory only if <b>File Content Type</b> is set to <b>Text file</b> . Select the encoding format of the data file content.
File Separator	This parameter is mandatory only if <b>File Content Type</b> is set to <b>Text file</b> . Enter the field separator for the data file to distinguish different fields in each row of data. By default, the values are separated with commas (,).
Space Format Character	Define the space characters in the data file for data writing.
Write Mode	Select the mode in which integration data is written to a file. <ul style="list-style-type: none"> <li>● <b>Truncate</b>: Delete the file, re-create a file, and write data to the new file.</li> <li>● <b>Append</b>: Incrementally write data to an existing file.</li> </ul>
Add File Header	Determine whether to add a file header.

Parameter	Description
File Header	This parameter is mandatory only if <b>Add File Header</b> is set to <b>Yes</b> . Enter the file header information. Use commas (,) to separate multiple file headers.
Metadata	Define the data fields to be written to the destination file. The number of fields to be integrated at the source determines the number of metadata records defined on the destination. <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following figure shows a configuration example when the destination is FTP. **id**, **name**, and **info** are the data fields to be written to the FTP data source.

**Figure 6-24** FTP configuration example

The screenshot displays the configuration interface for an FTP destination. It includes several input fields and dropdown menus for parameters such as File Path, File Name Prefix, File Name Extension, File Content Type, File Type, File Name Encoding, File Content Encoding, File Separator, Space Format Character, Write Mode, and Add File Header. A Metadata table is also present, listing fields to be written to the destination.

Alias	Type	Operation
id	Integer	Edit   Delete
name	String	Edit   Delete
info	String	Edit   Delete

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## FI HDFS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select FI HDFS as the data source type at the destination.



1. On the **Create Task** page, configure destination information.

**Table 6-46** FI HDFS information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the FI HDFS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FI HDFS</b> .
Data Source Name	Select the FI HDFS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Separator	Enter the field separator for the text data in the FI HDFS data source. The separator is used to distinguish different fields in each row of data.
Storage Subpath	Enter a subpath of the file, which is user-defined.
Storage Block Size (M)	Select the size of data to be written by ROMA Connect to the MRS HDFS data source each time.
Storage Type	Select <b>Text file</b> as the data storage type of the FI HDFS data source.
Batch Number	User-defined batch number. The batch number cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Metadata	Define the data fields to be written to the destination text data. Separate different data fields with delimiters. The number of fields to be integrated at the source determines the number of metadata records defined on the destination. <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The configuration when the destination is FI HDFS is similar to that when the destination is MRS HDFS. For details, see [MRS HDFS configuration example](#).

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## FI Hive

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select FI Hive as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-47** FI Hive information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the FI Hive data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FI Hive</b> .
Data Source Name	Select the FI Hive data source that you configured in <a href="#">Connecting to Data Sources</a> .
Database Name	Select the database to which data will be written.
Table	Select the data table to which data will be written.
Separator	Enter the field separator for the text data in the FI Hive data source. The separator is used to distinguish different fields in each row of data. The existing characters in the data to be integrated cannot be used as separators.
Write Mode	Select the mode in which integration data is written to a data table. <ul style="list-style-type: none"> <li>• <b>Truncate:</b> Delete all data from the destination data table and then write data to the table.</li> <li>• <b>Append:</b> Incrementally write data to an existing data table.</li> </ul>

Parameter	Description
Batch Number	User-defined batch number. The batch number cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Storage Type	Select <b>RCFile</b> or <b>Text file</b> as the storage type for data to be written to the FI Hive data source.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## FI Kafka

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select FI Kafka as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-48** FI Kafka information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the FI Kafka data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>FI Kafka</b> .
Data Source Name	Select the FI Kafka data source that you configured in <a href="#">Connecting to Data Sources</a> .
Topic Name	Enter a topic name that is created in the FI Kafka service and starts with T_.
Key	Enter the key value of a message so that the message will be stored in a specified partition. It can be used as an ordered message queue. If this parameter is left unspecified, messages are stored in different message partitions in a distributed manner.

Parameter	Description
Metadata	<p>Define the data fields to be written to the destination Kafka. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The configuration when the destination is FI Kafka is similar to that when the destination is Kafka. For details, see [Kafka configuration example](#).

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## GaussDB 100

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select GaussDB 100 as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-49** GaussDB 100 information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the GaussDB 100 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>GaussDB 100</b> .
Data Source Name	Select the GaussDB 100 data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.

Parameter	Description
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## GaussDB 200

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select GaussDB 200 as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-50** GaussDB 200 information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the GaussDB 200 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>GaussDB 200</b> .
Data Source Name	Select the GaussDB 200 data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.

Parameter	Description
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## HL7

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select HL7 as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-51** HL7 information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the HL7 data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>HL7</b> .
Data Source Name	Select the HL7 data source that you configured in <a href="#">Connecting to Data Sources</a> .
Encoding Format	Select the encoding mode of data files in the HL7 data source. The value can be <b>UTF-8</b> or <b>GBK</b> .

Parameter	Description
Message Encoding Type	Select the message type of the data to be integrated. This parameter defines the purpose and usage of messages. Set this parameter based on the definition in the HL7 protocol.
Trigger Event Type	Select the event type corresponding to the message type. Set this parameter based on the definition in the HL7 protocol.
Protocol Version	Select the HL7 protocol version used by the HL7 data source.
Metadata	<p>Define the data fields to be written to the destination HL7. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata.</li> <li>• <b>Parsing Path:</b> location of metadata in HL7 messages. For details, see the metadata path configuration description in the subsequent section.</li> </ul>

### Description on Metadata Path Configuration

```
MSH|^~\&|hl7Integration|hl7Integration|||ADT^A01|||2.3|
EVN|A01|20191212155644
PID||PATID1234^5^M11||FN^Patrick^^MR||197001011||xx Street^^NY^^Ox4DP|||||
NK1|1|FN^John^^MR|Father||999-9999
NK1|2|MN^Georgie^^MSS|Mother||999-9999
```

The metadata parsing path of HL7 messages must be set based on the Terser syntax specifications. In the preceding example HL7 message, each line represents an information segment. Each information segment starts with three uppercase letters, which are paragraph symbols of the information segment and are used to indicate content of the information segment. Information segments are separated by separators.

- | : field separator, which is used to divide information segments into different fields. Each field in an information segment is numbered starting from 1 (excluding paragraph symbols). The rest may be deduced by analogy.
- ^ : component separator, which divides the field content into different components. In the fields that are divided into components, the position of a component is identified by a number, starting from 1. The rest may be deduced by analogy.
- ~ : subcomponent separator, which is used to divide a component into subcomponents.

For example, in the PID information segment, if the field position of 19700101 is 7, the parsing path is **/PID-7**. If the field position of xx Street is 11 and the component position is 1, the parsing path is **/PID-11-1**.

For the information segments with the same paragraph symbol in the HL7 message, the repeated paragraph symbol is identified by adding a number

enclosed by brackets after the paragraph symbol. In repeated paragraph symbols, the first is (0), the second is (1), and so on.

For example, in the NK1 information segment, "Father" is located in the first NK1 information segment, and a field location is 3, a parsing path of the NK1 information segment is **NK1(0)-3**. Similarly, the parsing path of Mother is **NK1(1)-3**.

Writing the **19700101** and **xx Street** fields in the preceding HL7 message is used as an example. The following describes the configuration when the destination is HL7:

**Figure 6-25** HL7 configuration example

* Encoding Format	UTF-8	
* Message Encoding Type	ADT	
* Trigger Event Type	A01	
* Version	2.3	
* Metadata	Alias	Type
	ID	Integer
	Address	String
		Parsing Path
		/PID-7
		/PID-11-1

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## HANA

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select HANA as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-52** HANA information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the HANA data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>HANA</b> .
Data Source Name	Select the HANA data source that you configured in <a href="#">Connecting to Data Sources</a> .



Parameter	Description
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## IBM MQ

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select IBM MQ as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-53** IBM MQ information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the IBM MQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>IBM MQ</b> .
Data Source Name	Select the IBM MQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
Destination Type	Select the message transfer model of the IBM MQ data source. The value can be <b>Topic</b> or <b>Queue</b> .

Parameter	Description
Destination Name	Enter the name of a topic or queue to which to send data. Ensure that the topic or queue already exists.
Metadata	<p>Define each underlying key-value data element to be written to the destination in JSON format. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path:</b> full path of metadata. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, the complete path of element d is **a.b.d**, and elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    },
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

The configuration when the destination is IBM MQ is similar to that when the destination is ActiveMQ. For details, see [ActiveMQ configuration example](#).

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## Kafka

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select Kafka as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-54** Kafka information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the Kafka data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>Kafka</b> .
Data Source Name	Select the Kafka data source that you configured in <a href="#">Connecting to Data Sources</a> .
Topic Name	Select the name of the topic to which data is to be written.
Key	Enter the key value of a message so that the message will be stored in a specified partition. It can be used as an ordered message queue. If this parameter is left unspecified, messages are stored in different message partitions in a distributed manner.
Metadata	Define the data fields to be written to the destination Kafka. The number of fields to be integrated at the source determines the number of metadata records defined on the destination. <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following figure shows a configuration example when the destination is Kafka. **id**, **name**, and **info** are the data fields to be written to the Kafka data source.

**Figure 6-26** Kafka configuration example

The screenshot shows a configuration form for Kafka. It includes a dropdown menu for 'Topic Name' with the value 'topic-doc', an empty text input for 'Key', and a table for 'Metadata' with columns 'Alias' and 'Type'. The metadata table lists three fields: 'id' (Integer), 'name' (String), and 'info' (String).

* Topic Name	topic-doc								
Key									
* Metadata	<table border="1"> <thead> <tr> <th>Alias</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>Integer</td> </tr> <tr> <td>name</td> <td>String</td> </tr> <tr> <td>info</td> <td>String</td> </tr> </tbody> </table>	Alias	Type	id	Integer	name	String	info	String
Alias	Type								
id	Integer								
name	String								
info	String								

The structure of the message written to Kafka is {"id":"xx", "name":"yy", "info":"zz"}, where xx, yy, and zz are the data values transferred from the source.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## MySQL

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select MySQL as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-55** MySQL information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the MySQL data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MySQL</b> .
Data Source Name	Select the MySQL data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	This parameter specifies whether to update only the values of input fields. If this parameter is disabled, all integrated synchronization fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## MongoDB

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select MongoDB as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-56** MongoDB information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the MongoDB data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MongoDB</b> .
Data Source Name	Select the MongoDB data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Destination Set	Select the data set to be written to the MongoDB data source. (The data set is equivalent to a data table in a relational database.) Then, click <b>Select Fields in Set</b> and select only the column fields that you want to write.
Upsert	This parameter indicates whether to update or insert data to the destination, that is, whether to directly updating existing data fields in the data set on the destination.
Upsert key	This parameter is mandatory only if <b>Upsert</b> is enabled. Select the data field to be upserted.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## MRS Hive

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select MRS Hive as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

#### NOTE

If the source data field contains special characters \r, \n, and \01, ROMA Connect deletes these special characters and then writes the data to MRS Hive.

**Table 6-57** MRS Hive information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the MRS Hive data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS Hive</b> .
Data Source Name	Select the MRS Hive data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Database Name	Select the database to which data will be written. <b>NOTE</b> You need to use a self-built database instead of the default database of MRS Hive.
Table	Select the data table to which data will be written.
Separator	Enter the field separator for the text data in the MRS Hive data source. The separator is used to distinguish different fields in each row of data.
Write Mode	Select the mode in which integration data is written to a data table. <ul style="list-style-type: none"> <li>• <b>Truncate:</b> Delete all data from the destination data table and then write data to the table.</li> <li>• <b>Append:</b> Incrementally write data to an existing data table.</li> </ul>
Batch Number	User-defined batch number. The batch number cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Storage Type	Select <b>RCFile</b> or <b>Text file</b> as the storage type for data to be written to the MRS Hive data source.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## MRS HDFS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select MRS HDFS as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

#### **NOTE**

If the source data field contains special characters `\r`, `\n`, and `\01`, ROMA Connect deletes these special characters and then writes the data to MRS HDFS.

**Table 6-58** MRS HDFS information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the MRS HDFS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS HDFS</b> .
Data Source Name	Select the MRS HDFS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Separator	Enter the field separator for the text data in the MRS HDFS data source. The separator is used to distinguish different fields in each row of data.
Storage Subpath	Enter the path of the data to be integrated in the hdfs:///hacluster directory of MRS HDFS.
Storage Block Size (M)	Select the size of data to be written each time ROMA Connect writes data to the MRS HDFS data source.
Storage Type	Select <b>Text file</b> as the data storage type of the MRS HDFS data source.
Batch Number	<p>User-defined batch number. The batch number cannot be the same as the destination field in mapping information.</p> <p>The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.</p>
Metadata	<p>Define the data fields to be written to the destination text data. Separate different data fields with delimiters. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>



The following figure shows a configuration example when the destination is MRS HDFS. **id**, **name**, and **info** are the data fields to be written to the MRS HDFS data source.

**Figure 6-27** MRS HDFS configuration example

The screenshot shows a configuration form for MRS HDFS. It includes the following fields:

- \* Separator:** A text input field containing a comma (,).
- \* Storage Subpath:** A text input field containing /data/test.
- \* Storage Block Size (M):** A dropdown menu with 64 selected.
- \* Storage Type:** A dropdown menu with textfile selected.
- Batch Number:** A dropdown menu with a question mark icon.
- \* Metadata:** A table with two columns: Alias and Type.
 

Alias	Type
id	Integer
name	String
info	String

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## MRS HBase

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select MRS HBase as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

#### NOTE

If the source data field contains special characters \r, \n, and \01, ROMA Connect deletes these special characters and then writes the data to MRS HBase.

**Table 6-59** MRS HBase information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the MRS HBase data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Data Source Type	Select <b>MRS HBase</b> .
Data Source Name	Select the MRS HBase data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select the data table to which data will be written.
Column Family	Define the data column fields to be written to the destination data table. The number of fields to be integrated at the source determines the number of metadata records defined on the destination. <b>Field Name:</b> user-defined name of a field in a data column.

The following figure shows a configuration example when the destination is MRS HBase. **id**, **name**, and **info** are the data fields to be written to the MRS HBase data source.

**Figure 6-28** MRS HBase configuration example

★ Table

★ Column Family

**Field Name**

---

---

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## MRS Kafka

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select MRS Kafka as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-60** MRS Kafka information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the MRS Kafka data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>MRS Kafka</b> .
Data Source Name	Select the MRS Kafka data source that you configured in <a href="#">Connecting to Data Sources</a> .
Topic Name	Enter a topic name that is created in the MRS Kafka service and starts with T_.
Key	Enter the key value of a message so that the message will be stored in a specified partition. It can be used as an ordered message queue. If this parameter is left unspecified, messages are stored in different message partitions in a distributed manner.
Metadata	Define the data fields to be written to the destination Kafka. The number of fields to be integrated at the source determines the number of metadata records defined on the destination. <ul style="list-style-type: none"><li>• <b>Alias</b>: user-defined metadata name.</li><li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li></ul>

The configuration when the destination is MRS Kafka is similar to that when the destination is Kafka. For details, see [Kafka configuration example](#).

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## OBS

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled**, you can select OBS as the data source type at the source.

1. On the **Create Task** page, configure destination information.

**Table 6-61** OBS information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the OBS data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>OBS</b> .
Data Source Name	Select the OBS data source that you configured in <a href="#">Connecting to Data Sources</a> .
Object Type	Select the type of the data file to be written to the OBS data source. Currently, <b>Text file</b> and <b>Binary file</b> are supported.
Encoding Format	This parameter is mandatory only if <b>Object Type</b> is set to <b>Text file</b> . Select the encoding mode of the data file to be written to the OBS data source. The value can be <b>UTF-8</b> or <b>GBK</b> .
Path	Enter the name of the object to be written to the OBS data source. The value of <b>Path</b> cannot end with a slash (/).
File Name Prefix	Enter the prefix of the file name. This parameter is used together with <b>Time Format</b> to define the name of the file to be written to the OBS data source.
Time Format	Select the time format to be used in the file name. This parameter is used together with <b>File Name Prefix</b> to define the data file to be written to the OBS data source.
File Type	Select the format of the data file to be written to the OBS data source. A text file can be in TXT or CSV format, and a binary file can be in XLS or XLSX format.
Advanced Attributes	This parameter is mandatory only if <b>File Type</b> is set to <b>csv</b> . Select whether to configure the advanced properties of the file.
Newline	This parameter is mandatory only if <b>Advanced Attributes</b> is set to <b>Enable</b> . Enter a newline character in the file content to distinguish different data lines in the file.

Parameter	Description
Enclosure Character	<p>This parameter is mandatory only if <b>Advanced Attributes</b> is set to <b>Enable</b>.</p> <p>If you select <b>Use</b>, each data field in the data file is enclosed by double quotation marks ("). If a data field contains the same symbol as a separator or newline character, the field will not be split into two fields. For example, if source data contains a data field <b>aa bb</b> and the vertical bar ( ) is set as the separator when the source data is integrated to the destination data file, the field is <b>aa bb</b> in the destination data file and will not be split into <b>aa</b> and <b>bb</b>.</p>
Field Separator	<p>This parameter is mandatory only if <b>File Type</b> is set to <b>txt</b> or <b>Advanced Attributes</b> is set to <b>Enable</b>.</p> <p>Enter the field separator for the file contents to distinguish different fields in each row of data.</p>
Add File Header	<p>Determine whether to add a file header to the data file to be written. The file header is the first line or several lines at the beginning of a file, which helps identify and distinguish the file content.</p>
File Header	<p>This parameter is mandatory only if <b>Add File Header</b> is set to <b>Yes</b>.</p> <p>Enter the file header information. Use commas (,) to separate multiple file headers.</p>
Metadata	<p>Define the data fields to be written to the destination file. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>

The following figure shows a configuration example when the destination is OBS. **id**, **name**, and **info** are the data fields to be written to the OBS data source.

**Figure 6-29** OBS configuration example

* Object Type	Text file								
* Encoding Format	UTF-8								
* Path	data/test								
* File Name Prefix	roma								
* Time Format	yyyy-MM-dd								
* File Type	txt								
* Field Separator	,								
* Add File Header	No								
* Metadata	<input checked="" type="checkbox"/> Text Mode <table border="1"> <thead> <tr> <th>Alias <span>?</span></th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>Double</td> </tr> <tr> <td>name</td> <td>String</td> </tr> <tr> <td>info</td> <td>String</td> </tr> </tbody> </table>	Alias <span>?</span>	Type	id	Double	name	String	info	String
Alias <span>?</span>	Type								
id	Double								
name	String								
info	String								

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## Oracle

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select Oracle as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-62** Oracle information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the Oracle data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>Oracle</b> .

Parameter	Description
Data Source Name	Select the Oracle data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## PostgreSQL

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select PostgreSQL as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-63** PostgreSQL information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the PostgreSQL data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>PostgreSQL</b> .
Data Source Name	Select the PostgreSQL data source that you configured in <a href="#">Connecting to Data Sources</a> .

Parameter	Description
Table	Select the data table to which data is to be written and click <b>Select Table Field</b> to select the data column fields to be integrated.
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## Redis

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select Redis as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-64** Redis information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the Redis data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>Redis</b> .
Data Source Name	Select the Redis data source that you configured in <a href="#">Connecting to Data Sources</a> .



Parameter	Description
Key Prefix	Enter the key name prefix of the data to be integrated in the Redis data source. The key name in the Redis data source consists of the key prefix, separator, and key suffix field. Each row of data is stored in the Redis as the key value. For details about the key format, see <a href="#">Description on Key and Value Formats</a> .
Key Suffix	Select a field whose value is unique in the source data as the key suffix. The key name in the Redis data source consists of the key prefix, separator, and key suffix field. In this way, each row of data can be integrated into different keys of the Redis data source.  If <b>Data Type</b> is set to <b>List</b> , <b>Set</b> , or <b>ZSet</b> , <b>Key Suffix</b> can be left blank. That is, only one key is generated based on the key prefix. In this case, all data rows are integrated to the same key of the Redis data source as elements.
Separator	This parameter is mandatory only if <b>Key Suffix</b> is not left blank.  Enter the separator between <b>Key Prefix</b> and <b>Key Suffix</b> . The key name in the Redis data source consists of <b>Key Prefix</b> , <b>Separator</b> , and <b>Key Suffix</b> .
Data Type	Select the data type of the key in the Redis data source. Options are as follows: <ul style="list-style-type: none"> <li>• String</li> <li>• List</li> <li>• Map</li> <li>• Set</li> <li>• ZSet</li> </ul>
List Appending Mode	This parameter is mandatory only if <b>Data Type</b> is set to <b>List</b> .  Select the data appending mode for the key of the list type. <ul style="list-style-type: none"> <li>• <b>lpush</b>: The current data is inserted to the header of the list.</li> <li>• <b>rpush</b>: The current data is inserted to the end of the list.</li> </ul>
sortColumn	This parameter is mandatory only if <b>Data Type</b> is set to <b>ZSet</b> .  Select the source data field for sorting data elements.
Validity Period (s)	Period for which a key in the Redis data source remains valid. The value <b>0</b> indicates that the key never expires.

Parameter	Description
Write Format	<p>This parameter is mandatory only if <b>Data Type</b> is set to <b>String</b>, <b>List</b>, <b>Set</b>, or <b>Zset</b>. If <b>Data Type</b> is set to <b>Map</b>, the JSON format is used by default.</p> <p>Select <b>JSON</b> or <b>CUSTOMIZE</b> as the format of the data to be written to the Redis data source.</p>
Metadata	<p>Define the format of the value written to be written to the destination key. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias</b>: user-defined metadata name.</li> <li>• <b>Type</b>: data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul> <p>If <b>Write Format</b> is set to <b>JSON</b>, the metadata is stored as the key value in the Redis data source in JSON format. If <b>Write Format</b> is set to <b>CUSTOMIZE</b>, customize the format of the destination value. All metadata is combined with the prefix and suffix, and then stored as the key value in the Redis data source. For details about the value format, see <a href="#">Description on Key and Value Formats</a>.</p>

### Description on Key and Value Formats

Assume that at the destination: **Key Prefix** is set to **roma**; **Key Suffix** is set to the unique key **aaa** of the source data, which ensures that the key name is unique; **Separator** is set to the vertical bar (|), which will be used as a separator between key prefixes and key suffixes.

```
+-----+-----+
| aaa | bbb |
+-----+-----+
| 1 | x |
| 2 | y |
| 3 | z |
+-----+-----+
```

- If **Data Type** is set to **String** and **Write Format** is set to **JSON**, the keys and values written to the Redis data source are shown in [Figure 6-30](#).

```
key          value
-----
roma|1      "{\"bbb\":\"x\",\"aaa\":\"1\"}"
roma|2      "{\"bbb\":\"y\",\"aaa\":\"2\"}"
roma|3      "{\"bbb\":\"z\",\"aaa\":\"3\"}"
```

**Figure 6-30** Metadata configuration (JSON)

* Metadata	Field Name	Type	Operation
	<input type="text" value="bbb"/>	<input type="text" value="string"/>	Delete
	<input type="text" value="aaa"/>	<input type="text" value="string"/>	Delete
<input type="button" value="Synchronize Fields from Data Source"/> <input type="button" value="⊕ Add"/>			

- If **Data Type** is set to **String** and **Write Format** is set to **CUSTOMIZE**, the keys and values written to the Redis data source are shown in **Figure 6-31**.

```

key      value
-----
roma|1   "bbb_x&aaa_1"
roma|2   "bbb_y&aaa_2"
roma|3   "bbb_z&aaa_3"
    
```

**Figure 6-31** Metadata configuration (CUSTOMIZE)

* Metadata	Field Name	Type	Operation
	<input type="text" value="bbb"/>	<input type="text" value="string"/>	<a href="#">Delete</a>
	<input type="text" value="aaa"/>	<input type="text" value="string"/>	<a href="#">Delete</a>
<input type="button" value="Synchronize Fields from Data Source"/> <input type="button" value="⊕ Add"/>			

2. After configuring the destination information, proceed with **Configuring a Data Mapping Rule**.

## RabbitMQ

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select RabbitMQ as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-65** RabbitMQ information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the RabbitMQ data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>RabbitMQ</b> .
Data Source Name	Select the RabbitMQ data source that you configured in <a href="#">Connecting to Data Sources</a> .
New Queue Creation	Determine whether to create a queue in the RabbitMQ data source. <ul style="list-style-type: none"> <li>• If you select <b>Yes</b>, a new queue is created and the data to be integrated is sent to the queue.</li> <li>• If you select <b>No</b>, the data to be integrated is sent to an existing queue.</li> </ul>

Parameter	Description
Exchange Mode	<p>Select a routing mode for the exchange in the RabbitMQ data source to forward messages to the queue. If <b>New Queue Creation</b> is set to <b>Yes</b>, select the routing mode for the new queue. If <b>New Queue Creation</b> is set to <b>No</b>, select the routing mode that is the same as that of the existing destination queue.</p> <ul style="list-style-type: none"> <li>• <b>Direct:</b> If the routing key in a message fully matches the queue, the message will be forwarded to the queue.</li> <li>• <b>Topic:</b> If the routing key in a message approximately matches the queue, the message will be forwarded to the queue.</li> <li>• <b>Fanout:</b> All messages will be forwarded to the queue.</li> <li>• <b>Headers:</b> If the Headers attribute of a message fully matches the queue, the message will be forwarded to the queue.</li> </ul>
Exchange Name	<p>Enter the exchange name of the RabbitMQ data source. If <b>New Queue Creation</b> is set to <b>Yes</b>, the exchange name of the new queue is used. If <b>New Queue Creation</b> is set to <b>No</b>, configure the exchange name that is the same as that of the existing destination queue.</p>
Routing Key	<p>This parameter is mandatory only if <b>Exchange Mode</b> is set to <b>Direct</b> or <b>Topic</b>.</p> <p>RabbitMQ uses the routing key as the judgment condition. Messages that meet the condition will be forwarded to the queue. If <b>New Queue Creation</b> is set to <b>Yes</b>, enter the routing key of the new queue. If <b>New Queue Creation</b> is set to <b>No</b>, enter the routing key that is the same as that of the existing destination queue.</p>
Message Parameters	<p>This parameter is mandatory only if <b>Exchange Mode</b> is set to <b>Headers</b>.</p> <p>RabbitMQ uses Headers as a judgment condition. Messages that meet the condition will be forwarded to a new queue. If <b>New Queue Creation</b> is set to <b>Yes</b>, enter the headers of the new queue. If <b>New Queue Creation</b> is set to <b>No</b>, enter the headers that are the same as those of the existing destination queue.</p>
Queue Name	<p>This parameter is mandatory only if <b>New Queue Creation</b> is set to <b>Yes</b>.</p> <p>Enter the name of a new queue.</p>
Automatic Deletion	<p>This parameter specifies whether a queue will be automatically deleted if no client is connected.</p>

Parameter	Description
Persistence	This parameter specifies whether messages in a queue are stored permanently.
Metadata	<p>Define each underlying key-value data element to be written to the destination in JSON format. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> <li>• <b>Parsing Path:</b> full path of metadata. For details, see <a href="#">Description on Metadata Parsing Path Configuration</a>.</li> </ul>

### Description on Metadata Parsing Path Configuration

- Data in JSON format does not contain arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b.c**, the complete path of element d is **a.b.d**, and elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b.c**, and **Parsing Path** of element d must be set to **a.b.d**.

```
{
  "a": {
    "b": {
      "c": "xx",
      "d": "xx"
    }
  }
}
```

- Data in JSON format contains arrays:

For example, in the following JSON data written to the destination, the complete path of element a is **a**, the complete path of element b is **a.b**, the complete path of element c is **a.b[i].c**, and the complete path of element d is **a.b[i].d**. Elements c and d are underlying data elements.

In this scenario, **Parsing Path** of metadata c must be set to **a.b[i].c**, and **Parsing Path** of element d must be set to **a.b[i].d**.

```
{
  "a": {
    "b": [{
      "c": "xx",
      "d": "xx"
    },
    {
      "c": "yy",
      "d": "yy"
    }
  ]
}
```

The preceding JSON data that does not contain arrays is used as an example. The following describes the configuration when the destination is RabbitMQ:

**Figure 6-32** RabbitMQ configuration example

The screenshot shows a configuration form for RabbitMQ. It includes the following fields and values:

- New Queue Creation:** Radio buttons for Yes and No, with 'No' selected.
- Exchange Mode:** A dropdown menu set to 'Direct'.
- Exchange Name:** A text input field containing 'doctest'.
- Routing Key:** A text input field containing 'roma'.
- Automatic Deletion:** Radio buttons for Yes and No, with 'Yes' selected.
- Persistency:** Radio buttons for Yes and No, with 'Yes' selected.
- Metadata:** A table with three columns: Alias, Type, and Parsing Path.
 

Alias	Type	Parsing Path
c	Integer	a.b.c
d	String	a.b.d

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## SQL Server

[Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select SQL Server as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-66** SQL Server information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the SQL Server data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>SQL Server</b> .
Data Source Name	Select the SQL Server data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.

Parameter	Description
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## TaurusDB

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select TaurusDB as the data source type at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-67** TaurusDB information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the TaurusDB data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select <b>TaurusDB</b> .
Data Source Name	Select the TaurusDB data source that you configured in <a href="#">Connecting to Data Sources</a> .
Table	Select an existing table and click <b>Select Table Field</b> to select only the column fields that you want to integrate.

Parameter	Description
Batch Number Field	Select a field whose type is String and length is greater than 14 characters in the destination table as the batch number field. In addition, the batch number field cannot be the same as the destination field in mapping information.  The value of this field is a random number, which is used to identify the data in the same batch. The data inserted in the same batch uses the same batch number, indicating that the data is inserted in the same batch and can be used for location or rollback.
Update Changed Fields Only	If this option is enabled, only the table fields whose values change are updated. If this option is disabled, all table fields are updated.

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## Custom Data Sources

### [Back to Overview](#)

If **Integration Mode** is set to **Scheduled** or **Real-Time**, you can select a custom data source at the destination.

1. On the **Create Task** page, configure destination information.

**Table 6-68** Custom data source information at the destination

Parameter	Description
Instance	Set this parameter to the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.
Integration Application	Select the integration application to which the custom data source belongs. Ensure that the integration application has been configured in <a href="#">Connecting to Data Sources</a> .
Data Source Type	Select a custom data source type.
Data Source Name	Select the custom data source that you configured in <a href="#">Connecting to Data Sources</a> .



Parameter	Description
Metadata	<p>Define each underlying key-value data element to be written to the destination in JSON format. The number of fields to be integrated at the source determines the number of metadata records defined on the destination.</p> <ul style="list-style-type: none"> <li>• <b>Alias:</b> user-defined metadata name.</li> <li>• <b>Type:</b> data type of metadata. The value must be the same as the data type of the corresponding parameter in the source data.</li> </ul>
<p>In addition to the preceding parameters, different custom data sources define different writer parameters. Set the parameters based on the original definition specifications of the connector. You can locate the connector used by the custom data source on the <b>Assets</b> page of the ROMA Connect console and view the writer parameter definition of the connector.</p>	

The following figure shows an example of configuring a custom data source for sending emails. The destination is the custom data source. The **receiver** and **title** parameters are the destination parameters defined in the connector. **id**, **name**, and **info** are the data fields to be written to the custom data source.

**Figure 6-33** Custom data source configuration example

* receiver	<input type="text" value="test@example.com"/>								
* title	<input type="text" value="User information"/>								
* Metadata	<table border="1"> <thead> <tr> <th>Alias ?</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>id</td> <td>Integer</td> </tr> <tr> <td>name</td> <td>String</td> </tr> <tr> <td>info</td> <td>String</td> </tr> </tbody> </table>	Alias ?	Type	id	Integer	name	String	info	String
Alias ?	Type								
id	Integer								
name	String								
info	String								

2. After configuring the destination information, proceed with [Configuring a Data Mapping Rule](#).

## 6.3.4 Configuring a Data Mapping Rule

### Overview

This topic describes how to configure mapping information for a data integration task. Based on mappings between source data fields and destination data fields, ROMA Connect converts the obtained source data and writes it to the destination.

### Configuring Mapping Information

1. On the **Create Task** page, configure mapping information in auto or manual mode.

 **NOTE**


- Do not use the keywords of the corresponding database as the field names at the source and destination. Otherwise, the task may become abnormal.
  - If MRS Hive is used as the data source type at the destination and you need to configure partition field writing, see [Mapping Configuration of MRS Hive Partition Fields](#).
- **Auto Mapping Configuration**

If metadata is defined at both the source and destination, you can use the automatic mapping mode to configure mapping information.

Click **Automatic Mapping**. A mapping rule between the source and destination data fields is automatically created. If the data table fields at the source and destination are inconsistent, click **Edit** on the right of the mapping to change the data fields at the source or destination. Alternatively, click **Add Mapping** to add the mapping between the data fields at the source and those at the destination.

**Figure 6-34** Automatic mapping

Destination Field	Destination FL...	Length	Source Field/Constant	Source Field Type	Length	Operation
col01	long	--	col01	integer	--	<a href="#">Edit</a>   <a href="#">Delete</a>
col02	string	--	col02	string	--	<a href="#">Edit</a>   <a href="#">Delete</a>
col03	string	--	col03	string	--	<a href="#">Edit</a>   <a href="#">Delete</a>

 Add Mapping

– **Manual Mapping Configuration**

You can manually add mapping rules between source data fields and destination data fields. This method applies to the integration scenario of all data types. You can configure the mapping rule by entering a key-value pair or entering a value in the text box.

▪ **Key-Value Pair Mode**

By default, the key-value pair input mode is used. Click **Add Field Mapping** to add mapping rules from source data fields to destination data fields one by one.

▪ **Text Mode**

Click **Text Mode** and enter the mapping rule script in the text box. The format is as follows:

```
[[
  {
    "sourceKey": "a1",
    "targetKey": "b1"
  },
  {
    "sourceKey": "a2",
    "targetKey": "b2"
  }
]]
```

**sourceKey** indicates a source data field, and **targetKey** indicates a destination data field. In the preceding example, source field **a1** is mapped to destination field **b1**, and source field **a2** is mapped to destination field **b2**.

- After the mapping information is configured, if you need to configure the abnormal data storage and post-integration operation, go to **(Optional) Configuring Fault Information Storage** and **(Optional) Configuring the Post-Integration Operation**. Otherwise, click **Create** to complete the data integration task configuration.

## Mapping Configuration of MRS Hive Partition Fields

When the data source type is set to MRS Hive at the destination, partition fields can be written. You can configure the partition fields based on the site requirements.

**Source Field** corresponding to the partition field must be manually entered. The specific requirements are as follows:

**Format:** {Partition field source field}.format({Character string parsing format}, {Partition field parsing format}","{year|month|day|hour|minute|second},{offset})

- If {Partition field source field} is of the String type, {String parsing format} must be specified.
- If {Partition field source field} is of the Timestamp type, {String parsing format} can be left blank.
- If {Partition field source field} is empty, the time when data is written to the destination is used as the partition field.

**Figure 6-35** Partition field mapping of MRS Hive

ⓘ Ensure that the field type of the source matches that of the destination. Otherwise, an error may occur during type conversion.

Destination Field	Source Field	Operation
id	id	Edit   Delete
name	name	Edit   Delete
yyymm	createtime.format("ddMMyyyy","yyyyMM",hour,1) ⓘ	Edit   Delete

⊕ Add Mapping

For example, if the partition field on the destination is yyymm and the **createtime** field on the source is used as the source field of the partition field, the time format of the **createtime** field is ddMMyyyy (day, month, and year), and the partition field on the destination is yyyyMM (year and month). If the value of the partition field is one hour later than the value of the source field, the source field name of the partition field yyymm in the mapping information is as follows:

- If the value of **createtime** is of the String type, set this parameter to createtime.format("ddMMyyyy", "yyyyMM", hour, 1).
- If the value of **createtime** is of the Timestamp type, set this parameter to createtime.format("", "yyyyMM", hour, 1).
- If the time when data is written to the destination is used as the partition field, set this parameter to .format("", "yyyyMM", hour, 1).

## 6.3.5 (Optional) Configuring Fault Information Storage

### Overview

This topic describes how to configure data storage for abnormal data integration tasks. This configuration is available only if the data source type is DB2, DWS, GaussDB 200, GaussDB 100, MySQL, TaurusDB, Oracle, PostgreSQL, or SQL Server at the destination.

During each task execution, if some data at the source meets integration conditions but cannot be integrated to the destination due to network jitter or abnormal data conversion, ROMA Connect stores the data to the OBS bucket as text files.

### Procedure

Before configuring fault information storage, ensure that you have connected the OBS data source to ROMA. For details, see [Connecting to an OBS Data Source](#).

1. On the **Create Task** page, configure parameters for fault information storage.

**Table 6-69** Parameters for fault information storage

Parameter	Description
Data Source Type	This parameter can only be set to <b>OBS</b> .
Integration Application	Select the integration application to which the OBS data source belongs.
Data Source Name	Select the OBS data source that you configured.
Path	Enter the object name of the OBS data source where abnormal data is to be stored. The value of <b>Path</b> cannot end with a slash (/).

2. If you need to configure the post-integration operation, go to [\(Optional\) Configuring the Post-Integration Operation](#). Otherwise, click **Create** to complete the data integration task configuration.

## 6.3.6 (Optional) Configuring the Post-Integration Operation

### Overview

After a scheduled data integration task is successfully executed, messages can be sent to Kafka or MRS Kafka based on the configuration. In this way, a third-party platform (for example, DGC) can subscribe to Kafka or MRS Kafka to obtain messages and determine whether to perform data extraction based on the site requirements.

This section describes how to configure message sending after a task is executed. If message sending is not required, you do not need to configure message sending.

## Prerequisites

- The Kafka and MRS Kafka data sources have been created under the corresponding application. For details, see [Connecting to a Kafka Data Source](#) and [Connecting to an MRS Kafka Data Source](#).
- A topic has been created and granted with the publish and subscribe permissions. For details about how to create a topic and grant permissions, see [Message Integration Guide](#).

## Procedure

1. On the **Create Task** page, configure post-integration operation information.

**Table 6-70** Parameters for configuring the post-integration operation

Parameter	Description
Action	<p>Determine whether to send a message after the task is executed.</p> <p><b>Do no send messages:</b> No message is sent after the task is successfully executed.</p> <p><b>Send messages to Kafka:</b> A message will be sent to Kafka after the task is successfully executed.</p> <p><b>Send messages to MRS Kafka:</b> A message will be sent to MRS Kafka after the task is successfully executed.</p>
Application	Select the integration application.
Data Source	<p>Select the data source type that is the same as that selected for <b>Action</b>.</p> <p>For example, if <b>Send messages to Kafka</b> is selected for <b>Action</b>, select <b>Kafka</b> for <b>Data Source</b>.</p>
Topic	Select the topic to which the message is to be sent.
Custom Message Content	<p>Customize the content of the message to be sent by using <code>\${}</code> to reference variables. The supported variables are as follows:</p> <p>taskName: task name.</p> <p>dataCount: data volume.</p> <p>targetTable: destination table.</p> <p>dataSize: data size.</p> <p>For example, the content <code>\${taskName}</code> indicates that only the task name is sent.</p>

2. Click **Create**.

## 6.4 Creating a Composite Data Integration Task

### 6.4.1 Configuring Oracle CDC (LogMiner)

#### Overview

Change Data Capture (CDC) enables ROMA Connect to synchronize data sources in real time and synchronously delete data tables.

ROMA Connect supports two CDC modes: XStream and LogMiner. This section describes how to enable the CDC function in LogMiner mode for the Oracle database. In addition, the CDC configuration varies according to the Oracle database type (CDB or non-CDB).

#### Prerequisites

- If the Oracle database is deployed in primary/secondary mode, the secondary Oracle database is not used.
- Fields of the binary type, such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB, cannot be collected.
- If a table contains fields of the SDO\_GEOMETRY type, the Oracle database does not generate redo logs when data changes. Therefore, composite tasks cannot be used for data collection.

#### Configuration When the Oracle Database Is Not a CDB Database

It is recommended that the database administrator configure the CDC function.

1. Enable log archiving.

- a. Run the following command to connect to the database as user **sys**:

In practice, you can connect to the database in multiple modes. The following uses the command line mode as an example.

```
sqlplus /nolog  
CONNECT sys/password@host:port AS SYSDBA;
```

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
  - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on site requirements.
- b. Run the following command to check whether the log archiving function is enabled:  

```
archive log list;
```

    - If the message "Database log mode: No Archive Mode" is displayed, log archiving is disabled. Go to the next step.

- If the message "Database log mode: No Archive Mode" is displayed, log archiving is enabled. Go to [1.f](#).
- c. Run the following command to set archive log parameters:
- ```
alter system set db_recovery_file_dest_size = 100G;  
alter system set db_recovery_file_dest = '/opt/oracle/oradata/recovery_area' scope=spfile;
```
- Among them:
- **100G** indicates the size of the log file storage space. Set this parameter based on site requirements.
  - **/opt/oracle/oradata/recovery\_area** indicates the log storage path. Set this parameter based on site requirements. Ensure that the path has been created in advance.
- d. Run the following command to enable log archiving function:

---

**NOTICE**

- Enabling the log archiving function requires database restart, which will interrupt services. Exercise caution when performing this operation.
- Archived logs occupy a large amount of disk space. If the disk space is full, services are affected. Therefore, you need to periodically delete expired archive logs.

---

```
shutdown immediate;  
startup mount;  
alter database archivelog;  
alter database open;
```

- e. Run the following command to check whether the log archiving function is enabled:
- ```
archive log list;
```
- If the message "Database log mode: Archive Mode" is displayed, log archiving is enabled.
- f. Run the following command to exit the database:
- ```
exit;
```
2. Install the LogMiner tool.
- a. Run the following command to connect to the database instance as user **sys**:
- ```
sqlplus sys/password@host:port/SID as sysdba
```
- Among them:
- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
  - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on site requirements.

- **SID** indicates the name of the instance where the data to be synchronized is located. Set this parameter based on site requirements.
  - b. Run the following command to check whether LogMiner is installed successfully:

```
desc DBMS_LOGMNR
desc DBMS_LOGMNR_D
```

    - If no information is displayed, LogMiner is not installed. Go to the next step.
    - If information is displayed, LogMiner has been installed. Go to [3](#).
  - c. Run the following commands to install LogMiner:

```
@$ORACLE_HOME/rdbms/admin/dbmslm.sql
@$ORACLE_HOME/rdbms/admin/dbmslmd.sql
```
3. Create a LogMiner user and grant permissions to the user.
- a. Run the following commands to create a LogMiner user role and configure permissions for the role:

```
create role roma_logminer_privs;
grant create session,
execute_catalog_role,
select any transaction,
flashback any table,
select any table,
lock any table,
select any dictionary to roma_logminer_privs;
grant select on SYSTEM.LOGMNR_COL$ to roma_logminer_privs;
grant select on SYSTEM.LOGMNR_OBJ$ to roma_logminer_privs;
grant select on SYSTEM.LOGMNR_USER$ to roma_logminer_privs;
grant select on SYSTEM.LOGMNR_UID$ to roma_logminer_privs;
grant select on V_$DATABASE to roma_logminer_privs;
grant select_catalog_role to roma_logminer_privs;
grant LOGMINING to roma_logminer_privs;
```

Among them:

    - **roma\_logminer\_privs** indicates the role name of the LogMiner user. Set this parameter based on site requirements.
    - **grant LOGMINING to roma\_logminer\_privs;** can be added only if the Oracle version is 12c. Otherwise, delete this row.
  - b. Run the following command to create user **LogMiner**:

```
create user roma_logminer identified by password default tablespace users;
grant roma_logminer_privs to roma_logminer;
alter user roma_logminer quota unlimited on users;
```

Among them:

    - **roma\_logminer** indicates the name of the LogMiner user. Set this parameter based on site requirements.
    - **password** indicates the password of the LogMiner user. Set it based on site requirements.
    - **roma\_logminer\_privs** indicates the role of the LogMiner user, which is created in [3.a](#).
  - c. Run the following command to modify log record parameters:

```
alter database add supplemental log data (all) columns;
```
  - d. Run the following command to disconnect the database:



```
exit;
```

## Configuration When the Oracle Database Is a CDB Database

It is recommended that the database administrator configure the CDC function.

1. Enable log archiving.

- a. Run the following command to connect to the database as user **sys**:

In practice, you can connect to the database in multiple modes. The following uses the command line mode as an example.

```
sqlplus /nolog  
CONNECT sys/password@host:port AS SYSDBA;
```

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
- **host** indicates the IP address of the server where the database instance is located. Set this parameter based on site requirements.
- **port** indicates the port used by the database instance. Set this parameter based on site requirements.

- b. Run the following command to check whether the log archiving function is enabled:

```
archive log list;
```

- If the message "Database log mode: No Archive Mode" is displayed, log archiving is disabled. Go to the next step.
- If the message "Database log mode: Archive Mode" is displayed, log archiving is enabled. Go to [1.f](#).

- c. Run the following command to set archive log parameters:

```
alter system set db_recovery_file_dest_size = 100G;  
alter system set db_recovery_file_dest = '/opt/oracle/oradata/recovery_area' scope=spfile;
```

Among them:

- **100G** indicates the size of the log file storage space. Set this parameter based on site requirements.
- **/opt/oracle/oradata/recovery\_area** indicates the log storage path. Set this parameter based on site requirements. Ensure that the path has been created in advance.

- d. Run the following command to enable log archiving function:

---

### NOTICE

- Enabling the log archiving function requires database restart, which will interrupt services. Exercise caution when performing this operation.
  - Archived logs occupy a large amount of disk space. If the disk space is full, services are affected. Therefore, you need to periodically delete expired archive logs.
-

- ```
shutdown immediate;
startup mount;
alter database archivelog;
alter database open;
```
- e. Run the following command to check whether the log archiving function is enabled:
 

```
archive log list;
```

If the message "Database log mode: Archive Mode" is displayed, log archiving is enabled.
  - f. Run the following command to disconnect the database:
 

```
exit;
```
2. Install the LogMiner tool.
    - a. Run the following command to connect to the database instance as user **sys**:
 

```
sqlplus sys/password@host:port/SID as sysdba
```

Among them:

      - **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
      - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on site requirements.
      - **port** indicates the port used by the database instance. Set this parameter based on site requirements.
      - **SID** indicates the name of the instance where the data to be synchronized is located. Set this parameter based on site requirements.
    - b. Run the following command to check whether LogMiner is installed successfully:
 

```
desc DBMS_LOGMNR
desc DBMS_LOGMNR_D
```

      - If no information is displayed, LogMiner is not installed. Go to the next step.
      - If information is displayed, LogMiner has been installed. Go to [3](#).
    - c. Run the following commands to install LogMiner:
 

```
@$ORACLE_HOME/rdbms/admin/dbmslm.sql
@$ORACLE_HOME/rdbms/admin/dbmslmd.sql
```
  3. Create a LogMiner user and grant permissions to the user.
    - a. Run the following commands to create a LogMiner user role and configure permissions for the role:
 

```
create role ###roma_logminer_privs container=all;
grant create session,
execute_catalog_role,
select any transaction,
flashback any table,
select any table,
lock any table,
logmining,
set container,
select any dictionary to ###roma_logminer_privs container=all;
grant select on SYSTEM.LOGMNR_COL$ to ###roma_logminer_privs container=all;
grant select on SYSTEM.LOGMNR_OBJ$ to ###roma_logminer_privs container=all;
grant select on SYSTEM.LOGMNR_USER$ to ###roma_logminer_privs container=all;
```

```
grant select on SYSTEM.LOGMNR_UID$ to c##roma_logminer_privs container=all;  
grant select on V_$DATABASE to c##roma_logminer_privs container=all;  
grant select_catalog_role to c##roma_logminer_privs container=all;
```

**c##roma\_logminer\_privs** indicates the role name of the LogMiner user. Set this parameter based on site requirements.

- b. Run the following command to create user **LogMiner**:  
create user **c##roma\_logminer** identified by **password** default tablespace users container=all;  
grant **c##roma\_logminer\_privs** to **c##roma\_logminer** container=all;  
alter user **c##roma\_logminer** quota unlimited on users container=all;

Among them:

- **c##roma\_logminer** indicates the name of the LogMiner user. Set this parameter based on site requirements.
  - **password** indicates the password of the LogMiner user. Set it based on site requirements.
  - **c##roma\_logminer\_privs** indicates the role of the LogMiner user, which is created in [3.a](#).
- c. Run the following command to modify log record parameters:  
alter database add supplemental log data (all) columns;
  - d. Run the following command to disconnect the database:  
exit;

## 6.4.2 Configuring Oracle CDC (XStream)

### Overview

Change Data Capture (CDC) enables ROMA Connect to synchronize data sources in real time and synchronously delete data tables.

ROMA Connect supports two CDC modes: XStream and LogMiner. This section describes how to enable the CDC function in XStream mode for the Oracle database. In addition, the CDC configuration varies according to the Oracle database type (CDB or non-CDB).

### Prerequisites

- The Oracle database must have the OGG license (the OGG does not need to be installed).
- If the Oracle database is deployed in primary/secondary mode, the secondary Oracle database is not used.
- Fields of the binary type, such as BINARY, VARBINARY, TINYBLOB, BLOB, MEDIUMBLOB, and LONGBLOB, cannot be collected.

### Configuration When the Oracle Database Is Not a CDB Database

It is recommended that the database administrator configure the CDC function.

1. Enable log archiving and XStream.
  - a. Run the following command to connect to the database as user **sys**:  
In practice, you can connect to the database in multiple modes. The following uses the command line mode as an example.

```
sqlplus /nolog  
CONNECT sys/password@host:port AS SYSDBA;
```

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
  - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
- b. Run the following command to enable Xstream:
- ```
alter system set enable_goldengate_replication=true;
```
- c. Run the following command to check whether the log archiving function is enabled:
- ```
archive log list;
```
- If the message "Database log mode: No Archive Mode" is displayed, log archiving is disabled. Go to the next step.
  - If the message "Database log mode: No Archive Mode" is displayed, log archiving is enabled. Go to [1.g](#).
- d. Run the following command to set archive log parameters:
- ```
alter system set db_recovery_file_dest_size = 100G;  
alter system set db_recovery_file_dest = '/opt/oracle/oradata/recovery_area' scope=spfile;
```
- Among them:
- **100G** indicates the size of the log file storage space. Set this parameter based on the site requirements.
  - **/opt/oracle/oradata/recovery\_area** indicates the log storage path. Set this parameter based on the site requirements. Ensure that the path has been created in advance.
- e. Run the following command to enable log archiving function:

---

#### NOTICE

- Enabling the log archiving function requires database restart, which will interrupt services. Exercise caution when performing this operation.
- Archived logs occupy a large amount of disk space. If the disk space is full, services are affected. Therefore, you need to periodically delete expired archive logs.

---

```
shutdown immediate;  
startup mount;  
alter database archivelog;  
alter database open;
```

- f. Run the following command to check whether the log archiving function is enabled:
- ```
archive log list;
```



```
CREATE TABLESPACE xstream_tbs DATAFILE '/opt/oracle/oradata/orcl/xstream_tbs.dbf' SIZE
25M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
CREATE USER xstrm IDENTIFIED BY password DEFAULT TABLESPACE xstream_tbs QUOTA
UNLIMITED ON xstream_tbs;
GRANT CREATE SESSION TO xstrm;
GRANT SELECT ON V_$DATABASE to xstrm;
GRANT FLASHBACK ANY TABLE TO xstrm;
GRANT SELECT ANY TABLE to xstrm;
GRANT LOCK ANY TABLE TO xstrm;
grant select_catalog_role to xstrm;
```

Among them:

- **xstream\_tbs** indicates the tablespace name of the ROMA Connect connection user. Set it based on site requirements.
  - **/opt/oracle/oradata/orcl/xstream\_tbs.dbf** indicates the tablespace file of the ROMA Connect connection user. Set it based on site requirements.
  - **xstrm** indicates the name of the ROMA Connect connection user. Set it based on site requirements.
  - **password** indicates the password of the ROMA Connect connection user. Set it based on site requirements.
- d. Run the following command to modify log record parameters:  
alter database add supplemental log data (all) columns;
- e. Run the following command to disconnect the database:  
exit;
3. Create an XStream outbound server.

 **NOTE**

An XStream outbound server can be used only in one integration task. If the database needs to be used in multiple integration tasks, you need to create multiple XStream outbound servers.

- a. Run the following command to connect to the database instance as user **xstrmadmin**:
- ```
sqlplus xstrmadmin/password@host:port/SID
```

Among them:

- **xstrmadmin** indicates the username of the XStream administrator, which is created in [2.b](#).
  - **password** indicates the password of the XStream administrator, which is configured in [2.b](#).
  - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
  - **SID** indicates the name of the instance where the data to be synchronized is located. Set this parameter based on the site requirements.
- b. Run the following command to create an XStream outbound server:

```
DECLARE
  tables DBMS_UTILITY.UNCL_ARRAY;
  schemas DBMS_UTILITY.UNCL_ARRAY;
BEGIN
  tables(1) := NULL;
  schemas(1) := 'ROMA';
  DBMS_XSTREAM_ADM.CREATE_OUTBOUND(
    server_name => 'dbzxout',
    table_names => tables,
    schema_names => schemas
  );
END;
```

Among them:

- **ROMA** indicates the schemas where the data table to be synchronized is located, that is, the schemas from which the CDC needs to capture data. Set this parameter based on site requirements.
  - **dbzxout** indicates the outbound server name, which must be unique. Set this parameter based on site requirements.
- c. Run the following command to disconnect the database:  
exit;
- d. Run the following command to connect to the database instance as user **sys**:  
sqlplus sys/**password**@**host:port**/**SID** as sysdba

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
  - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
  - **SID** indicates the name of the instance where the data to be synchronized is located. Set this parameter based on the site requirements.
- e. Run the following command to allow user **xstrm** to connect to the XStream outbound server:
- ```
BEGIN
  DBMS_XSTREAM_ADM.ALTER_OUTBOUND(
    server_name => 'dbzxout',
    connect_user => 'xstrm'
  );
END;
```

Among them:

- **dbzxout** is the outbound server name, which is configured in [3.b](#).
  - **xstrm** is the username for connecting to ROMA Connect. It is configured in [2.c](#).
- f. Run the following command to disconnect the database:  
exit;

## Configuration When the Oracle Database Is a CDB Database

It is recommended that the database administrator configure the CDC function.

1. Enable log archiving and XStream.

- a. Run the following command to connect to the database as user **sys**:  
In practice, you can connect to the database in multiple modes. The following uses the command line mode as an example.

```
sqlplus /nolog  
CONNECT sys/password@host:port AS SYSDBA;
```

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
  - **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
- b. Run the following command to enable Xstream:
- ```
alter system set enable_goldengate_replication=true;
```
- c. Run the following command to check whether the log archiving function is enabled:
- ```
archive log list;
```
- If the message "Database log mode: No Archive Mode" is displayed, log archiving is disabled. Go to the next step.
  - If the message "Database log mode: No Archive Mode" is displayed, log archiving is enabled. Go to [1.g](#).
- d. Run the following command to set archive log parameters:
- ```
alter system set db_recovery_file_dest_size = 100G;  
alter system set db_recovery_file_dest = '/opt/oracle/oradata/recovery_area' scope=spfile;
```
- Among them:
- **100G** indicates the size of the log file storage space. Set this parameter based on the site requirements.
  - **/opt/oracle/oradata/recovery\_area** indicates the log storage path. Set this parameter based on the site requirements. Ensure that the path has been created in advance.
- e. Run the following command to enable log archiving function:

---

### NOTICE

- Enabling the log archiving function requires database restart, which will interrupt services. Exercise caution when performing this operation.
  - Archived logs occupy a large amount of disk space. If the disk space is full, services are affected. Therefore, you need to periodically delete expired archive logs.
-



```
shutdown immediate;  
startup mount;  
alter database archivelog;  
alter database open;
```

- f. Run the following command to check whether the log archiving function is enabled:

```
archive log list;
```

If the message "Database log mode: Archive Mode" is displayed, log archiving is enabled.

- g. Run the following command to disconnect the database:

```
exit;
```

2. Create an XStream user and grant permissions to the user.

- a. Run the following command to connect to the database instance as user **sys**:

```
sqlplus sys/password@host:port/SID as sysdba
```

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.
- **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
- **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
- **SID** indicates the name of the CDB database where the data to be synchronized is located. Set this parameter based on the site requirements.

- b. Run the following commands to create an XStream administrator and configure permissions for the administrator:

```
CREATE TABLESPACE xstream_admin_tbs DATAFILE '/opt/oracle/oradata/ORCLCDB/  
xstream_admin_tbs.dbf' SIZE 25M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;  
alter session set container = ORCLPDB1;  
CREATE TABLESPACE xstream_admin_tbs DATAFILE '/opt/oracle/oradata/ORCLCDB/ORCLPDB1/  
xstream_admin_tbs.dbf' SIZE 25M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;  
alter session set container = CDB$ROOT;
```

```
CREATE USER c##xstrmadmin IDENTIFIED BY password DEFAULT TABLESPACE  
xstream_admin_tbs QUOTA UNLIMITED ON xstream_admin_tbs CONTAINER=ALL;  
GRANT CREATE SESSION, SET CONTAINER TO c##xstrmadmin CONTAINER=ALL;  
BEGIN  
  DBMS_XSTREAM_AUTH.GRANT_ADMIN_PRIVILEGE(  
    grantee      => 'c##xstrmadmin',  
    privilege_type => 'CAPTURE',  
    grant_select_privileges => TRUE,  
    container    => 'ALL'  
  );  
END;
```

Among them:

- **xstream\_admin\_tbs** indicates the tablespace name of the XStream administrator. Set it based on site requirements.
- **/opt/oracle/oradata/ORCLCDB/xstream\_admin\_tbs.dbf** indicates the tablespace file of the XStream administrator in the CDB. Set it based on site requirements.

- **ORCLPDB1** indicates the name of the PDB database.
  - **/opt/oracle/oradata/ORCLCDB/ORCLPDB1/xstream\_adm\_tbs.dbf** indicates the tablespace file of the XStream administrator in the PDB. Set it based on site requirements.
  - **c##xstrmadmin** indicates the username of the XStream administrator. Set it based on site requirements.
  - **password** indicates the password of the XStream administrator. Set it based on site requirements.
- c. Run the following commands to create a user for connect ROMA Connect to the database and configure permissions for the user:
- ```
CREATE TABLESPACE xstream_tbs DATAFILE '/opt/oracle/oradata/ORCLCDB/xstream_tbs.dbf'
SIZE 25M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
alter session set container = ORCLPDB1;
CREATE TABLESPACE xstream_tbs DATAFILE '/opt/oracle/oradata/ORCLCDB/ORCLPDB1/
xstream_tbs.dbf' SIZE 25M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;
alter session set container = CDB$ROOT;

CREATE USER c##xstrm IDENTIFIED BY password DEFAULT TABLESPACE xstream_tbs QUOTA
UNLIMITED ON xstream_tbs CONTAINER=ALL;
GRANT CREATE SESSION TO c##xstrm CONTAINER=ALL;
GRANT SET CONTAINER TO c##xstrm CONTAINER=ALL;
GRANT SELECT ON V_$DATABASE to c##xstrm CONTAINER=ALL;
GRANT FLASHBACK ANY TABLE TO c##xstrm CONTAINER=ALL;
GRANT SELECT ANY TABLE to c##xstrm CONTAINER=ALL;
GRANT LOCK ANY TABLE TO c##xstrm CONTAINER=ALL;
grant select_catalog_role to c##xstrm CONTAINER=ALL;
```
- Among them:
- **xstream\_tbs** indicates the tablespace name of the ROMA Connect connection user. Set it based on site requirements.
  - **/opt/oracle/oradata/ORCLCDB/xstream\_tbs.dbf** indicates the tablespace file of the ROMA Connect connection user in the CDB. Set it based on site requirements.
  - **ORCLPDB1** indicates the name of the PDB database.
  - **/opt/oracle/oradata/ORCLCDB/ORCLPDB1/xstream\_tbs.dbf** indicates the tablespace file of the ROMA Connect connection user in the PDB. Set it based on site requirements.
  - **c##xstrm** indicates the name of the ROMA Connect connection user. Set it based on site requirements.
  - **password** indicates the password of the ROMA Connect connection user. Set it based on site requirements.
- d. Run the following command to modify log record parameters:
- ```
alter database add supplemental log data (all) columns;
```
- e. Run the following command to disconnect the database:
- ```
exit;
```
3. Create an XStream outbound server.

 NOTE

An XStream outbound server can be used only in one integration task. If the database needs to be used in multiple integration tasks, you need to create multiple XStream outbound servers.

- a. Run the following command to connect to the database instance as user **c##xstrmadmin**:

```
sqlplus c##xstrmadmin/password@host:port/SID
```

Among them:

- **c##xstrmadmin** indicates the username of the XStream administrator, which is created in [2.b](#).
- **password** indicates the password of the XStream administrator, which is configured in [2.b](#).
- **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
- **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
- **SID** indicates the name of the CDB database where the data to be synchronized is located. Set this parameter based on the site requirements.

- b. Run the following command to create an XStream outbound server:

```
DECLARE
  tables DBMS_UTILITY.UNCL_ARRAY;
  schemas DBMS_UTILITY.UNCL_ARRAY;
BEGIN
  tables(1) := NULL;
  schemas(1) := 'ROMA';
  DBMS_XSTREAM_ADM.CREATE_OUTBOUND(
    server_name => 'dbzxout',
    table_names => tables,
    schema_names => schemas);
END;
```

Among them:

- **ROMA** indicates the schemas where the data table to be synchronized is located, that is, the schemas from which the CDC needs to capture data. Set this parameter based on site requirements.
  - **dbzxout** indicates the outbound server name, which must be unique. Set this parameter based on site requirements.
- c. Run the following command to disconnect the database:
- ```
exit;
```
- d. Run the following command to connect to the database instance as user **sys**:
- ```
sqlplus sys/password@host:port/SID as sysdba
```

Among them:

- **password** indicates the password of user **sys** of the database. You can obtain the password from the database administrator.

- **host** indicates the IP address of the server where the database instance is located. Set this parameter based on the site requirements.
  - **port** indicates the port used by the database instance. Set this parameter based on the site requirements.
  - **SID** indicates the name of the CDB database where the data to be synchronized is located. Set this parameter based on the site requirements.
- e. Run the following command to allow user **c##xstrm** to connect to the XStream outbound server:
- ```
BEGIN
  DBMS_XSTREAM_ADM.ALTER_OUTBOUND(
    server_name => 'dbzxout',
    connect_user => 'c##xstrm'
  );
END;
```
- Among them:
- **dbzxout** is the outbound server name, which is configured in [3.b](#).
  - **c##xstrm** is the username for connecting to ROMA Connect. It is configured in [2.c](#).
- f. Run the following command to disconnect the database:
- ```
exit;
```

## 6.4.3 Configuring MySQL CDC (Binlog)

### Overview

Change Data Capture (CDC) enables ROMA Connect to synchronize data sources in real time and synchronously delete data tables.

This section describes how to enable the CDC function in Binlog mode for the MySQL database.

### Prerequisites

Fields of the binary type, such as TINYBLOB, BLOB, MEDIUMBLOB and LONGBLOB, cannot be collected.

### Procedure

It is recommended that the database administrator configure the CDC function. The following uses the Linux environment as an example.

1. Enable Binlog.
  - a. Use a command line tool to connect to the server where the MySQL database is located and run the following command to log in to the database as user **root**:

```
mysql -uroot -ppassword
```

**password** indicates the password of user **root** of the database. You can obtain the password from the database administrator.

- b. Run the following command to check whether Binlog is enabled for the MySQL database:  

```
show variables like 'log_bin';
```

    - If the value of **log\_bin** is **OFF**, the Binlog function is disabled. Go to the next step.
    - If the value of **log\_bin** is **ON**, the Binlog function is enabled. Run the following SQL command to check whether the parameter settings meet the requirements:  

```
show variables like '%binlog_format%';  
show variables like '%binlog_row_image%';
```

The value of **binlog\_format** must be **ROW**, and the value of **binlog\_row\_image** must be **FULL**. If yes, go to [2](#). If no, go to the next step.
  - c. Run the following command to exit the database:  

```
exit;
```
  - d. Run the following command to open the configuration file and press **i** to enter the editing mode:  

```
vi /etc/my.cnf
```
  - e. Add the following content to the file to enable Binlog:  

```
server-id = 123  
log_bin = mysql-bin  
binlog_format = row  
binlog_row_image = full  
expire_logs_days = 10  
gtid_mode = on  
enforce_gtid_consistency = on
```

Among them:
    - The value of **server-id** must be an integer greater than 1. Set this parameter based on site requirements. The value of **Server Id** set during data integration task creation must be different from the value of this parameter.
    - **expire\_logs\_days** indicates the retention period of Binlog files. Binlog files that have been retained for more than two days will be automatically deleted.
    - **gtid\_mode = on** and **enforce\_gtid\_consistency = on** are added only when the MySQL version is 5.6.5 or later. Otherwise, delete the two rows.
  - f. Press **Esc** to exit the input mode, enter **:wq**, and press **Enter** to save the settings and exit.
  - g. Run the following command to restart the MySQL database:  

```
service mysqld restart
```
  - h. Log in to the database as user **root** and run the following command to check whether the value of the **log\_bin** variable is **ON**, that is, whether the Binlog function is enabled:  

```
show variables like 'log_bin';
```
2. Run the following commands in the database to create a user for connecting ROMA Connect to the database and configure permissions for the user:  

```
CREATE USER 'roma'@'%' IDENTIFIED BY 'password';  
GRANT SELECT, RELOAD, SHOW DATABASES, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'roma'@'%';
```

Among them:

- **roma** indicates the name of the ROMA Connect connection user. Set it based on site requirements.
  - **password** indicates the password of the ROMA Connect connection user. Set it based on site requirements.
3. (Optional) If the MySQL database version is 8.0, run the following command to change the password authentication mode of the database connection user:  

```
ALTER USER roma IDENTIFIED WITH mysql_native_password BY 'password';
```

Among them:
    - **roma** is the database connection username created in 2.
    - **password** is the password of the database connection user.
  4. Run the following command to exit the database:  

```
exit;
```

## 6.4.4 Configuring SQL Server CDC

### Overview

Change Data Capture (CDC) enables ROMA Connect to synchronize data sources in real time and synchronously delete data tables.

This section describes how to enable the CDC function for the SQL Server database.

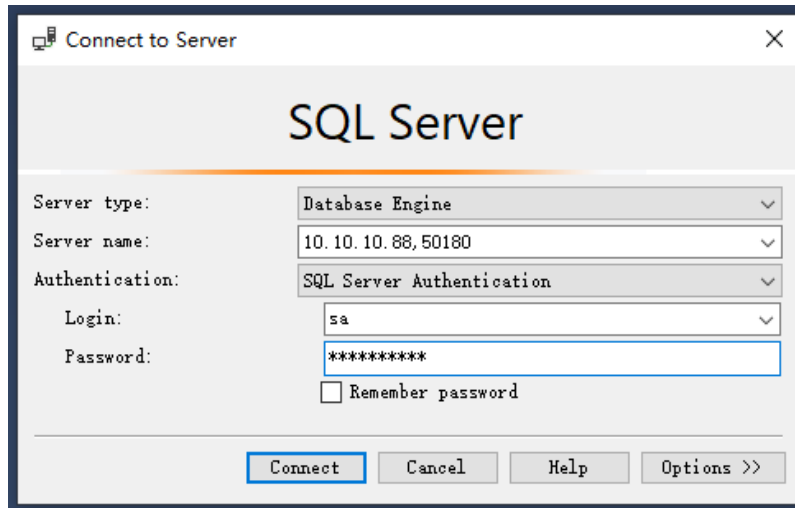
### Prerequisites

Fields of the binary type, such as BINARY, VARBINARY, and IMAGE, cannot be collected.

### Procedure

It is recommended that the database administrator configure the CDC function. The following uses the Windows environment as an example.

1. Enable CDC.
  - a. Use the SSMS client to connect to the server where the SQL Server database is located, and log in to the database as user **sa**.  
For details about how to download an SSMS client, see [Download SQL Server Management Studio \(SSMS\)](#).

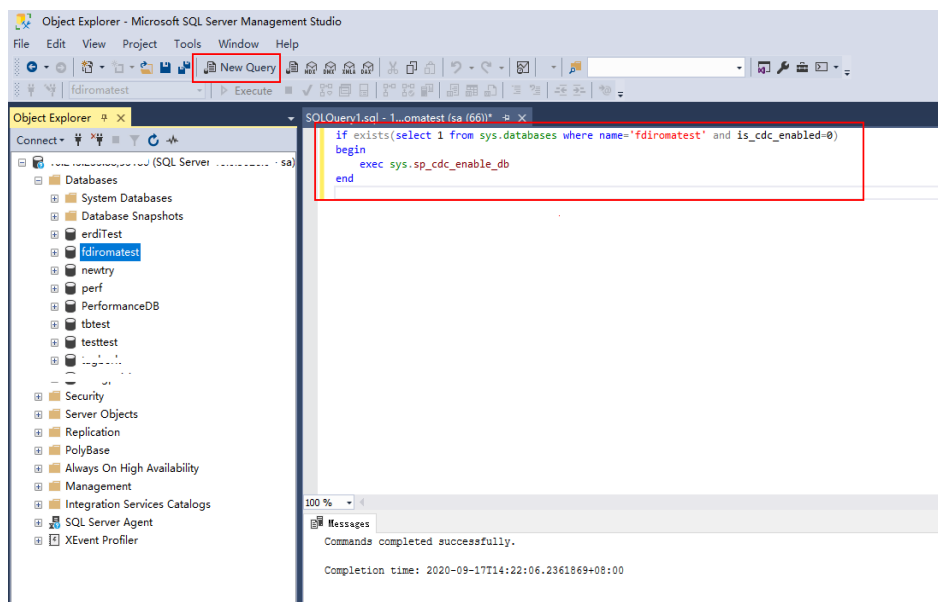


**Password** indicates the password of user **sa** of the database. You can obtain the password from the database administrator.

- b. Click **New Query** and select the database for which you want to enable CDC. Run the following command to start CDC for the SQL Server database:

```
if exists(select 1 from sys.databases where name='fdiromatest' and is_cdc_enabled=0)
begin
    exec sys.sp_cdc_enable_db
end
```

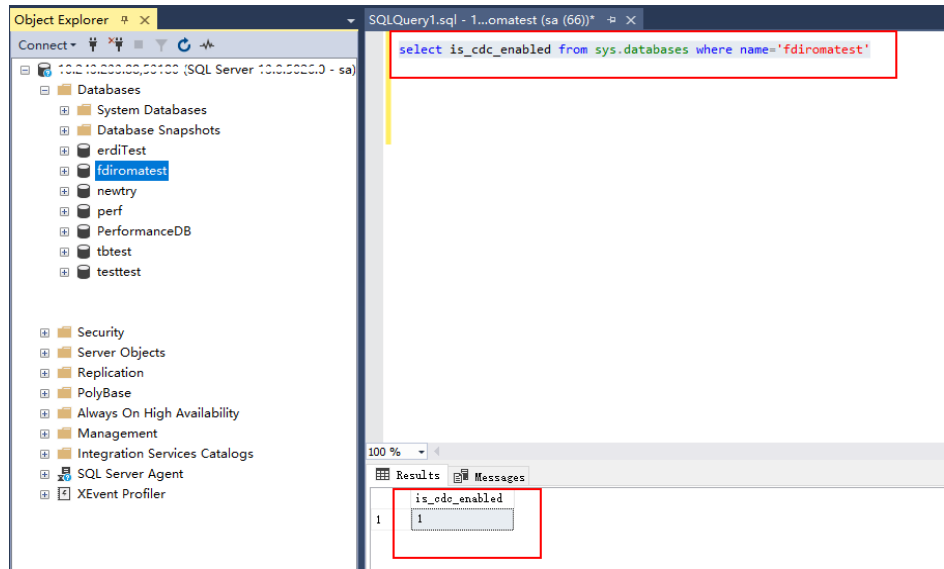
Replace *fdiromatest* with the actual database name.



- c. Run the following command to check whether CDC is successfully enabled. If **1** is displayed, CDC is successfully enabled.

```
select is_cdc_enabled from sys.databases where name='fdiromatest'
```

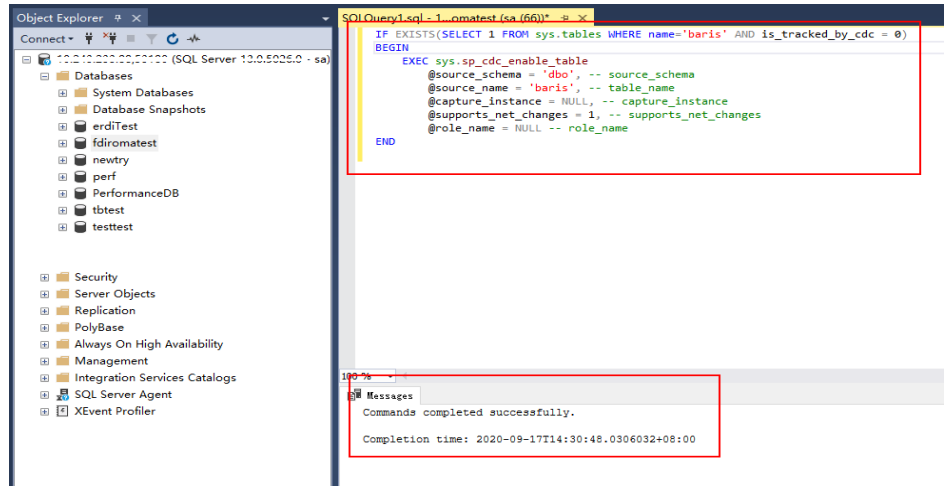
Replace *fdiromatest* with the actual database name.



d. Enable the table-level configuration.

```
IF EXISTS(SELECT 1 FROM sys.tables WHERE name='baris' AND is_tracked_by_cdc = 0)
BEGIN
EXEC sys.sp_cdc_enable_table
@source_schema = 'dbo', -- source_schema
@source_name = 'baris', -- table_name
@capture_instance = NULL, -- capture_instance
@supports_net_changes = 1, -- supports_net_changes
@role_name = NULL -- role_name
END
```

Replace *baris* with the name of the table for which CDC needs to be enabled.

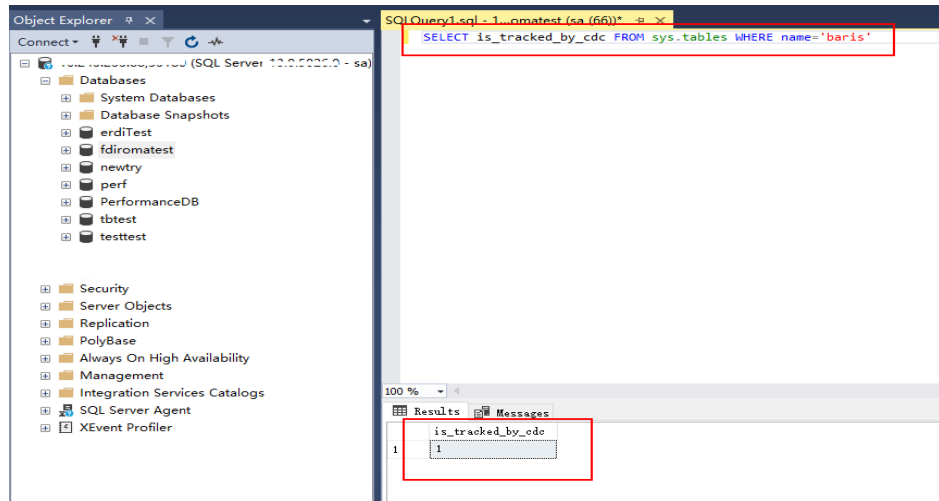


e. Run the following command to query the execution result. If the result is 1, the operation is successful.

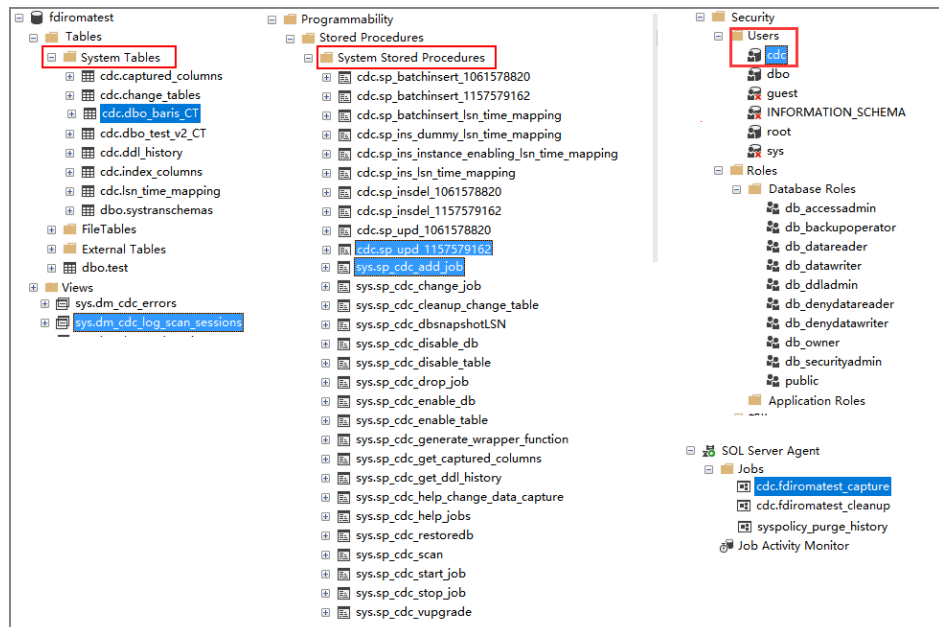
```
SELECT is_tracked_by_cdc FROM sys.tables WHERE name='baris'
```

Replace *baris* with the name of the table for which CDC needs to be enabled.





After CDC is created, the system creates a series of system catalogs, views, stored procedures, and jobs to implement CDC functions.



2. After the CDC is configured, you need to create a user and assign permissions to the user.
  - a. Right-click **Security** and choose **New > Login** from the shortcut menu.
  - b. On the **General** page, enter the created username and password. Select **SQL Server authentication** and enter the password.
  - c. On the **Server Roles** page, select **dbcreator** and **public**.
  - d. On the **User Mapping** page, select the users and database role members that are mapped to this login.
    - i. In the **Users mapped to this login** area, select the database for which you want to configure CDC, for example, **fdiromatest**.
    - ii. In the **Database role membership for** area, select **db\_datareader**, **db\_owner**, and **public**.
  - e. Click **OK**.

## Follow-Up Operations

If the system table structure changes or the table level is adjusted, you need to enable CDC again. The configuration procedure is as follows:

1. Disable CDC and set schema and name based on the site requirements.

```
EXEC sys.sp_cdc_disable_table
@source_schema = N'dbo',
@source_name = 'baris',
@capture_instance = 'all'
```

2. Enable CDC again and set schema and name based on the site requirements.

Enable the table-level configuration.

```
IF EXISTS(SELECT 1 FROM sys.tables WHERE name='baris' AND is_tracked_by_cdc = 0)
BEGIN
EXEC sys.sp_cdc_enable_table
@source_schema = 'dbo', -- source_schema
@source_name = 'baris', -- table_name
@capture_instance = NULL, -- capture_instance
@supports_net_changes = 1, -- supports_net_changes
@role_name = NULL -- role_name
END
```

## 6.4.5 Creating a Composite Task

### Overview

You can create a composite task if you need to continuously synchronize real-time data. This task allows FDI to implement real-time and incremental synchronization of multiple data tables from the source to destination, improving the data integration and synchronization efficiency.

The composite task supports flexible mappings of fields between data tables. For example, multiple fields in one data table at the source can be mapped to different data tables at the destination, or fields in multiple data tables at the source can be mapped to one data table at the destination.

### Prerequisites

- You have connected to data sources at the source and destination. For details, see [data source management](#).

In the data source configuration at the source, the value of **Database** must be the same as the actual database name (case-sensitive). Otherwise, data synchronization will fail.

- The CDC function has been enabled at the source. The CDC implementation modes vary depending on data source types. For details, see the following:
  - [Configuring Oracle CDC \(LogMiner\)](#) (recommended)
  - [Configuring Oracle CDC \(XStream\)](#)
  - [Configuring MySQL CDC \(Binlog\)](#)
  - [Configuring SQL Server CDC](#)
- The retention period of CDC archiving logs in a data source at the source must be greater than the log time parsed by the integration task. Otherwise, the integration task cannot find archive logs, resulting in incremental synchronization failures. Therefore, it is not recommended that a data

integration task be stopped for a long time. It is recommended that archive logs be retained for at least two days.

- Do not perform Data Definition Language (DDL) operations on the source database during the first data synchronization.
- If a large number of composite tasks are created, the database server and FDI plug-in process will consume resources. Therefore, you are advised not to create too many composite tasks for a database.
- You can configure multiple database tables under multiple schemas in a single CDC task to implement unified collection for full or incremental data.
- During the running of a composite task, you can add a table and perform full or incremental collection on the new table after the restart.
- Synchronization is not supported for the following types of data sources at the source:
  - Fields of the large text type and binary type
  - A data table whose name contains lowercase letters cannot be synchronized.
  - Data tables that do not have primary keys cannot be synchronized.

If a table contains a small amount of data, you are advised to collect full data once a day. Currently, data in a PostgreSQL table can be cleared before being written to the table. If data is collected from the Oracle database but no primary key is available in the table, you can use the internal RowId of the Oracle database as the primary key. The RowId is a string of 18 characters generated using digits and letters.

- Data tables or data fields whose names are reserved in the database
  - Data deleted in truncate mode cannot be synchronized. Data deleted in entire table mode cannot be synchronized.
- For the MySQL data source at the source:

If the MySQL database uses the MGR cluster mode, the source data source must be directly connected to the active node instead of the route node.

If the MySQL database contains a large amount of data, the connection to the database may time out when data is synchronized for the first time. You can modify the **interactive\_timeout** and **wait\_timeout** parameters of the MySQL database to avoid this problem.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Fast Data Integration > Task Management**. On the page displayed, click **Create Composite Task**.
3. On the **Create Composite Task** page, configure basic task information.

**Table 6-71** Basic task information

| Parameter       | Description                                                                                                                                                                                                                                                |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Task Name       | After a task is created, the task name cannot be modified. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                             |
| Description     | It is recommended that you add task descriptions based on the actual task usage to differentiate tasks. The task description can be modified after being created.                                                                                          |
| Tag             | Add a tag to classify tasks for quick search. You can select an existing tag for association. If no tag is available, click <b>Add Tag</b> to add a new tag. The new tag is saved when the task is saved. You can search for the tag when creating a task. |
| Operation Types | Select the operation types for database logs, including insert, delete, and update. For example, if you select <b>Insert</b> and <b>Update</b> , only the logs related to data insert and update in the database are obtained.                             |

4. Configure a mapping between data sources at the source and destination.

**Table 6-72** Source and destination configuration information

| Parameter   | Description             |                                                                                                                                                                                                                                                                           |
|-------------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Source      | Source Instance         | Select the ROMA Connect instance that is being used.                                                                                                                                                                                                                      |
|             | Integration Application | Select the integration application to which the data source at the source belongs.                                                                                                                                                                                        |
|             | Data Source Type        | Select a data source type at the source. The source data source type can only be MySQL, TaurusDB, Oracle, or SQL Server.                                                                                                                                                  |
|             | Data Source             | Select a data source at the source. The data source must have been created in advance.                                                                                                                                                                                    |
|             | Server ID               | This parameter is mandatory only if <b>MySQL</b> is selected as the data source type. The value must be an integer greater than <b>1</b> and must be different from the <b>server-id</b> value set in <a href="#">Configuring MySQL CDC</a> and in other composite tasks. |
| Destination | Destination Instance    | Select the ROMA Connect instance that is being used. After the source instance is configured, the destination instance is automatically associated and does not need to be configured.                                                                                    |

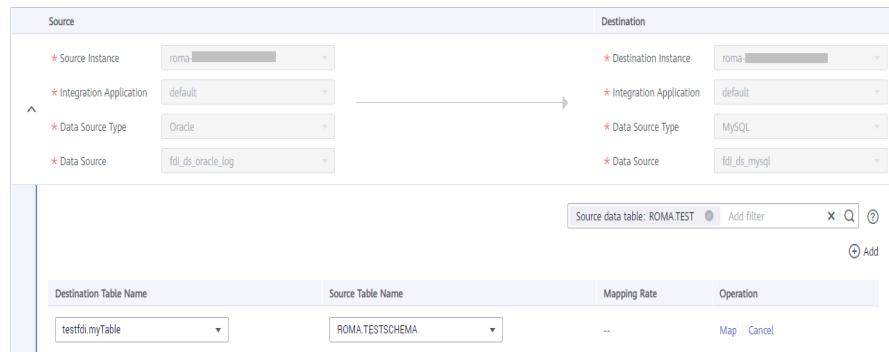
| Parameter |                         | Description                                                                                                                                                                                                     |
|-----------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | Integration Application | Select the integration application to which the data source at the destination belongs.                                                                                                                         |
|           | Data Source Type        | Select a data source type at the destination. The destination data source type can only be HANA, Kafka, MySQL, TaurusDB, Oracle, PostgreSQL, or SQL Server.                                                     |
|           | Data Source             | Select a data source at the destination. The data source must have been created in advance.                                                                                                                     |
|           | Topic Name              | This parameter is mandatory only if <b>Kafka</b> is selected as the data source type.<br><br>Select the topic whose data is to be integrated to the destination Kafka. The message data is stored in the topic. |

5. Configure data table mappings between the source and destination in manual or automatic mode.

 **NOTE**


- The length of a data field at the destination must be greater than or equal to that of the data field at the source. Otherwise, the synchronized data will be lost.
  - A maximum of 1000 data tables can be synchronized in a task.
  - If the data source type at the destination is **Kafka**, the table displayed on the destination is a virtual table. You only need to edit the field mappings in the table.
- Automatic mapping
    - i. Click **Automatic mapping**. The mappings between data tables are automatically generated.
    - ii. Click **Edit** to modify a mapping between data tables as required.
    - iii. Click **Map**. In the dialog box displayed, you can modify the mappings between fields in the data tables as required or add new mappings.  
The length of a data field at the destination must be greater than or equal to that of the data field at the source. Otherwise, the synchronized data will be lost.
  - Manually adding a mapping
    - i. Click **Add** to manually add a mapping between data tables.

**Figure 6-36** Manually adding a mapping



- ii. Select data table names for **Destination Table Name** and **Source Table Name**.

If there are a large number of data tables in the database, you can add filter criteria to filter required data tables at the source and destination.

Click the filter criteria text box, select **Destination data table** or **Source data table**, enter the data table name, and click .

- For the Oracle database, enter a filter criterion in the format of *Schema name.Data table name*. For other relational databases, enter a filter criterion in the format of *Database name.Data table name*.
  - % indicates any character string. For example, roma% indicates all data tables whose names start with roma.
  - The entered filter criteria are case sensitive.
  - You can add a filter criterion for both the destination and source tables.
- iii. Click **Map**. The dialog box for configuring the field mappings is displayed. You can click **Edit** to modify existing fields and field mappings based on the site requirements.

You can also click **Add Mapping** to add the fields to be synchronized and the field mappings. The mapping configuration items are described as follows:

- **Destination Field:** Select the corresponding field name in the destination table, for example, **ID**.
- **Source Field/Constant:** Select the field name or constant in the source table, for example, **CODE**.
- **Prefix:** Enter the prefix of the synchronization field.
- **Suffix:** Enter the suffix of the synchronization field.

The following is an example of configuring the prefix and suffix. For example, if the field content is **test**, the prefix is **tab1**, and the suffix is **1**, the field after synchronization is **tab1test1**.

**Figure 6-37** Configuring field mappings

| Destination Field | Destination FL. | Length | Source Field/Constant | Source Field Type | Length | Prefix | Suffix | Operation     |
|-------------------|-----------------|--------|-----------------------|-------------------|--------|--------|--------|---------------|
| id                | INT             | 11     | ID                    | DECIMAL           | 19     | 2      | 3      | Save   Cancel |
| name              | VARCHAR         | 100    | NAME                  | VARCHAR           | 100    | tab1   | 1      | Save   Cancel |

⊕ Add Mapping

OK Cancel

6. Configure abnormal data storage.

**NOTE**

This part is available only when the data source type at the destination is MySQL, TaurusDB, Oracle, PostgreSQL, or SQL Server. Before configuring the abnormal data storage, you need to connect to the OBS data source. For details, see [Connecting to an OBS Data Source](#).

During each task execution, if some data at the source meets integration conditions but cannot be integrated to the destination due to network jitter or other exceptions, ROMA Connect stores the data to the OBS bucket as text files.

**Table 6-73** Abnormal data storage information

| Parameter               | Description                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Source Data Type        | This parameter can only be set to <b>OBS</b> .                                                                                          |
| Integration Application | Select the required integration application.                                                                                            |
| Data Source Name        | Select the OBS data source that you configured.                                                                                         |
| Path                    | Enter the object name of the OBS data source where abnormal data is to be stored. The value of <b>Path</b> cannot end with a slash (/). |

7. Click **Save**.

**NOTE**

If any of the following cases occurs after a composite task is started, you can choose **More > Reset Task** in the **Operation** column in the task list and select date and time for task reset based on site requirements. After the task is reset, it will synchronize data again, and then checks and synchronizes incremental data in real time.

- Composite tasks need to support synchronization of new data tables and data fields at the source.
- The CDC archive logs at the source are cleared. As a result, the composite task fails to be synchronized.
- The MySQL database does not use the GTID mode, and an active/standby switchover occurs. As a result, the composite task fails to be synchronized.

You can reset the task only when **Task Status** is **Stopped**.

## 6.5 Starting or Stopping a Data Integration Task

### Overview

After you create a data integration task, **Task Status** is displayed as **Stopped** by default. You need to manually start the task by clicking **Start**.

- After you start a scheduled task, ROMA Connect integrates data on a scheduled basis. During the first execution, all source data that meets the conditions is integrated to the destination. Then, full data that meets the conditions or only incremental data will be integrated based on the task configuration.
- After you start a real-time task, ROMA Connect continuously detects data changes at the source. During the first execution, all source data that meets the conditions is integrated to the destination. Subsequently, only new data will be integrated to the destination each time.

#### NOTE

- If two data integration tasks use MRS data sources of different versions (including MRS Hive, MRS HDFS, and MRS HBase) and Kerberos authentication is enabled for the MRS data sources, the two data integration tasks cannot be executed at the same time. Otherwise, the integration tasks fail.
- In the task list, **Executed** indicates the time when the schedule of a created task starts to take effect.

If you need to perform other operations, such as modification, on a started task, you must stop the task first.

### Starting a Task

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **Fast Data Integration > Task Management**.
3. On the **Task Management** page, filter the task to be started based on the integration mode, task status, or task name.

You can start the task by using either of the following methods:

- **Clicking Start:** The task runs according to the configured integration mode.

Select the task to be started and click **Start** above the task list. After the task is started, **Task Status** changes to **Started**.

- **Selecting Manual Scheduling:** Before manually scheduling a task, ensure that **Task Status** is **Stopped**.

Choose **More > Manual Scheduling** on the right of the data integration task to manually schedule the task.

- After a scheduled task is manually scheduled, the task is executed only once, and then the task status changes to **Stopped**.



- After a real-time task is manually scheduled, the task is started and the task status changes to **Started**. The effect of manually scheduling a real-time task is the same as that of starting a scheduled task directly.

## Stopping a Task

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **Fast Data Integration > Task Management**.
3. On the **Task Management** page, filter the task to be stopped based on the integration mode, task status, or task name.
4. Select the task and click **Stop** to stop the task.

# 6.6 Managing a Data Integration Task

## 6.6.1 Viewing Data Integration Tasks

### Overview

After you create a data integration task, you can view the task information, including the task configuration information, run logs, operation logs, and scheduling logs.

You can also modify the task configuration. The procedure for modifying a task is similar to that for creating a task. For details, see [Creating a Data Integration Task](#).

---

#### NOTICE

Exercise caution when modifying task configuration information to avoid dirty data or incomplete data integration at the destination. Especially, modifying source or destination information may cause changes to the data integration solution.

---

### Viewing Task Information



1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **Fast Data Integration > Task Management**.
3. On the **Task Management** page, click a task name. On the task details page, view related configuration and log information.

You can filter tasks by **Integration Mode**, **Task Status**, **Task Name**, or other criteria.

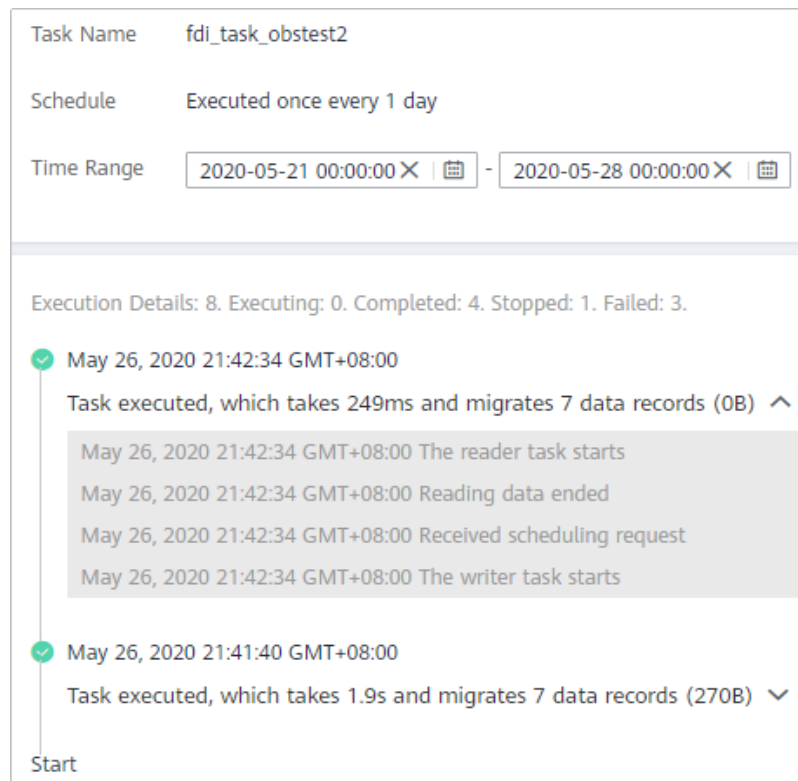
**Table 6-74** Task information

| Content          | Description                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Task Information | Basic task information, including the task name, task ID, integration mode, and task status.                                                                                                                                                                                                                                                                                                                |
| Task Details     | Task configuration information, including the source information, destination information, and mapping information (not available for a combined task).                                                                                                                                                                                                                                                     |
| Schedule         | Schedule information of a task. Only scheduled tasks have schedule information. For real-time tasks, no schedule information is involved.<br>Click <b>Edit</b> to modify the schedule configuration.                                                                                                                                                                                                        |
| Logs             | Includes run logs and operation logs of tasks. <ul style="list-style-type: none"> <li>• A run log records information about the task execution, including the schedule execution time, data reading time, data volume, and duration of each phase.</li> <li>• An operation log records the management operations of tasks, including creating, updating, deleting, starting, and stopping tasks.</li> </ul> |

## Viewing Scheduling Details

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **Fast Data Integration > Schedule Monitoring**.
3. On the **Schedule Monitoring** page, filter information such as the running status, scheduling mode, scheduling period, and latest execution time based on the task status and task name.  
For a composite task, you can also view the synchronization progress, time for reading full data in the table, and other information.
4. Click **View Log** on a task to view the detailed scheduling logs of the task.
  - You can specify **Time Range** to view scheduling logs in a specified period.
  - You can click  (detail mode) or  (list mode) on the right to switch the log display mode.

**Figure 6-38** Scheduling log



## 6.6.2 Importing or Exporting Data Integration Tasks

### Overview

You can import or export data integration tasks.

### Exporting Data Integration Tasks

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **Fast Data Integration > Task Management**.
3. On the **Task Management** page, select the required tasks and click **Export Task**. The system automatically exports the selected tasks to the local PC.

### Importing Data Integration Tasks

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **Fast Data Integration > Task Management**.
3. On the **Task Management** page, click **Import Task** and select the file to import.

# 7 Service Integration Guide

---

[Usage Introduction](#)

[Exposing APIs](#)

[Creating and Exposing Data APIs](#)

[Creating and Exposing Function APIs](#)

[Calling an API](#)

[API Management](#)

[Control Policy Management](#)

[Configuring a Custom Authorizer](#)

[Configuring Signature Verification for Backend Services](#)

[Configuring API Cascading](#)

## 7.1 Usage Introduction

### Function Description

APIC is the API integration component of ROMA Connect. It encapsulates APIs, data sources, and custom functions into RESTful APIs, and then exposes them to external systems. ROMA Connect has the following advantages for service integration:

- **Convenient API lifecycle management**

ROMA Connect provides full-lifecycle management for APIs, including creating, debugging, publishing, taking offline, authorizing, editing, and deleting APIs.

- **Custom API backend services**

ROMA Connect provides two types of backends:

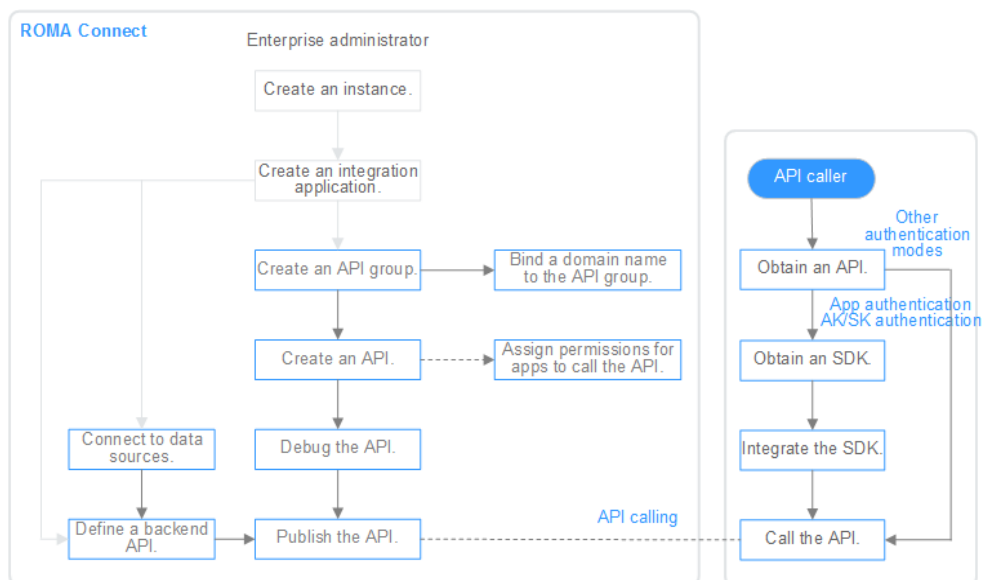
- Data backends: Data sources are exposed as APIs. For details about the supported data source types, see [Data Sources Supported by APIC](#).

- Function backends: Function capabilities are exposed as APIs.
- **Visualized API monitoring portal**  
ROMA Connect provides a visualized dashboard for API calling and analysis to help you monitor the performance metrics related to API calling and identify potential risks that may affect services.
- **Multi-layer security protection**  
ROMA Connect provides multiple authentication modes, refined request throttling, and strict access control to ensure secure API calling.

## Process Flow

The following figure shows the process of using ROMA Connect for service integration.

**Figure 7-1** Process of using ROMA Connect for service integration



1. You have **created an instance and integration application.**
2. **Expose an API.**
  - Exposing APIs
    - i. **Create an API group.**  
Each API belongs to an API group. Before creating an API, create an API group.
    - ii. **Bind a domain name to the API group.**  
Before exposing an API, you need to bind an independent domain name to the API group so that users can access the API.  
Before binding an independent domain name to an API, you can use the default subdomain name allocated by the system to test API calling. ROMA Connect limits the number of times that the default subdomain name can be accessed. A maximum of 1000 times can be accessed every day.

- iii. **Create an API.**  
Encapsulate existing backend services into RESTful APIs and expose them to external systems.
  - iv. **Debug the API.**  
Verify that the API service functions are normal using the debugging function provided by ROMA Connect.
  - v. **Publish the API.**  
Publish an API in an environment. The API can be called only after being published in the environment.
  - vi. (Optional) **Grant permissions for APIs.**  
This operation is required only for APIs that use the APP authentication mode. After an API is authorized to a specified integration application, users can use the key and secret of the integration application to perform security authentication on API requests.
- Creating and exposing a data API
    - i. **Connect to data sources.**  
Connect to a data source to ensure that data can be read from the data source.
    - ii. **Create a data API.**  
Define data sources as APIs and expose them to external systems through ROMA Connect.
    - iii. (Optional) **Grant permissions for APIs.**  
This operation is required only for APIs that use the APP authentication mode. After an API is authorized to a specified integration application, users can use the key and secret of the integration application to perform security authentication on API requests.
  - Creating and exposing a function API
    - i. **Create a function API.**  
Define custom functions as APIs and expose them to external systems.
    - ii. (Optional) **Grant permissions for APIs.**  
This operation is required only for APIs that use the APP authentication mode. After an API is authorized to a specified integration application, users can use the key and secret of the integration application to perform security authentication on API requests.
3. **Call the API.**  
Obtain the API and its access address to call the API. API calling requires different authentication operations based on the authentication mode used by the API.

## 7.2 Exposing APIs

## 7.2.1 Creating an API Group

### Overview

An API group is a set of APIs used for the same service. An API developer can create an API group to manage all APIs in the group. Each API belongs to an API group. Before creating an API, create an API group.

### Prerequisites

Each API group must belong to an integration application. Before creating an API group, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **API Groups** tab page, click **Create**.
3. In the **Create API Group** dialog box, enter the group-related parameters and click **OK**.

**Table 7-1** API group parameters

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter an API group name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                  |
| Type                    | Type of the API group. <ul style="list-style-type: none"> <li>• <b>Integration application:</b> An API group belongs to a specific integration application. Only users who have permissions on the integration application can view and perform operations on the API group.</li> <li>• <b>Global:</b> All users can view and perform operations on the API group.</li> </ul> |
| Integration Application | This parameter is mandatory only if <b>Integration application</b> is selected for <b>Type</b> .<br>Select the integration application to which the API group belongs. If no integration application is available, click <b>Create Integration Application</b> on the right to create an integration application.                                                             |
| Description             | Enter a brief description of the API group.                                                                                                                                                                                                                                                                                                                                   |

 **NOTE**

- The system automatically allocates a subdomain name to the API group for internal testing. The subdomain name can be accessed up to 1000 times a day.
- When opening APIs, you must bind an independent domain name to the API group.

## 7.2.2 Binding Domain Names

### Overview

Before you expose an API, you need to bind an independent domain name to the API group so that users can use the independent domain name to access the API.

For internal testing purposes, you can use the default subdomain name allocated by the system to access APIs in the API group a maximum of 1000 times a day. You cannot modify the subdomain name.

### Prerequisites

- You have obtained an independent domain name. .
- An independent domain name has been resolved to the subdomain name of the APIC connection address. For details, see "Adding a CNAME Record Set" in the *Domain Name Service User Guide*.
- If the APIs in the API group support HTTPS, you need to add an SSL certificate for the independent domain name. You need to obtain the content and key of the SSL certificate in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **API Groups** tab page, click **More > Manage Domain Name** in the **Operation** column of the target API group.
3. On the **Domain Names** tab page, click **Bind Independent Domain Name**.
4. In the **Bind Independent Domain Name** dialog box displayed, enter the domain name and click **OK**.

**Table 7-2** Independent domain name configuration

| Parameter   | Description                                                 |
|-------------|-------------------------------------------------------------|
| Domain Name | Enter the domain name to be bound.                          |
| TLS Version | Select the minimum TLS version used for domain name access. |

5. If the APIs in the API group support HTTPS, add an SSL certificate for the independent domain name. Otherwise, skip this step.
  - a. Click **Add SSL Certificate** on the right of the independent domain name.



- b. In the **Add SSL Certificate** dialog box, configure certificate information and click **OK**.

**Table 7-3** SSL certificate configuration

| Parameter | Description                                      |
|-----------|--------------------------------------------------|
| Name      | Enter the name of a new certificate.             |
| Content   | Enter the SSL certificate content in PEM format. |
| Key       | Enter the SSL certificate key in PEM format.     |

**NOTE**

The certificate content and key will not be visible after you click **OK**. If you need to modify the certificate content or key, delete the certificate and add it again.

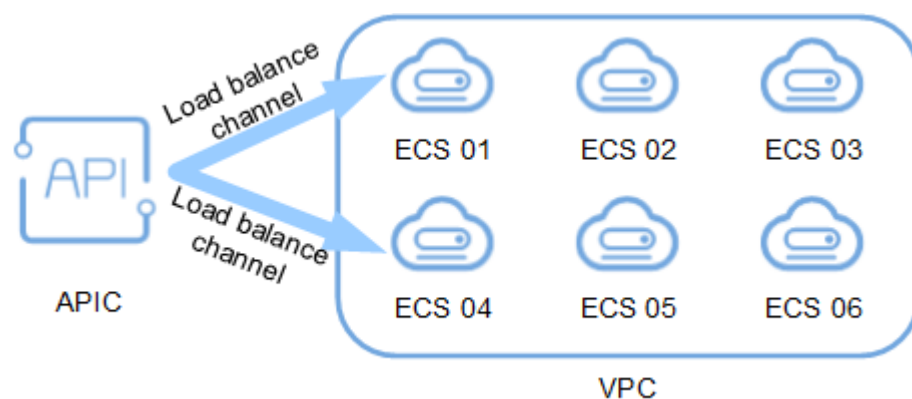
## 7.2.3 (Optional) Creating a Load Balance Channel

### Overview

A load balance channel allows ROMA Connect to access backend services deployed on a specified server in load balancing mode. ROMA Connect can access ECSs on the same cloud service platform as well as your own private servers.

For example, six ECSs have been deployed in a VPC, and a load balance channel has been created to reach ECS 01 and ECS 04. ROMA Connect can access these two ECSs through the load balance channel.

**Figure 7-2** Accessing the ECSs in a load balance channel



### Prerequisites

The network between ROMA Connect and the servers in the load balance channel is normal.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **Load Balance Channels** tab page, click **Create**.
3. On the **Create Load Balance Channel** page, configure basic information and health check information and click **Next**.

**Table 7-4** Parameters in Basic Information and Health Check Configuration

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                        |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                   | Enter a load balance channel name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                             |
| Port                   | Enter the access port number of the ECS in the load balance channel.                                                                                                                                                                                                                                                                                               |
| Member Type            | Select the member type of the ECS in the load balance channel. The member type cannot be changed after the load balance channel is created. <ul style="list-style-type: none"> <li>• <b>Instance:</b> Add a load balance channel member by selecting an ECS.</li> <li>• <b>IP address:</b> Add a load balance channel member by entering an IP address.</li> </ul> |
| Distribution Algorithm | Select the distribution algorithm for backend service requests. The load balance channel determines the ECS to which the request is to be sent based on the distribution algorithm.                                                                                                                                                                                |
| Protocol               | Select the protocol used for the health check. The value can be <b>TCP</b> , <b>HTTP</b> , or <b>HTTPS</b> .                                                                                                                                                                                                                                                       |
| Path                   | This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTP</b> or <b>HTTPS</b> .<br>Enter the health check URL.                                                                                                                                                                                                                                         |
| Method                 | This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTP</b> or <b>HTTPS</b> .<br>Select the HTTP request method used for the health check. The value can be <b>GET</b> or <b>HEAD</b> .                                                                                                                                                              |
| Advanced Settings      |                                                                                                                                                                                                                                                                                                                                                                    |
| Health Check Port      | Enter the destination port of the health check. By default, the port number configured for the load balance channel is used.                                                                                                                                                                                                                                       |

| Parameter           | Description                                                                                                                                                                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Healthy Threshold   | Number of consecutive successful checks required for an ECS to be considered healthy. For example, if <b>Healthy Threshold</b> is set to <b>2</b> , ROMA Connect considers the ECS status as healthy when the check is successful for two consecutive times.           |
| Unhealthy Threshold | Number of consecutive failed checks required for an ECS to be considered unhealthy. For example, if <b>Unhealthy Threshold</b> is set to <b>5</b> , ROMA Connect considers the ECS status as abnormal when the check fails for five consecutive times.                 |
| Timeout Interval    | Enter the response timeout duration of a health check, in the unit of seconds. If no response is received within the specified duration, the health check fails.                                                                                                       |
| Interval (s)        | Enter the interval between consecutive checks.                                                                                                                                                                                                                         |
| Response Codes      | This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTP</b> or <b>HTTPS</b> .<br><br>When the server returns a specified HTTP response code, the server considers the response as successful. Multiple response codes can be specified at the same time. |

4. Add ECSs or backend server addresses to the load balance channel based on the value of **Member Type** set for the load balance channel.
  - Adding an ECS
    - i. Click **Select ECS**.
    - ii. In the **Select ECS** dialog box, select a subnet, select the ECS to be added, and click **OK**.
  - Adding a backend server address
    - i. Click **Add Backend Server Address**.
    - ii. On the page displayed, configure backend server information.

**Table 7-5** Backend server information

| Parameter              | Description                                 |
|------------------------|---------------------------------------------|
| Backend Server Address | Enter the IP address of the backend server. |

| Parameter | Description                                                                                                                                                                                                                                                                  |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Weight    | This parameter is mandatory only if <b>WRR</b> or <b>WLC</b> is selected for <b>Distribution Algorithm</b> of the load balance channel.<br><br>Enter the weight of the backend server. A larger weight indicates that more requests will be forwarded to the backend server. |

5. Click **Finish**.

## 7.2.4 Creating an API

### Overview

Create APIs to encapsulate existing backend services into standard RESTful APIs and make them accessible to other users.

### Prerequisites

- Each API must belong to an integration application. Before creating an API, ensure that an integration application is available. Otherwise, [create an integration application](#) first.
- Each API must belong to an API group. Before creating an API, ensure that the API group is available. Otherwise, [create an API group](#) first.
- If you need to use a load balance channel to access the ECS where the backend service is located, [create a load balance channel](#) first.
- If you need to use the custom mode for API security authentication, [create a custom authorizer](#) first.

### Setting Basic Information

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, click **Create API**.
3. On the **Create API** page, configure basic information about the API.

**Table 7-6** Basic API information

| Parameter               | Description                                                                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter an API name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                  |
| Integration Application | Select the integration application to which the API belongs. If no integration application is available, click <b>Create Integration Application</b> on the right to create an integration application. |

| Parameter                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Group                 | Select the API group to which the API belongs. If no API group is available, click <b>Create API Group</b> on the right to create an API group.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Security Authentication   | Select the authentication mode of the API. The App authentication mode is recommended. <ul style="list-style-type: none"> <li>• <b>App</b>: ROMA Connect authenticates API requests. When a user calls an API, the key and secret of the integration application are used for authenticating API requests. APIs using this mode can be called by all users.</li> <li>• <b>IAM</b>: IAM authenticates API requests. When a user calls an API, the token or AK/SK is used for authenticating API requests. APIs using this mode can be called only by users on the same cloud service platform.</li> <li>• <b>Custom</b>: The custom function API is used for authenticating API requests. APIs using this mode can be called by all users.</li> <li>• <b>None</b>: Authentication is not required for API requests. APIs using this mode can be called by all users.</li> </ul> |
| Simple Authentication     | This parameter is available only if <b>Security Authentication</b> is set to <b>App</b> .<br>This parameter specifies whether simple security authentication is used for API calling. This parameter is valid only when the API request protocol is HTTPS. If this parameter is enabled, AppCode is carried for security authentication when an API is called, and signature verification is not required for API requests.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Two-Factor Authentication | This parameter is available only if you have set <b>Security Authentication</b> to <b>App</b> or <b>IAM</b> .<br>This parameter specifies whether to perform two-factor authentication on API calling. If this parameter is selected, the custom function API is also used to authenticate API requests when <b>APP</b> or <b>IAM</b> has been selected for <b>Security Authentication</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Custom Authorizer         | This parameter is mandatory only if you have set <b>Security Authentication</b> to <b>App</b> or <b>IAM</b> and have selected <b>Two-Factor Authentication</b> , or only if you have set <b>Security Authentication</b> to <b>Custom</b> .<br>Select a custom authorizer you have created. If no custom authorizer is available, click <b>Create Custom Authorizer</b> on the right to create a frontend custom authorizer.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Tag Name                  | Add tags for the API to quickly filter and search for the API.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Description               | Enter a brief description of the API.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

4. Click **Next**.

## Defining API Request

1. Define request information of an API.

**Table 7-7** API request configuration

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Request Protocol | Select the request protocol used by the API. The value can be <b>HTTP</b> , <b>HTTPS</b> , or <b>HTTP&amp;HTTPS</b> . <b>HTTPS</b> is recommended for transmitting important or sensitive data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Path             | <p>Enter the request path of the API, in the <i>/getUserInfo/{userId}</i> format. The request path can contain the path parameter in the format of <i>{Parameter name}</i>.</p> <ul style="list-style-type: none"> <li>• The value of the path parameter must match the entire segment between slashes (/). For example, <i>/abc{userId}</i> is not supported. If <b>Matching</b> is set to <b>Exact match</b>, a plus sign (+) can be added to the end of the path parameter, for example, <i>/users/{p+}</i>. The variable <i>p</i> matches the segments between one or multiple pairs of slashes (/).</li> <li>• Path parameters contained in the request path must be defined as input parameters.</li> <li>• The value of <b>Path</b> is case sensitive.</li> </ul>                                                                                                                                                                                                                               |
| Matching         | <p>Select the matching mode of the API request path.</p> <ul style="list-style-type: none"> <li>• <b>Exact match</b>: The path in an API request must be the same as the value of <b>Path</b>.</li> <li>• <b>Prefix match</b>: The path in an API request must be prefixed with the value of <b>Path</b>. For example, if <b>Path</b> is set to <i>/test/AA</i> and <b>Matching</b> is set to <b>Prefix match</b>, the API can be accessed using <i>/test/AA/BB</i> or <i>/test/AA/CC</i> but cannot be accessed using <i>/test/AACC</i>.</li> </ul> <p><b>NOTE</b><br/>When <b>Matching</b> is set to <b>Prefix match</b>, the rest characters of the API access address except the matched prefix are transparently transmitted to the backend service. For example, if the request path is set to <i>/test</i> and the backend request path is set to <i>/test2</i>, the request path received by the backend service is <i>/test2/AA/CC</i> when <i>/test/AA/CC</i> is used to access the API.</p> |
| Method           | Select the request method of the API. <b>ANY</b> indicates that the API can be accessed using any request method.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CORS      | <p>This parameter specifies whether CORS is supported for the API.</p> <p>For security purposes, the browser restricts the cross-domain requests from being initiated from a page script. In this case, the page can access only the resources from the same source. CORS allows the browser to send XMLHttpRequest requests to the server in a different domain. For details about CORS, see <a href="#">Configuring CORS for APIs</a>.</p> |

| Parameter                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Optional)<br>Input<br>Parameters | <p>Define API request parameters based on the site requirements. Parameters contained in the request path must be defined as input parameters.</p> <p>In the <b>Input Parameters</b> area, click <b>Add Input Parameter</b> and add request parameters of the API.</p> <ul style="list-style-type: none"> <li>• <b>Name:</b> name of a request parameter. If <b>Location</b> is set to <b>PATH</b>, the parameter name must be the same as that in the request path.</li> <li>• <b>Location:</b> position of the request parameter in the API request. The value can be <b>PATH</b>, <b>HEADER</b>, or <b>QUERY</b>.</li> <li>• <b>Type:</b> data type of the request parameter. The value can be <b>STRING</b> or <b>NUMBER</b>.</li> <li>• <b>Mandatory:</b> specifies whether a request parameter is mandatory in an API request.</li> <li>• <b>Passthrough:</b> indicates whether to transparently transmit request parameters to the backend service.</li> <li>• <b>Default Value:</b> The default value of the request parameter can be set only when <b>Mandatory</b> is set to <b>No</b>.</li> <li>• <b>Enumerated Value:</b> enumerated value of the request parameter. The value of the request parameter can only be selected from the enumerated values. Use commas (,) to separate multiple enumerated values.</li> <li>• <b>Max. Length/Max. Value:</b> If <b>Type</b> is set to <b>STRING</b>, set this parameter to the maximum character string length as the parameter value. If <b>Type</b> is set to <b>NUMBER</b>, set this parameter to the maximum value as the parameter value.</li> <li>• <b>Min.Length/Min.Value:</b> If <b>Type</b> is set to <b>STRING</b>, set this parameter to the minimum string length as the parameter value. If <b>Type</b> is set to <b>NUMBER</b>, set this parameter to the minimum value as the parameter value.</li> <li>• <b>Example:</b> example of a request parameter value.</li> <li>• <b>Description:</b> description of a request parameter.</li> </ul> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• The parameter name is not case-sensitive. It cannot be <b>x-stage</b> or start with <b>x-apig-</b> or <b>x-sdk-</b></li> <li>• If <b>Location</b> is specified as <b>HEADER</b>, ensure that the parameter name is not <b>Authorization</b> or <b>X-Auth-Token</b> and does not contain underscores (_).</li> </ul> |
| Body                              | <p>This parameter is available only if <b>Method</b> is set to <b>POST</b>, <b>PUT</b>, <b>PATCH</b>, or <b>ANY</b>.</p> <p>Enter the description of the request body in the API request to help the API caller understand how to correctly encapsulate the API request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |



2. Click **Next**.

## Defining Backend Requests

1. Configure the basic definition of the default backend.

**Table 7-8** Backend service access parameters

| Backend Type | Parameter            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HTTP/HTTPS   | Protocol             | Select the request protocol used by the backend service. WebSocket is supported for HTTP and HTTPS. <b>HTTPS</b> is recommended for transmitting important or sensitive data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|              | Method               | Select the request method of a backend service. <b>ANY</b> indicates that the backend service can be accessed using any request method.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|              | Load Balance Channel | This parameter specifies whether to use the load balance channel to access backend services. If you select <b>Configure now</b> , you need to <a href="#">create a load balance channel</a> in advance.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|              | Backend Address      | <p>This parameter is mandatory only if <b>Load Balance Channel</b> is set to <b>Do not configure</b>.</p> <p>Enter the access address of the backend service in the format of <i>Host:Port</i>. <i>Host</i> indicates the IP address or domain name for accessing the backend service. If no port is specified, port 80 is used for HTTP by default, and port 443 is used for HTTPS by default.</p> <ul style="list-style-type: none"> <li>• If the backend address needs to contain environment variables, use <i>#Variable name#</i> to add the environment variables to the backend address, for example, <b>#ipaddress#</b>. Multiple environment variables can be added, for example, <b>#ipaddress##test#</b>.</li> <li>• If you want to call a custom backend by using its backend request address, add two built-in gateway parameters to <a href="#">system parameters</a>. <ul style="list-style-type: none"> <li>– <b>apild</b>: Set <b>Backend Parameter Name</b> to <b>x-auth-app</b> and <b>Backend Parameter Location</b> to <b>HEADER</b>.</li> <li>– <b>providerAppId</b>: Set <b>Backend Parameter Name</b> to <b>x-ld-appid</b> and <b>Backend Parameter Location</b> to <b>HEADER</b>.</li> </ul> </li> </ul> |

| Backend Type | Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | Load Balance Channel   | This parameter is mandatory only if <b>Load Balance Channel</b> is set to <b>Configure now</b> .<br>Select the load balance channel used to access the backend service.                                                                                                                                                                                                                                                                            |
|              | Host Header            | This parameter is available only if <b>Load Balance Channel</b> is set to <b>Configure now</b> .<br>Define the Host header field carried in the backend service request.                                                                                                                                                                                                                                                                           |
|              | Path                   | Enter the request path of the backend service, in the <code>/getUserInfo/{userId}</code> format. The request path can contain the path parameter in the format of <code>{Parameter name}</code> .<br><br>If the path needs to contain an environment variable, enclose the environment variable in number signs (#), for example, <code>/#path#</code> . Multiple environment variables can be added, for example, <code>/#path##request#</code> . |
|              | Timeout (ms)           | Enter the timeout interval of a backend service request. The default value is <b>5000</b> .                                                                                                                                                                                                                                                                                                                                                        |
|              | Two-way Authentication | This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTPS</b> .<br>Determine whether to enable two-way authentication between ROMA Connect and backend services. If you enable this option, you also need to configure the certificate for client authentication.                                                                                                                                                                     |
|              | Backend Authentication | Determine whether to enable backend authentication. If this option is enabled, a custom function API is used to perform security authentication on backend service requests.                                                                                                                                                                                                                                                                       |
|              | Custom Authorizer      | This parameter is mandatory only if <b>Backend Authentication</b> is enabled.<br>Select a custom authorizer you have created. If no custom authorizer is available, click <b>Create Custom Authorizer</b> on the right to create a custom authorizer of the backend type.                                                                                                                                                                          |

| Backend Type | Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mock         | Response               | If the backend service is unavailable, you can use the Mock mode to return the expected result to the API caller for debugging and verification. For example, if <b>Response</b> is set to <b>Successful Info</b> , the API always returns <b>Successful Info</b> as the response result no matter what request parameters are set. The Mock mode is generally used in the API development and debugging phase. |
|              | Backend Authentication | Determine whether to enable backend authentication. If this option is enabled, a custom function API is used to perform security authentication on backend service requests.                                                                                                                                                                                                                                    |
|              | Custom Authorizer      | This parameter is mandatory only if <b>Backend Authentication</b> is enabled.<br>Select a custom authorizer you have created. If no custom authorizer is available, click <b>Create Custom Authorizer</b> on the right to create a custom authorizer of the backend type.                                                                                                                                       |

 **NOTE**

- If an environment variable is defined in the backend request path, the API cannot be debugged.
  - For environment variables defined in the backend request path, corresponding variables and their values must be created. Otherwise, the APIs cannot be published because there will be no values that can be assigned to the variables.
  - Environment variable names are case-sensitive.
2. (Optional) Configure the backend parameters for the default backend to map the request parameters transferred when the API is called to the corresponding location in the backend service request. If no request parameter is defined in the API request, skip this step.
    - a. In the **Backend Parameters** area, add backend parameters in either of the following ways:
      - Click **Import Input Parameter** to add all defined API request parameters to the backend parameters.
      - Click **Add Backend Parameter Mapping** to add backend parameters one by one as required.
    - b. Modify the mapping between API request parameters and backend parameters.
      - The name and location of the backend parameters can be different from those of the input parameters.

- If **Backend Parameter Location** is **PATH**, the backend parameter name must be the same as the parameter name in **Path**.
- The backend parameter name is not case-sensitive. It cannot be **x-stage** or start with **x-apig-** or **x-sdk-**
- If **Backend Parameter Location** is **HEADER**, the backend parameter name is case-insensitive and cannot contain underscores (\_).

The parameters and backend request path in the following table are used as an example. Parameters **test01** and **test03** are located in **PATH** and **QUERY** in an API request. Through parameter mappings, the backend service receives the values of **test01** and **test03** in **HEADER**. Parameter **test02** is located in **HEADER** in an API request. Through the parameter mapping, the backend service receives the value of **test02** using **test05** in **HEADER**.

| Input Parameter Name                    | Input Parameter Location | Backend Parameter Name | Backend Parameter Location |
|-----------------------------------------|--------------------------|------------------------|----------------------------|
| test01                                  | PATH                     | test01                 | HEADER                     |
| test02                                  | HEADER                   | test05                 | PATH                       |
| test03                                  | QUERY                    | test03                 | HEADER                     |
| Backend request path: /apitest/{test05} |                          |                        |                            |

Assume that test01 is aaa, test02 is bbb, and test03 is ccc.

The API request is as follows:

```
curl -ik -H 'test02:bbb' -X GET https://example.com/v1.0/aaa?test03=ccc
```

The backend service request is as follows:

```
curl -ik -H 'test01:aaa' -H 'test03:ccc' -X GET https://example.com/v1.0/bbb
```

3. (Optional) Configure constant parameters of the default backend. Constant parameters can be defined to receive fixed constants. When sending a request to a backend service, ROMA Connect adds constant parameters to the specified location in the request and then sends the request to the backend service.

In the **Constant Parameters** area, click **Add Constant Parameter** to add the constant parameters of the backend service request.

**Table 7-9** Constant parameters

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name        | Enter the name of a constant parameter. If <b>Location</b> is <b>PATH</b> , the parameter name must be the same as that in the backend request path.<br><b>NOTE</b> <ul style="list-style-type: none"><li>The parameter name is not case-sensitive. It cannot be <b>x-stage</b> or start with <b>x-api-</b> or <b>x-sdk-</b></li><li>If <b>Location</b> is <b>HEADER</b>, the parameter name is case-insensitive and cannot contain underscores (_).</li></ul> |
| Location    | Select the location of the constant parameter in the backend service request. The value can be <b>PATH</b> , <b>HEADER</b> , or <b>QUERY</b> .                                                                                                                                                                                                                                                                                                                 |
| Value       | Enter the value of the constant parameter.                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Description | Enter the description of the constant parameter.                                                                                                                                                                                                                                                                                                                                                                                                               |

 **NOTE**

- ROMA Connect sends requests containing constant parameters to backend services after percent-encoding of special parameter values. Ensure that the backend services support percent-encoding. For example, parameter value **[api]** becomes **%5Bapi%5D** after percent-encoding.
  - For values of path parameters, ROMA Connect will the following percent-encode the following characters: ASCII codes 0–31, blank symbols, ASCII codes 127–255, and special characters `?></%#"[\]^`{}`
  - For values of query parameters, ROMA Connect will the following percent-encode the following characters: ASCII codes 0–31, blank symbols, ASCII codes 127–255, and special characters `>=<+&%#"[\]^`{}`
4. (Optional) Configure system parameters of the default backend. If a backend service needs to receive parameter information generated during system running, such as gateway built-in parameters, frontend authentication parameters, and backend authentication parameters, you can set system parameters. When sending a request to a backend service, ROMA Connect adds system parameters to the specified location in the request and then sends the request to the backend service.

In the **System Parameters** area, click **Add System Parameter** to add the system parameters of the backend service request.

**Table 7-10** System parameters

| Parameter                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Parameter Type      | <p>Select the type of a system parameter.</p> <ul style="list-style-type: none"> <li>• <b>Default gateway parameter:</b> Parameters that can be configured for ROMA Connect.</li> <li>• <b>Frontend authentication parameter:</b> Parameters to be displayed in the frontend custom authentication result. This option is available only if <b>Custom</b> is selected for <b>Security Authentication</b> in the step of <a href="#">setting basic information</a>.</li> <li>• <b>Backend authentication parameter:</b> Parameters to be displayed in the backend custom authentication result. This option is available only if <b>Backend Authentication</b> is enabled in the step of <a href="#">defining backend request</a>.</li> </ul> |
| System Parameter Name      | <p>Enter the name of a system parameter.</p> <ul style="list-style-type: none"> <li>• If <b>System Parameter Type</b> is <b>Default gateway parameter</b>, select the parameters that can be obtained by the system.</li> <li>• If <b>System Parameter Type</b> is set to <b>Frontend authentication parameter</b> or <b>Backend authentication parameter</b>, you can define a value for <b>System Parameter Name</b>. However, the customized value must be set in the return result of custom authentication.</li> </ul>                                                                                                                                                                                                                  |
| Backend Parameter Name     | <p>Enter the name of the backend parameter to be mapped.</p> <p><b>NOTE</b></p> <ul style="list-style-type: none"> <li>• The parameter name is not case-sensitive. It cannot be <b>x-stage</b> or start with <b>x-apig-</b> or <b>x-sdk-</b></li> <li>• If <b>Backend Parameter Location</b> is <b>HEADER</b>, the parameter name is case-insensitive and cannot contain underscores (_).</li> </ul>                                                                                                                                                                                                                                                                                                                                         |
| Backend Parameter Location | <p>Select the location of the system parameter in the backend service request. The value can be <b>PATH</b>, <b>HEADER</b>, or <b>QUERY</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Description                | <p>Enter the description of the system parameter.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

5. (Optional) Add a backend policy. You can add multiple backend policies for an API as required and set different policy conditions to forward API requests to different backend services.
  - a. Click **Add Backend Policy** to add a backend policy for the API.
  - b. Configure information about the backend policy.

**Table 7-11** Parameters for creating a backend policy

| Parameter         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name              | Enter a backend policy name to identify different backend policies.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Effective Mode    | <p>Select the effective mode of the backend policy.</p> <ul style="list-style-type: none"> <li>● <b>Any condition met:</b> If an API request meets any of the conditions in a policy, the request is forwarded to the backend.</li> <li>● <b>All conditions met:</b> API requests are forwarded to the backend only when all policy conditions are met.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Policy Conditions | <p>Add conditions for the backend policy to take effect.</p> <ul style="list-style-type: none"> <li>● <b>Source:</b> source of the judgment condition in the policy. <ul style="list-style-type: none"> <li>- <b>Source IP address:</b> The source IP address of the API request is used as the judgment condition.</li> <li>- <b>Input parameter:</b> The input parameter of the API request is used as the judgment condition.</li> </ul> </li> <li>● <b>Parameter Name:</b> This parameter needs to be specified only if <b>Source</b> is set to <b>Input parameter</b>. Select a defined API input parameter.</li> <li>● <b>Condition Type:</b> This parameter needs to be specified only if <b>Source</b> is set to <b>Input parameter</b>. Select the type of the judgment condition. <ul style="list-style-type: none"> <li>- <b>Equal:</b> The condition is met when the request parameter value is equal to the specified value.</li> <li>- <b>Enumerated:</b> The condition is met when the request parameter value is equal to any of the enumerated values.</li> <li>- <b>Matching:</b> The condition is met when the request parameter value is equal to any value in the regular expression.</li> </ul> </li> <li>● <b>Condition Value:</b> Enter the value of the judgment condition. <ul style="list-style-type: none"> <li>- If <b>Condition Type</b> is set to <b>Equal</b>, enter a value.</li> <li>- If <b>Condition Type</b> is set to <b>Enumerated</b>, enter multiple values and separate them with commas (,).</li> <li>- If <b>Condition Type</b> is set to <b>Matching</b>, enter a regular expression, for example, [0-5].</li> <li>- If <b>Source</b> is set to <b>Source IP address</b>, enter one or more IP addresses and separate them with commas (,).</li> </ul> </li> </ul> |

For example, there are three policy conditions whose **Source** is **Input Parameter**, as listed in the following table. If the request parameter value is **11**, policy A is met. If the request parameter value is **5**, policy B is met. If the request parameter value is **15**, policy C is met.

**Table 7-12** Policy parameters

| Policy   | Condition Type | Condition Value |
|----------|----------------|-----------------|
| Policy A | Equal          | 11              |
| Policy B | Enumerated     | 1, 2, 5, 8      |
| Policy C | Matching       | [13-20]         |

6. Click **Next** and **define the responses**.

## Defining Responses

1. Configure response examples to help API callers understand the responses to an API request.

**Table 7-13** Response configuration

| Parameter                | Description                                                                    |
|--------------------------|--------------------------------------------------------------------------------|
| Example Success Response | Example of a successful response returned when the API is successfully called. |
| Example Failure Response | Example of a failure response returned when the API fails to be called.        |

2. Click **Finish**.

## 7.2.5 Debugging an API

### Overview

After creating an API, you can use the debugging function provided by ROMA Connect to debug the API.

 **NOTE**

- APIs with backend request paths containing variables cannot be debugged.
- During API debugging, the request throttling policy bound to the API becomes invalid.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.



2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **More > Debug**.
3. In the API request parameter configuration area on the left of the page, configure API request information based on the API definition.

**Table 7-14** Request parameter configuration

| Parameter         | Description                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Protocol          | This parameter can be set only if the API protocol is set to <b>HTTP&amp;HTTPS</b> .                                                     |
| Method            | You can select a request method only if the API request method is set to <b>ANY</b> .                                                    |
| Path              | You can add custom suffix (Suffix) to the request path only if the matching mode of the API request path is set to <b>Prefix match</b> . |
| Path Parameters   | This parameter can be only if the API request path contains the path parameter.                                                          |
| Query Parameters  | Add and configure query parameters based on the API definition.                                                                          |
| Header Parameters | Add and configure header parameters based on the API definition.                                                                         |
| Body              | Body parameters can be configured only if <b>Method</b> is set to <b>POST, PUT, or PATCH</b> .                                           |

4. After setting the request parameters, click **Send Request**. The request sent by the API and the response received by the API are displayed in the right pane of the page.
  - If the API is successfully called, the status code 200 and a normal response body are returned.
  - If the API fails to be called, the status code 4xx or 5xx and error code description are returned. For details, see [Appendix: API Error Codes](#).

You can send more requests with different parameters and values to verify the API.

## 7.2.6 (Optional) Creating the Environment and Environment Variables

### Overview

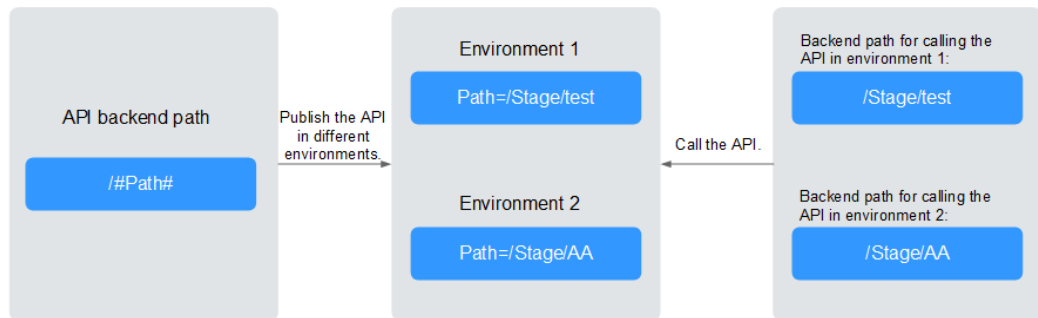
An environment refers to the usage scope of an API. An API can be called only after it is published in an environment. APIs can be published in different customized environments, such as the development environment and test environment. RELEASE is the default and formal publishing environment of the system.

Environment variables are specific to environments. If environment variables are defined in backend information of an API, you need to add the variables to the

environment. You can create variables in different environments to call different backend services using the same API.

For example, when an API is created, variable **Path** is defined in the backend request path. Two variables with the same name are respectively created and assigned values **/Stage/test** and **/Stage/AA** in environments 1 and 2. If the API is published and called in environment 1, the path **/Stage/test** is used. If the API is published and called in environment 2, the path **/Stage/AA** is used.

**Figure 7-3** Application of environment variables



## Creating an Environment

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Environments** tab page, click **Create**.
3. In the dialog box displayed, configure environment information and click **OK**.

**Table 7-15** Parameters for creating an environment

| Parameter   | Description                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------|
| Name        | Enter an environment name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Description | Enter a brief description of the environment.                                                                  |

When a user needs to call an open API, the API in the RELEASE environment is called by default. To access an API in another environment, add the **X-Stage** parameter to the header of the API request. The parameter value is the environment name. For example, to access an API in the Develop environment, add **X-Stage: Develop** to the header of the API request.

## Creating an Environment Variable

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **API Groups** tab page, click **More > Manage Variable** in the **Operation** column of the target API group.

3. Select the environment for which you want to add a variable and click **Create Variable**.
4. In the dialog box displayed, configure environment variable information and click **OK**.

**Table 7-16** Parameters for creating an environment variable

| Parameter | Description                                                                                             |
|-----------|---------------------------------------------------------------------------------------------------------|
| Name      | Enter the variable name, which must be the same as that defined in the API backend service information. |
| Value     | Enter the value of the environment variable.                                                            |

## 7.2.7 Publishing an API

### Overview

After creating an API, you need to publish it to an environment so that it can be called by other users.

 **NOTE**

If a published API is modified, you must publish it again for the modifications to take effect in the environment.

### Prerequisites

The system provides the **RELEASE** environment by default. If you want to publish your API in another environment, [create an environment](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **Publish**.
3. On the **Publish API** page, configure publishing information and click **Publish**.

**Table 7-17** Parameters for publishing an API

| Parameter   | Description                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Environment | Select the environment in which the API is to be published. If your required environment is not available, click <b>Create Environment</b> on the right to create one. |
| Description | Enter the description of the API publication.                                                                                                                          |

 **NOTE**

If the API has already been published in an environment, publishing it again will overwrite the existing one.

On the **APIs** tab page, click the API name to access the API details page. On the **Publication History** tab page, you can view the publishing history of the API and the API configuration in each version.

## 7.2.8 (Optional) Authorizing Apps to Call an API

### Overview

Only APIs that use the App authentication mode need to be authorized to specified integration applications. When a user calls an API, the key and secret of the integration application are used for authenticating API requests.

If **Simple Authentication** is enabled for an API, the AppCode configured in the integration application can be used for simple security authentication, and the key and secret do not need to be used for signature authentication.

 **NOTE**

The integration application to which the API belongs can be directly used to call the API without authorization.

### Prerequisites

- The security authentication mode of the API is set to **App**.
- The API has been published in an environment. Otherwise, [publish the API](#) first.

### Granting Permissions for Integration Applications

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **Authorize App**.
3. On the **Authorize App** page, click **Select App**.
4. In the **Select App** dialog box, configure authorization information and click **OK**.

After the authorization is complete, click  on the left of the API to view the list of authorized integration applications.

**Table 7-18** Authorization configuration

| Parameter               | Description                                                                               |
|-------------------------|-------------------------------------------------------------------------------------------|
| Environment             | Select the environment in which the API to be authorized has been published.              |
| Integration Application | Select the integration application to which the API calling permission is to be assigned. |

| Parameter         | Description                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Access Parameters | Set access parameters for the authorized integration application. The access parameters will be added to the backend signature authentication information and sent to a backend service. The backend service then returns different response parameters based on the carried access parameters. |
| Green Channel     | Determine whether to enable the green channel. After the green channel is enabled, you also need to set <b>Whitelist</b> and <b>Blacklist</b> .                                                                                                                                                 |
| Whitelist         | This parameter is mandatory only if <b>Green Channel</b> is enabled.<br>Enter the IP addresses or IP address segments to be added to the whitelist. The IP addresses in the whitelist can call APIs without authentication.                                                                     |
| Blacklisted       | This parameter is available only if <b>Green Channel</b> is enabled.<br>Enter the IP addresses or IP address segments to be added to the blacklist. The IP addresses in the blacklist are not allowed to call APIs.                                                                             |

## Adding AppCode

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Calling**. On the **Clients** tab page, click the name of the client (that is, the integration application) authorized by the API.
3. On the **AppCode** tab page of the client details page, click **Add AppCode**.
4. In the dialog box displayed, configure AppCode information and click **OK**.

**Table 7-19** AppCode configuration

| Parameter    | Description                                                                                                                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AppCode Type | Select the method for generating AppCode. <ul style="list-style-type: none"> <li>• <b>Automatically generated</b>: AppCode is randomly generated by the system.</li> <li>• <b>Custom</b>: Enter a user-defined AppCode.</li> </ul> |
| AppCode      | Enter the value of AppCode.                                                                                                                                                                                                        |

## 7.3 Creating and Exposing Data APIs

## 7.3.1 Connecting to Data Sources

Before creating a data API, you need to connect to a data source to ensure that data can be read from the source. The data source connection configuration varies depending on data source types. After a data source is connected, click its name to view details, which include relevant task information.

- [Connecting to a DWS Data Source](#)
- [Connecting to a GaussDB 100 Data Source](#)
- [Connecting to a GaussDB 200 Data Source](#)
- [Connecting to a HIVE Data Source](#)
- [Connecting to a MongoDB Data Source](#)
- [Connecting to a MySQL Data Source](#)
- [Connecting to an Oracle Data Source](#)
- [Connecting to a PostgreSQL Data Source](#)
- [Connecting to a Redis Data Source](#)
- [Connecting to a SQL Server Data Source](#)

## 7.3.2 Creating a Data API

### Overview

ROMA Connect allows you to define a data source as a backend service so that data can be read from or written to the data source and exposed to external systems through APIs.

### Prerequisites

- Data sources have been connected to ROMA Connect. For details, see [Connecting to Data Sources](#).
- If a backend service needs to use a signature key for authentication, [create a signature key](#) first.
- If you need to use the custom mode for API security authentication, [create a custom authorizer](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. Create a backend.
  - a. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Create**.
  - b. On the **Create Backend** page, set backend parameters and click **Create**.

**Table 7-20** Backend request information

| Parameter                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                            | Enter a backend name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Integration Application         | Select the integration application to which the backend belongs. If no integration application is available, click <b>Create Integration Application</b> on the right to create one.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Backend Request Method          | Select the request method of the backend. The value can be <b>GET</b> , <b>POST</b> , <b>PUT</b> , or <b>DELETE</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Backend Request Path            | Enter the request path of the backend, in the <i>/getUserInfo/userId</i> format.<br>The value of <b>Request Path</b> is case sensitive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Backend Security Authentication | Select the security authentication mode of the backend. <ul style="list-style-type: none"> <li>• <b>Signature key</b>: A signature key is used to authenticate the request sender. If a signature key is used for authentication, the same signature key must be bound to the API that calls the backend service.</li> <li>• <b>None</b>: No authentication is required for calling requests.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Description                     | Enter a brief description of the backend.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Advanced Settings               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Version                         | Select the backend version, for example, <b>1.0</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Input Parameters                | Define request parameters of the backend service based on the site requirements.<br>In the <b>Input Parameters</b> area, click <b>Add Input Parameter</b> and add request parameters of the custom backend. <ul style="list-style-type: none"> <li>• <b>Name</b>: name of a request parameter, which is user-defined.</li> <li>• <b>Parameter Location</b>: location of the request parameter in the backend request. The value can be <b>Headers</b> or <b>Parameters</b>.</li> <li>• <b>Default Value</b>: It is used as the default test value of parameters only in the subsequent custom backend test procedure. This parameter does not take effect during custom backend deployment.</li> <li>• <b>Mandatory</b>: specifies whether a request parameter is mandatory in a backend request.</li> <li>• <b>Description</b>: Enter the description of the parameter.</li> </ul> |

| Parameter     | Description                                                                                                    |
|---------------|----------------------------------------------------------------------------------------------------------------|
| Returned Type | Select the response data format of the backend. The value can be <b>JSON</b> , <b>XML</b> , or <b>STREAM</b> . |
| Formatting    | This parameter specifies whether to format the response message body based on the selected return type.        |

After the backend is created, the online IDE of the backend is automatically displayed. The backend type is data backend by default.

3. Configure a data backend.
  - a. Click **Add Data Source** on the left of the online IDE.
  - b. On the **Add Data Source** page displayed, configure data source information and click **Add**.

**Table 7-21** Data source configuration

| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Select Data Source    | Select a data source that you configured in <a href="#">Connecting to Data Sources</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Select Statement Type | Select the type of a statement to be executed. The value can be <b>SQL</b> or <b>SP</b> .<br>If the Redis or MongoDB data source is used, select <b>SQL</b> . The actual statement is NoSQL.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Advanced Settings     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Returned Object       | Enter the name of the returned object. The execution result of the statement is encapsulated in the object and returned.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Paging                | This parameter indicates whether the statement execution results are returned on multiple pages. If multiple data sources are added to the same data backend, you cannot set <b>Paging</b> .<br>If paging is enabled, you can add the query parameters <b>pageNum</b> and <b>pageSize</b> to the backend request to page the query results and specify the page number of the data to be returned. <ul style="list-style-type: none"> <li>• <b>pageNum</b>: page number of the data to be returned. The value starts from 1.</li> <li>• <b>pageSize</b>: number of data records on each page.</li> </ul> The structure of the response result varies depending on whether paging is enabled or disabled. For details, see <a href="#">Example of Paging Results</a> . |
| Precompiling          | This parameter specifies whether to precompile execution statements to prevent SQL injection risks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |



### Example of Paging Results

If **Returned Object** is set to **mydata**, five data records are returned after the statement is executed to query data from the data source.

- If **Paging** is not enabled, all five data records are returned as the response result to the user. The following is an example of the response result.

```
{
  "mydata": [
    {
      "id": 1,
      "name": "aaa"
    },
    {
      "id": 2,
      "name": "bbb"
    },
    {
      "id": 3,
      "name": "ccc"
    },
    {
      "id": 4,
      "name": "ddd"
    },
    {
      "id": 5,
      "name": "eee"
    }
  ]
}
```

- When **Paging** is enabled, if **pageNum** is set to **1** and **pageSize** is set to **2**, the five data records queried by the statement will be displayed on multiple pages based on **pageSize**, with two data records on each page. Only the two data records on the first page will be returned to the user as the response result based on **pageNum**. In the response result, **total** indicates the total number of data records queried by running the statement, that is, **5**. Response example:

```
{
  "mydata": {
    "total": 5,
    "data": [
      {
        "id": 1,
        "name": "aaa"
      },
      {
        "id": 2,
        "name": "bbb"
      }
    ]
  },
  "pageSize": 2,
  "pageNum": 1
}
```

- c. After adding a data source, select the data source on the left of the online IDE, and then add an execution statement for the data source in the editing box on the right.

For details, see [Developing Data API Statements](#).

For the Redis or MongoDB data source, the data processing command of Redis or MongoDB is used.

 **NOTE**

If an SQL statement references backend request parameters of multiple data types at the same time, ROMA Connect converts the input parameters to the String type by default. Therefore, when you execute the SQL statement, invoke the corresponding function to convert the data type of a non-String parameter.

- d. Click **Save** in the upper right corner of the page to save the backend configuration.
4. Test the backend functions.  
In the upper right corner of the page, click **Test**. In the **Test Parameters** area, add request parameters based on the definition of the backend service and click **Test** to send the request.
    - In the **Execution Result** area, view the response of the backend.
    - In the **Execution History** area, view the historical test records of the backend. Click a test record to import historical test parameters to the test parameter list on the left and perform the test again.
  5. Deploy the backend.  
After the backend test is complete, click **Deploy** in the upper right corner of the page. In the dialog box displayed, click **Yes** to deploy the backend.
  6. Publish a data API.
    - a. Click **Publish** in the upper right corner of the page.
    - b. On the page displayed, configure API information and click **Publish** to create a frontend data API for the backend and publish the API in the environment.

**Table 7-22** Parameters for publishing an API

| Parameter   | Description                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Group   | Select the API group to which the corresponding API belongs. If no API group is available, click <b>Create API Group</b> on the right to create one.                   |
| Environment | Select the environment in which the API is to be published. If your required environment is not available, click <b>Create Environment</b> on the right to create one. |

| Parameter                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backend Security Authentication | <p>Select the authentication mode of the API. The App authentication mode is recommended.</p> <ul style="list-style-type: none"> <li>● <b>App</b>: ROMA Connect authenticates API requests. When a user calls the API, the key and secret of the integration applications are used for authenticating API requests.</li> <li>● <b>IAM</b>: IAM authenticates API requests. When a user calls an API, the token or AK/SK is used for authenticating API requests.</li> <li>● <b>Custom</b>: The custom function API is used for authenticating API requests.</li> <li>● <b>None</b>: Authentication is not required for API requests.</li> </ul> |
| Custom Authorizer               | <p>This parameter is mandatory only if <b>Frontend Security Authentication</b> is set to <b>Custom</b>.</p> <p>Select a frontend custom authorizer you have created.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Frontend Request Protocol       | <p>Select the request protocol used by the frontend API. The value can be <b>HTTP</b>, <b>HTTPS</b>, or <b>HTTP&amp;HTTPS</b>. <b>HTTPS</b> is recommended for transmitting important or sensitive data.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Timeout (ms)                    | <p>Enter the timeout interval of a backend service request. The default value is <b>60000</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Advanced Settings               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Frontend Request Method         | <p>Select the request method of the frontend API. <b>ANY</b> indicates that the API can be accessed using any request method.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Frontend Request Path           | <p>Enter the request path of the frontend API, in the <i>/getUserInfo/userId</i> format.</p> <p>The value of <b>Request Path</b> is case sensitive.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| CORS                            | <p>This parameter specifies whether CORS is supported for the API.</p> <p>For security purposes, the browser restricts the cross-domain requests from being initiated from a page script. In this case, the page can access only the resources from the same source. CORS allows the browser to send XMLHttpRequest requests to the server in a different domain. For details about CORS, see <a href="#">Configuring CORS for APIs</a>.</p>                                                                                                                                                                                                    |

### 7.3.3 (Optional) Authorizing Apps to Call an API

#### Overview

If a data API uses the App authentication mode, you need to authorize specified integration applications to call this API. When a user calls the API, the key and secret of the integration applications are used for authenticating API requests.

#### Prerequisites

- The security authentication mode of the API is set to **App**.
- The API has been published in an environment. Otherwise, [publish the API](#) first.

#### Granting Permissions for Integration Applications

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **Authorize App**.
3. On the **Authorize App** page, click **Select App**.
4. In the **Select App** dialog box, configure authorization information and click **OK**.

After the authorization is complete, click  on the left of the API to view the list of authorized integration applications.

**Table 7-23** Authorization configuration

| Parameter               | Description                                                                                                                                                                                                                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Environment             | Select the environment in which the API to be authorized has been published.                                                                                                                                                                                                                    |
| Integration Application | Select the integration application to which the API calling permission is to be assigned.                                                                                                                                                                                                       |
| Access Parameters       | Set access parameters for the authorized integration application. The access parameters will be added to the backend signature authentication information and sent to a backend service. The backend service then returns different response parameters based on the carried access parameters. |
| Green Channel           | Determine whether to enable the green channel. After the green channel is enabled, you also need to set <b>Whitelist</b> and <b>Blacklist</b> .                                                                                                                                                 |
| Whitelist               | This parameter is mandatory only if <b>Green Channel</b> is enabled.<br><br>Enter the IP addresses or IP address segments to be added to the whitelist. The IP addresses in the whitelist can call APIs without authentication.                                                                 |

| Parameter   | Description                                                                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Blacklisted | This parameter is available only if <b>Green Channel</b> is enabled.<br>Enter the IP addresses or IP address segments to be added to the blacklist. The IP addresses in the blacklist are not allowed to call APIs. |

## 7.4 Creating and Exposing Function APIs

### 7.4.1 Creating a Function API

#### Overview

ROMA Connect allows you to define custom functions as backend services and open up function capabilities as APIs.

#### Prerequisites

- If a backend service needs to use a signature key for authentication, [create a signature key](#) first.
- If you need to use the custom mode for API security authentication, [create a custom authorizer](#) first.

#### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. Create a backend.
  - a. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Create**.
  - b. On the **Create Backend** page, set backend parameters and click **Create**.

**Table 7-24** Backend request information

| Parameter               | Description                                                                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a backend name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                            |
| Integration Application | Select the integration application to which the backend belongs. If no integration application is available, click <b>Create Integration Application</b> on the right to create one. |
| Backend Request Method  | Select the request method of the backend. The value can be <b>GET</b> , <b>POST</b> , <b>PUT</b> , or <b>DELETE</b> .                                                                |

| Parameter                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backend Request Path            | Enter the request path of the backend, in the <i>/getUserInfo/userId</i> format.<br>The value of <b>Request Path</b> is case sensitive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Backend Security Authentication | Select the security authentication mode of the backend. <ul style="list-style-type: none"> <li>● <b>Signature key:</b> A signature key is used to authenticate the request sender. If a signature key is used for authentication, the same signature key must be bound to the API that calls the backend service.</li> <li>● <b>None:</b> No authentication is required for calling requests.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Description                     | Enter a brief description of the backend.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Advanced Settings               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Version                         | Select the backend version, for example, <b>1.0</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Input Parameters                | Define request parameters of the backend service based on the site requirements.<br>In the <b>Input Parameters</b> area, click <b>Add Input Parameter</b> and add request parameters of the custom backend. <ul style="list-style-type: none"> <li>● <b>Name:</b> name of a request parameter, which is user-defined.</li> <li>● <b>Parameter Location:</b> location of the request parameter in the backend request. The value can be <b>Headers</b> or <b>Parameters</b>.</li> <li>● <b>Default Value:</b> It is used as the default test value of parameters only in the subsequent custom backend test procedure. This parameter does not take effect during custom backend deployment.</li> <li>● <b>Mandatory:</b> specifies whether a request parameter is mandatory in a backend request.</li> <li>● <b>Description:</b> Enter the description of the parameter.</li> </ul> |
| Returned Type                   | Select the response data format of the backend. The value can be <b>JSON</b> , <b>XML</b> , or <b>STREAM</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Formatting                      | This parameter specifies whether to format the response message body based on the selected return type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

After the backend is created, the online IDE of the backend is automatically displayed. The backend type is data backend by default.

3. Configure a function backend.
  - a. In the upper left corner of the online IDE, choose **File > New Function Backend > Blank Template**. In the dialog box displayed, click **Yes** to switch the backend type to function backend.

- b. Edit the function script in the editing box on the right. You can also use the script example provided by the system to edit the function script.  
For details about how to compile function API scripts, see [Developing Function APIs](#).
  - c. Click **Save** in the upper right corner of the page to save the function backend configuration.
4. Test the backend functions.
- In the upper right corner of the page, click **Test**. In the **Test Parameters** area, add request parameters based on the definition of the backend service and click **Test** to send the request.
- In the **Execution Result** area, view the response of the backend.
  - In the **Execution History** area, view the historical test records of the backend. Click a test record to import historical test parameters to the test parameter list on the left and perform the test again.
5. Deploy the backend.
- After the backend test is complete, click **Deploy** in the upper right corner of the page. In the dialog box displayed, click **Yes** to deploy the backend.
6. Publish a function API.
- a. Click **Publish** in the upper right corner of the page.
  - b. On the page displayed, configure API information and click **Publish** to create a frontend function API for the backend and publish the API in the environment.

**Table 7-25** Parameters for publishing an API

| Parameter                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Group                       | Select the API group to which the corresponding API belongs. If no API group is available, click <b>Create API Group</b> on the right to create one.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Environment                     | Select the environment in which the API is to be published. If your required environment is not available, click <b>Create Environment</b> on the right to create one.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Backend Security Authentication | Select the authentication mode of the API. The App authentication mode is recommended. <ul style="list-style-type: none"> <li>● <b>App</b>: ROMA Connect authenticates API requests. When a user calls the API, the key and secret of the integration applications are used for authenticating API requests.</li> <li>● <b>IAM</b>: IAM authenticates API requests. When a user calls an API, the token or AK/SK is used for authenticating API requests.</li> <li>● <b>Custom</b>: The custom function API is used for authenticating API requests.</li> <li>● <b>None</b>: Authentication is not required for API requests.</li> </ul> |

| Parameter                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Custom Authorizer         | This parameter is mandatory only if <b>Frontend Security Authentication</b> is set to <b>Custom</b> .<br>Select a frontend custom authorizer you have created.                                                                                                                                                                                                                                                                         |
| Frontend Request Protocol | Select the request protocol used by the frontend API. The value can be <b>HTTP</b> , <b>HTTPS</b> , or <b>HTTP&amp;HTTPS</b> . <b>HTTPS</b> is recommended for transmitting important or sensitive data.                                                                                                                                                                                                                               |
| Timeout (ms)              | Enter the timeout interval of a backend service request. The default value is <b>60000</b> .                                                                                                                                                                                                                                                                                                                                           |
| Advanced Settings         |                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Frontend Request Method   | Select the request method of the frontend API. <b>ANY</b> indicates that the API can be accessed using any request method.                                                                                                                                                                                                                                                                                                             |
| Frontend Request Path     | Enter the request path of the frontend API, in the <i>/getUserInfo/userId</i> format.<br>The value of <b>Request Path</b> is case sensitive.                                                                                                                                                                                                                                                                                           |
| CORS                      | This parameter specifies whether CORS is supported for the API.<br><br>For security purposes, the browser restricts the cross-domain requests from being initiated from a page script. In this case, the page can access only the resources from the same source. CORS allows the browser to send XMLHttpRequest requests to the server in a different domain. For details about CORS, see <a href="#">Configuring CORS for APIs</a> . |

## 7.4.2 (Optional) Authorizing Apps to Call an API

### Overview

If the function API uses the App authentication mode, you need to authorize specified integration applications to call this API. When a user calls the API, the key and secret of the integration applications are used for authenticating API requests.


### Prerequisites

- The security authentication mode of the API is set to **App**.
- The API has been published in an environment. Otherwise, [publish the API](#) first.



## Granting Permissions for Integration Applications

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **Authorize App**.
3. On the **Authorize App** page, click **Select App**.
4. In the **Select App** dialog box, configure authorization information and click **OK**.

After the authorization is complete, click  on the left of the API to view the list of authorized integration applications.

**Table 7-26** Authorization configuration

| Parameter               | Description                                                                                                                                                                                                                                                                                     |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Environment             | Select the environment in which the API to be authorized has been published.                                                                                                                                                                                                                    |
| Integration Application | Select the integration application to which the API calling permission is to be assigned.                                                                                                                                                                                                       |
| Access Parameters       | Set access parameters for the authorized integration application. The access parameters will be added to the backend signature authentication information and sent to a backend service. The backend service then returns different response parameters based on the carried access parameters. |
| Green Channel           | Determine whether to enable the green channel. After the green channel is enabled, you also need to set <b>Whitelist</b> and <b>Blacklist</b> .                                                                                                                                                 |
| Whitelist               | This parameter is mandatory only if <b>Green Channel</b> is enabled.<br>Enter the IP addresses or IP address segments to be added to the whitelist. The IP addresses in the whitelist can call APIs without authentication.                                                                     |
| Blacklisted             | This parameter is available only if <b>Green Channel</b> is enabled.<br>Enter the IP addresses or IP address segments to be added to the blacklist. The IP addresses in the blacklist are not allowed to call APIs.                                                                             |

## 7.5 Calling an API

## 7.5.1 Calling an Open API

### Overview

After an API is published in an environment, it can be called by other users. API calling operations vary depending on the authentication mode used by the API.

### Obtaining API Request Information

Before calling an API, you need to obtain the API request information from the API provider, including the access domain name, request protocol, request method, request path, and request parameters.

Obtain the request authentication information based on the authentication mode used by an API.

- App:
  - Signature authentication: Obtain the key and secret of the integration application (or client AppKey and AppSecret) authorized by the API from the API provider as well as the SDK for calling the API.
  - Simple authentication: Obtain the AppCode of the client authorized by the API from the API provider.
  - Other authentication modes: Obtain the key and secret of the integration application (or client AppKey and AppSecret) authorized by the API from the API provider.
- The account credential (password or AK/SK) of the cloud service platform is used for authentication. If the AK/SK is used for authentication, you also need to obtain the SDK for calling the API from the API provider.
- Custom: Obtain the custom authentication information to be carried in the request parameters from the API provider.
- None: No authentication information is required.

 **NOTE**

If you use the default subdomain name allocated by the system to access an API, you can access the API for a maximum of 1000 times every day.

### Calling an API

1. Set the parameters related to the request address.

| Scenario                                                | API Request Parameter Configuration                                                                                                                                          |
|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Using a domain name to call an API                      | ROMA Connect allows APIs to be called using the subdomain name assigned to the API group or the domain name bound to the API group. No additional configuration is required. |
| Using an IP address to call an API in the DEFAULT group | ROMA Connect allows you to use IP addresses to call APIs in the DEFAULT group. No additional configuration is required.                                                      |

| Scenario                                                    | API Request Parameter Configuration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Using an IP address to call an API in groups except DEFAULT | <ul style="list-style-type: none"> <li>• The <b>app_route</b> parameter has been set to <b>on</b> on the <a href="#">Configuration Parameters</a> tab page of a ROMA Connect instance, indicating that an API can be called by using an IP address.</li> <li>• ROMA Connect does not allow APIs in non-DEFAULT groups to be directly called using IP addresses. The header parameter <b>X-HW-ID</b> must be added to the request message, and the value must be the key or client AppKey of the integration application authorized by the API.</li> </ul> |

2. Set the authentication parameters.

| Authentication Mode                            | API Request Parameter Configuration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App authentication (signature authentication)  | Use the obtained SDK to sign the API request. For details, see <a href="#">Using App Authentication to Call APIs</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| App authentication (simple authentication)     | Add the header parameter <b>X-Apig-AppCode</b> to the API request. The parameter value is the AppCode obtained in <a href="#">Obtaining API Request Information</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| App authentication (app_secret authentication) | <ul style="list-style-type: none"> <li>• On the <a href="#">Configuration Parameters</a> tab page of a ROMA Connect instance, the <b>app_secret</b> parameter has been set to <b>on</b> (indicating that app_secret authentication is enabled) and <b>app_api_key</b> has been set to <b>off</b> (indicating that app_api_key authentication is disabled).</li> <li>• Add the header parameter <b>X-HW-ID</b> to the API request. The parameter value is the key of the integration application authorized for the API or the client's AppKey.</li> <li>• Add the header parameter <b>X-HW-AppKey</b> to the API request. The parameter value is the secret or AppSecret obtained in <a href="#">Obtaining API Request Information</a>.</li> </ul> |
| App authentication (app_basic authentication)  | <ul style="list-style-type: none"> <li>• The <b>app_basic</b> parameter has been set to <b>on</b> on the <a href="#">Configuration Parameters</a> tab page of a ROMA Connect instance, indicating that app_basic authentication is enabled.</li> <li>• Add the header parameter <b>Authorization</b> to the API request. The value is "<b>Basic "+base64(appkey +":"+appsecret)</b>", in which <b>appkey</b> and <b>appsecret</b> are the key and secret (or AppKey and AppSecret) obtained in <a href="#">Obtaining API Request Information</a>.</li> </ul>                                                                                                                                                                                       |

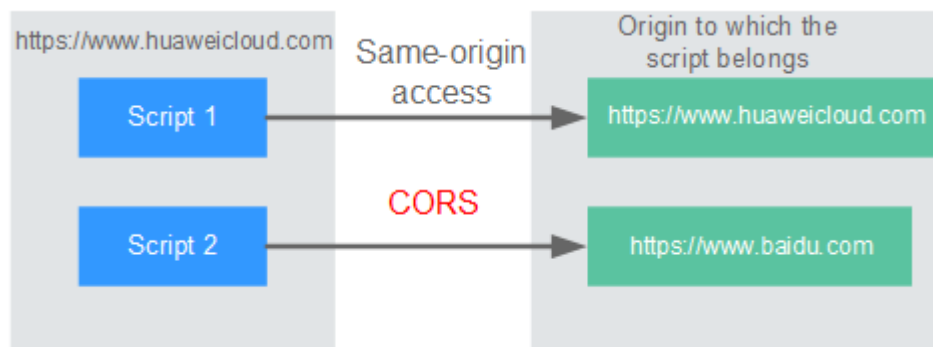
| Authentication Mode                            | API Request Parameter Configuration                                                                                                                   |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| App authentication (two-factor authentication) | An API request carries authentication information of both app authentication and custom authentication.                                               |
| authentication (token authentication)          | Obtain the token of the cloud service platform and carry the token in the API request for authentication.                                             |
| authentication (AK/SK authentication)          | When calling an API, use the obtained SDK to sign the API request. For details, see <a href="#">AK/SK Authentication</a> .                            |
| authentication (two-factor authentication)     | An API request carries authentication information of both IAM authentication and custom authentication.                                               |
| Custom                                         | Based on the definition of custom authentication, the related authentication information is carried in the API request parameters for authentication. |
| None                                           | No authentication is required, and the API can be directly called.                                                                                    |

## 7.5.2 Configuring CORS for APIs

### Overview

For security purposes, the browser restricts the cross-domain requests from being initiated from a page script. In this case, the page can access only the resources from the current domain. CORS allows the browser to send XMLHttpRequest requests to the server in a different domain.

**Figure 7-4** Process flow of the CORS mechanism



CORS requests are classified into two types: simple requests and not-so-simple requests.

- A request that meets both of the following conditions is a simple request:
  - The request method is HEAD, GET, or POST.
  - The HTTP header can contain only the following fields: Accept, Accept-Language, Content-Language, Last-Event-ID and Content-Type (only three values are allowed: application/x-www-form-urlencoded, multipart/form-data, and text/plain).

In the header of a simple request, browsers automatically add the **Origin** field to specify the origin (including the protocol, domain, and port) of the request. After receiving such a request, the target server determines based on the origin whether the request is safe and can be accepted. If the server sends a response containing the **Access-Control-Allow-Origin** field, the server accepts the request.

- Any request that does not meet the preceding two conditions is a not-so-simple request.

Before sending a not-so-simple request, browsers send an HTTP preflight request to the target server to determine whether the origin the web page is loaded from is in the allowed origin list, and which HTTP request methods and header fields can be used. If the preflight is successful, browsers send simple requests to the server.

By default, ROMA Connect does not support CORS. To enable ROMA Connect to support CORS, enable CORS when [creating an API](#). If the CORS request is a not-so-simple request, you also need to create an API that uses the OPTIONS method as the preflight request.

## Simple Requests

- **Scenario 1:** If CORS is enabled and the response from the backend does not contain a CORS header, ROMA Connect can handle requests from any domain, and returns the Access-Control-Allow-Origin CORS header. The following messages are used as examples:

- a. **Request sent by a browser and containing the Origin header field:**

```
GET /simple HTTP/1.1
Host: www.test.com
Origin: http://www.cors.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/json
Date: Tue, 15 Jan 2019 01:25:52 GMT
```

**Origin:** This field specifies the origin (<http://www.cors.com> in this example) of the request. It is mandatory. ROMA Connect and the backend service determine based on the origin whether the request is safe and can be accepted.

- b. **Response sent by the backend:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: roma

{"status":"200"}
```

- c. **Response sent by ROMA Connect:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
```

```
Server: roma
X-Request-Id: 454d689fa69847610b3ca486458fb08b
Access-Control-Allow-Origin: *

{"status":"200"}
```

**Access-Control-Allow-Origin:** This field is mandatory. The asterisk (\*) indicates that ROMA Connect accepts requests from any domain.

- **Scenario 2:** If CORS is enabled and the response from the backend contains a CORS header, the header will overwrite that added by ROMA Connect. The following messages are used as examples:

a. **Request sent by a browser and containing the Origin header field:**

```
GET /simple HTTP/1.1
Host: www.test.com
Origin: http://www.cors.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Accept: application/json
Date: Tue, 15 Jan 2019 01:25:52 GMT
```

**Origin:** This field specifies the origin (<http://www.cors.com> in this example) of the request. It is mandatory. ROMA Connect and the backend service determine based on the origin whether the request is safe and can be accepted.

b. **Response sent by the backend:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: roma
Access-Control-Allow-Origin: http://www.cors.com

{"status":"200"}
```

**Access-Control-Allow-Origin:** Indicates that the backend service accepts requests sent from <http://www.cors.com>.

c. **Response sent by ROMA Connect:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: roma
X-Request-Id: 454d689fa69847610b3ca486458fb08b
Access-Control-Allow-Origin: http://www.cors.com

{"status":"200"}
```

The CORS header in the backend response overwrites that in ROMA Connect's response.

## Not-So-Simple Requests

For a non-simple request, you also need to create an API that uses the OPTIONS method. The request parameters of an API accessed using the OPTIONS method must be set as follows:

- **API Group:** Same as the group to which the API with CORS enabled belongs.
- **Security Authentication:** No authentication is required for requests received by the new API no matter which security authentication mode has been selected.
- **Protocol:** Same as the protocol used by the API with CORS enabled.

- **Request Path:** Same as or matching the request path used by the API with CORS enabled.
- **Method:** Set to **OPTIONS**.
- **CORS:** Enabled.

The following are example requests and responses sent to or from a mock backend.

1. **Request sent from a browser to an API that is accessed using the OPTIONS method:**

```
OPTIONS /HTTP/1.1
User-Agent: curl/7.29.0
Host: localhost
Accept: */*
Origin: http://www.cors.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: X-Sdk-Date
```

- **Origin:** This field is mandatory and used to specify the origin from which the request has been sent.
- **Access-Control-Request-Method:** This field is mandatory and used to specify the HTTP methods to be used by the subsequent simple requests.
- **Access-Control-Request-Headers:** This field is mandatory and used to specify the additional header fields in the subsequent simple requests.

2. **Response sent by the backend:**

None

3. **Response sent by ROMA Connect:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 02:38:48 GMT
Content-Type: application/json
Content-Length: 1036
Server: roma
X-Request-Id: c9b8926888c356d6a9581c5c10bb4d11
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-Stage,X-Sdk-Date,X-Sdk-Nonce,X-Proxy-Signed-Headers,X-Sdk-Content-Sha256,X-Forwarded-For,Authorization,Content-Type,Accept,Accept-Ranges,Cache-Control,Range
Access-Control-Expose-Headers: X-Request-Id,X-Apig-Latency,X-Apig-Upstream-Latency,X-Apig-RateLimit-Api,X-Apig-RateLimit-User,X-Apig-RateLimit-App,X-Apig-RateLimit-Ip,X-Apig-RateLimit-Api-Allenv
Access-Control-Allow-Methods: GET,POST,PUT,DELETE,HEAD,OPTIONS,PATCH
Access-Control-Max-Age: 172800
```

- **Access-Control-Allow-Origin:** This field is mandatory. The asterisk (\*) indicates that ROMA Connect accepts requests from any domain.
- **Access-Control-Allow-Headers:** This field is required if it is contained in the request. It specifies the header information field supported by ROMA Connect.
- **Access-Control-Allow-Methods:** This field is mandatory and used to specify the HTTP request method supported by ROMA Connect. .
- **Access-Control-Max-Age:** This field is mandatory and used to specify the period (in seconds) during which the preflight result remains valid. No more preflight requests are needed within the period.

4. **Request sent by a browser and containing the Origin header field:**

```
PUT /simple HTTP/1.1
Host: www.test.com
Origin: http://www.cors.com
Content-Type: application/x-www-form-urlencoded; charset=utf-8
```

```
Accept: application/json
Date: Tue, 15 Jan 2019 01:25:52 GMT
```

5. **Response sent by the backend:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: roma
```

```
{"status":"200"}
```

6. **Response sent by ROMA Connect:**

```
HTTP/1.1 200 OK
Date: Tue, 15 Jan 2019 01:25:52 GMT
Content-Type: application/json
Content-Length: 16
Server: roma
X-Request-Id: 454d689fa69847610b3ca486458fb08b
Access-Control-Allow-Origin: *
```

```
{"status":"200"}
```

## 7.5.3 Appendix: API Error Codes

The following table lists the error codes returned when a user fails to call an API.

**Table 7-27** Error codes

| HTTP Status Code | Error Code | Description                                                          | Solution                                                                                                                                                                                                                                                                                                                                 |
|------------------|------------|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 404              | APIC.0101  | The API does not exist or has not been published in the environment. | <ul style="list-style-type: none"> <li>Check whether the domain name, method, and path are consistent with those of the registered API.</li> <li>Check whether the API is published. If the API is published in a non-production environment, check whether the X-Stage header in the request is the name of the environment.</li> </ul> |
| 500              | APIC.0103  | The backend does not exist.                                          | Contact technical support.                                                                                                                                                                                                                                                                                                               |
| 500              | APIC.0104  | The plug-ins do not exist.                                           | Contact technical support.                                                                                                                                                                                                                                                                                                               |
| 500              | APIC.0105  | The backend configurations do not exist.                             | Contact technical support.                                                                                                                                                                                                                                                                                                               |
| 400              | APIC.0106  | Orchestration error.                                                 | Check whether the frontend and backend parameters of the API are correct.                                                                                                                                                                                                                                                                |
| 400              | APIC.0201  | Bad request.                                                         | Set valid request parameters.                                                                                                                                                                                                                                                                                                            |



| HTTP Status Code | Error Code | Description                                       | Solution                                                                                                                                                                                                                          |
|------------------|------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 413              | APIC.0201  | Request entity too large.                         | Reduce the size of the request body to less than 12 MB.                                                                                                                                                                           |
| 414              | APIC.0201  | Request URI too large.                            | Reduce the size of the URI to less than 32 KB.                                                                                                                                                                                    |
| 494              | APIC.0201  | Request headers too large.                        | Reduce the size of request headers to ensure that the length of a single request header is less than 32 KB or the total length of all request headers is less than 128 KB.                                                        |
| 502              | APIC.0202  | Backend unavailable.                              | Check whether the backend address configured for the API is accessible.                                                                                                                                                           |
| 504              | APIC.0203  | Backend timeout.                                  | Increase the timeout duration of the backend service or shorten the processing time.                                                                                                                                              |
| 401              | APIC.0301  | Incorrect IAM authentication information.         | Check whether the token is correct.                                                                                                                                                                                               |
| 403              | APIC.0302  | The IAM user is not authorized to access the API. | Check whether the user is restricted by a blacklist or whitelist.                                                                                                                                                                 |
| 401              | APIC.0303  | Incorrect App authentication information.         | <ul style="list-style-type: none"> <li>• Check whether the request method, path, query parameters, and request body are consistent with those used for signature.</li> <li>• Check whether the client time is correct.</li> </ul> |
| 403              | APIC.0304  | The app is not authorized to access the API.      | Check whether the app has been authorized to access the API.                                                                                                                                                                      |
| 401              | APIC.0305  | Incorrect authentication information.             | Check whether the authentication information is correct.                                                                                                                                                                          |
| 403              | APIC.0306  | API access denied.                                | Check whether you have been authorized to access the API.                                                                                                                                                                         |
| 401              | APIC.0307  | The token must be updated.                        | Update the token.                                                                                                                                                                                                                 |

| HTTP Status Code | Error Code | Description                                         | Solution                                                                                                                  |
|------------------|------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 429              | APIC.0308  | The throttling threshold has been reached.          | Refresh the request throttling policy and try again.                                                                      |
| 403              | APIC.0401  | Unknown client IP address.                          | Contact technical support.                                                                                                |
| 403              | APIC.0402  | The IP address is not authorized to access the API. | Check whether the IP address is restricted by a blacklist or whitelist.                                                   |
| 503              | APIC.0404  | Access to the backend IP address has been denied.   | Use an available IP address to access the backend service.                                                                |
| 403              | APIC.0405  | The app is not accessed from a trusted IP address.  | Check whether the IP address is restricted by the client access control policy.                                           |
| 500              | APIC.0601  | Internal server error.                              | Contact technical support.                                                                                                |
| 400              | APIC.0602  | Bad request.                                        | Check whether the request is valid.                                                                                       |
| 500              | APIC.0605  | Backend domain name resolution failed.              | Check whether the domain name is correct and has been bound to a correct backend address.                                 |
| 500              | APIC.0606  | Failed to load the API configurations.              | Contact technical support.                                                                                                |
| 400              | APIC.0607  | The following protocol is supported: {xxx}          | Use HTTP or HTTPS to access the API.                                                                                      |
| 500              | APIC.0608  | Failed to obtain the admin token.                   | Contact technical support.                                                                                                |
| 500              | APIC.0609  | The VPC backend does not exist.                     | Contact technical support.                                                                                                |
| 502              | APIC.0610  | No backend available.                               | Check whether all backends are available.                                                                                 |
| 500              | APIC.0611  | The backend port does not exist.                    | Contact technical support.                                                                                                |
| 500              | APIC.0612  | An API cannot call itself.                          | Modify the backend configurations, and ensure that the number of layers the API is recursively called does not exceed 10. |
| 500              | APIC.0705  | Backend signature calculation failed.               | Contact technical support.                                                                                                |

## 7.6 API Management

### 7.6.1 Viewing API Calling Information

#### Overview

ROMA Connect provides visualized API analysis and statistics. You can view API calling statistics and call logs on the ROMA Connect console.

#### Viewing API Calling Statistics

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Analysis**. On the **API Calling** tab page, you can view the API calling statistics.
  - You can view the real-time statistics on the total number of APIs, total number of API groups, and total number of request throttling policies.
  - You can view the calling statistics of an API, including the number of requests, number of errors, data traffic, and call delay.

Select an integration application and API to filter your required API. You can also select the time range of the data to be viewed.

#### Viewing API Calling Logs

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Analysis**.
3. Enable the log analysis function.
  - a. On the **Access Logs** tab page, click **Configure Access Log**.
  - b. In the **Configure Access Log** dialog box, configure log access information and click **OK**.

**Table 7-28** Parameters for configuring an access log

| Parameter           | Description                                                                                                                                                                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Collect Access Logs | Determine whether to enable the logging function. API calling logs can be viewed only after you enable <b>Collect Access Logs</b> .                                                                                                                                              |
| Log Group           | Select the log group to which the log stream belongs.<br><br>If no log group is available, click <b>View Log Group</b> to switch to the LTS console and create a log group. For details, see "Log Management > Creating a Log Group" in the <i>Log Tank Service User Guide</i> . |

| Parameter  | Description                                                                                                                                                                                                                                                                     |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Log Stream | Select the log stream to be stored in API calling logs. If no log stream is available, click <b>View Log Stream</b> to switch to the LTS console and create a log stream. For details, see "Log Management > Creating a Log Stream" in the <i>Log Tank Service User Guide</i> . |

4. After the log analysis function is enabled, you can view the calling logs of all open APIs in real time on the console.
  - In the upper right corner of the page, you can select the time segment of the logs to be viewed.
  - Click **Log Field Description** in the upper right corner to view the log field description.
  - To download logs, you can configure log dump on the LTS console. For details, see "Log Transfer" in the *Log Tank Service User Guide*.

## 7.6.2 Taking an API Offline

### Overview

If a published API is not needed to provide services, you can take it offline from the environment in which it has been published.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **More > Take Offline**.
3. In the dialog box displayed, select the environment in which the API is to be taken offline and click **Yes**.

---

#### NOTICE

This operation will cause the API to be inaccessible in the environment. Ensure that you have notified users before this operation.

---

## 7.6.3 Taking a Custom Backend Offline

### Overview

If a deployed custom backend is not needed to provide services, you can take this custom backend offline. At this time, the API published with this custom backend will be also taken offline and deleted.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Take Offline** on the **Operation** column of the target backend.

When the backend status changes to **Developing**, the backend is taken offline.

---

### NOTICE

This operation will cause the API corresponding to the backend to be inaccessible in the environment. Ensure that you have notified users before this operation.

3. In the navigation pane on the left, choose **API Connect > API Management** to check whether the API corresponding to the backend has been deleted.

## 7.6.4 Importing and Exporting APIs

### Overview

ROMA Connect allows you to import and export APIs using files.

- Importing APIs: API files in YAML and JSON formats can be imported, and the file content must comply with the Swagger 2.0 specifications.
- Exporting APIs: API files in YAML and JSON formats can be exported.

### Prerequisites

- You have supplemented the [extended Swagger definition of APIs](#) in the API file to be imported.
- Ensure that the quotas of APIs and API groups meet the requirements before importing APIs.

### Importing APIs

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **More > Import API**.
3. On the **Import API** page, configure API import information.

**Table 7-29** Parameters for importing APIs

| Parameter                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Import To                     | <p>Select the method for importing APIs.</p> <ul style="list-style-type: none"> <li>• <b>New group:</b> Import APIs to a new API group. If you select this option, the system automatically creates an API group and imports the APIs to this group.</li> <li>• <b>Existing group:</b> Select an existing API group and add the imported APIs to this group.</li> </ul>                                                                                                                                                                                                                                                                                             |
| Type                          | <p>This parameter is mandatory only if <b>Import To</b> is set to <b>New group</b>.</p> <p>Select the type of the API group.</p> <ul style="list-style-type: none"> <li>• <b>Integration application:</b> An API group belongs to a specific integration application. Only users who have permissions on the integration application can view and perform operations on the API group.</li> <li>• <b>Global:</b> All users can view and perform operations on the API group.</li> </ul>                                                                                                                                                                             |
| Integration Application       | <p>This parameter is mandatory only if <b>Integration application</b> is selected for <b>Type</b>.</p> <p>Select the integration application to which the API group belongs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Overwrite                     | <p>This parameter is available only if <b>Import To</b> is set to <b>Existing group</b>.</p> <p>This parameter specifies whether to overwrite the existing API when the imported API conflicts with the API in the API group.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Overwrite Extended Definition | <p>This parameter specifies whether to overwrite the existing extended information when the extended information (such as custom authentication, request throttling policy, and access control policy) defined in the imported API conflicts with the existing extended information in ROMA Connect.</p> <ul style="list-style-type: none"> <li>• If you select this option, the system overwrites the existing information with the extended information defined in the imported file.</li> <li>• If you do not select this option, the existing extended information in the system is used, rather than the extended information in the imported file.</li> </ul> |

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Parameter Import | <p>Select and check the imported file.</p> <ul style="list-style-type: none"> <li>Click <b>File</b> and select the local API file in YAML or JSON format.</li> <li>After the file is imported, click <b>Check</b> to check whether the content format of the imported file meets the requirements.</li> <li>Click <b>Format</b> to format the content of the imported file.</li> <li>Click <b>Download</b> to download the imported file to the local host.</li> <li>Enable <b>Mock</b> if you want to use the Mock service as a backend service for importing APIs.</li> </ul> |

- Click **Fast Roll Out** to import APIs.

If you need to configure request information and backend service information of the imported APIs in a centralized manner before the import, click **Configure Global Settings** and configure API information as prompted. Click **Submit** to import the APIs.

If the imported API has not been published in the environment, you need to manually publish the API.

## Exporting APIs

- Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
- In the navigation pane, choose **API Connect > API Management**. On the **APIs** tab page, choose **More > Export API**.
- On the **Export API** page, configure API export information.

**Table 7-30** Parameters for exporting APIs

| Parameter   | Description                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Group   | Select the API group from which APIs are to be exported.                                                                                               |
| Environment | Select the environment in which the APIs to be exported have been published.                                                                           |
| API         | Select the APIs to be exported. If this parameter is not specified, all APIs in the API group in the selected environment will be exported by default. |

| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Definition | <p>Select the scope of the API definition to be exported.</p> <ul style="list-style-type: none"> <li>• <b>Basic:</b> Only the API frontend request information is exported. The API frontend request information includes both standard and extended Swagger fields.</li> <li>• <b>Full:</b> Both API frontend request information and backend service information are exported.</li> <li>• <b>Extended:</b> The API frontend request information, backend service information, as well as the request throttling policy and access control policy associated with the API are all exported.</li> </ul> |
| Format         | Select the format of the exported API file. The value can be <b>YAML</b> or <b>JSON</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Version        | Enter the version of the API file to be exported. If no version is specified, the version will be set to the current time.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

4. Click **Export** to export the API file to the local host. The content of the exported file is displayed in the right pane.

 **NOTE**

- If no independent domain name is bound to the API group to which the exported API belongs, the subdomain name of the API group will be exported.
- If multiple independent domain names are bound to the API group to which the exported API belongs, only one independent domain name will be exported randomly.

## 7.6.5 Importing and Exporting Custom Backends

### Overview

ROMA Connect allows you to import and export custom backends using a file.

- Importing custom backends: API files in YAML and JSON formats can be imported, and the file content must comply with the Swagger 2.0 specifications.
- Exporting custom backends: API files in YAML and JSON formats can be exported.

### Prerequisites

- You have supplemented the [extended Swagger definition of custom backends](#) in the API file to be imported.
- Before importing a custom backend, ensure that the quota of custom backends meets the requirements.

### Importing Custom Backends

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.



2. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Import**.
3. On the **Import Backend** page, configure backend import information.

**Table 7-31** Parameters for importing backends

| Parameter                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Overwrite Basic Definition    | This parameter specifies whether to overwrite the existing backends when the imported backends conflict with the existing ones.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Overwrite Extended Definition | <p>This parameter specifies whether to overwrite the existing extended information when the extended information (such as request throttling policy and access control policy) defined in the imported backend conflicts with the existing extended information in ROMA Connect.</p> <ul style="list-style-type: none"> <li>• If you select this option, the system overwrites the existing information with the extended information defined in the imported file.</li> <li>• If you do not select this option, the existing extended information in the system is used, rather than the extended information in the imported file.</li> </ul> |
| Swagger                       | <p>Click <b>Select</b> and select a local API file in YAML or JSON format.</p> <p>After the file is imported, you can preview and modify the file online.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

4. Click **Import**.  
The import result is displayed in the right pane. The **success** field contains the backends that are successfully imported, and the **failure** field contains the backends that fail to be imported, error codes, and error information.

## Exporting Custom Backends

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Export All**.  
Only backends in the **Developing** state can be exported.
3. On the **Export Backend** page, configure backend export information.

**Table 7-32** Parameters for exporting backends

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| API Definition          | Select the scope of the API definition to be exported. Only <b>Full</b> is supported, indicating that all requests and service information of backends are exported.                                                                                                                                                                                                                                                                                                                                                                     |
| Export Scope            | Select the export scope of the custom backend. <ul style="list-style-type: none"> <li>• <b>All backends:</b> Export all custom backends.</li> <li>• <b>Based on integration applications:</b> Export the custom backends under the specified integration application.</li> </ul> <p><b>NOTE</b><br/>If <b>All backends</b> is selected for <b>Export Scope</b> and the custom backends of different integration applications have the same request path, only one of the custom backends with the same request path can be exported.</p> |
| Integration Application | This parameter is mandatory only if <b>Based on integration applications</b> is selected for <b>Export Scope</b> .<br>Select the integration application to which the custom backends to export belong.                                                                                                                                                                                                                                                                                                                                  |
| Format                  | Select the format of the exported API file. The value can be <b>YAML</b> or <b>JSON</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                |

4. Click **Export** to export the API file to a local directory. The content of the exported file is displayed in the right pane.

## 7.6.6 Appendix: Extended Swagger Definition of APIs

On the basis of the basic Swagger definition, ROMA Connect defines the extended definition of APIs, such as the authentication mode and backend service definition. This section describes the extended definitions of APIs.

### 1: x-apigateway-auth-type

**Meaning:** Swagger-based apiKey authentication format, which defines an authentication mode provided by ROMA Connect.

**Scope of effect:** [Security Scheme Object](#)

#### Example

```
securityDefinitions:
  customize-name-iam:
    type: "apiKey"
    name: "unused"
    in: "header"
    x-apigateway-auth-type: "IAM"
  customize-name-app:
    type: "apiKey"
    name: "Authorization"
    in: "header"
    x-apigateway-auth-type: "AppSigv1"
  customize-name-iam-none:
```

```
type: "apiKey"
name: "unused"
in: "header"
x-apigateway-auth-type: "IAM_NONE"
```

**Table 7-33** Parameter description

| Parameter              | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                               |
|------------------------|-----------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type                   | Yes       | String | Authentication type. Only <b>apiKey</b> is supported.                                                                                                                                                                                                                                                                     |
| name                   | Yes       | String | Name of the parameter for authentication. <ul style="list-style-type: none"> <li>When <b>x-apigateway-auth-type</b> is set to <b>AppSigv1</b>, set <b>name</b> to <b>Authorization</b>.</li> <li>When <b>x-apigateway-auth-type</b> is set to <b>IAM</b> or <b>IAM_NONE</b>, set <b>name</b> to <b>unused</b>.</li> </ul> |
| in                     | Yes       | String | Location of the parameter. Only <b>header</b> is supported.                                                                                                                                                                                                                                                               |
| description            | No        | String | Description of the parameter.                                                                                                                                                                                                                                                                                             |
| x-apigateway-auth-type | Yes       | String | APIC authentication mode. <b>AppSigv1</b> , <b>IAM</b> , and <b>IAM_NONE</b> are supported.                                                                                                                                                                                                                               |

## 2: x-apigateway-request-type

**Meaning:** API request type, which can be **public** or **private**.

**Scope of effect:** [Operation Object](#)

### Example

```
paths:
  '/path':
    get:
      x-apigateway-request-type: 'public'
```

**Table 7-34** Parameter description

| Parameter                 | Mandatory | Type   | Description                                                                                                                                                                                                                                          |
|---------------------------|-----------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x-apigateway-request-type | Yes       | String | API visibility. The options include <b>public</b> and <b>private</b> . <ul style="list-style-type: none"> <li>• <b>public</b>: The API can be made available for sale.</li> <li>• <b>private</b>: The API will not be available for sale.</li> </ul> |

### 3: x-apigateway-match-mode

**Meaning:** Request URL matching mode, which can be **NORMAL** or **SWA**.

**Scope of effect:** [Operation Object](#)

**Example**

```
paths:
  '/path':
    get:
      x-apigateway-match-mode: 'SWA'
```

**Table 7-35** Parameter description

| Parameter               | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------|-----------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x-apigateway-match-mode | Yes       | String | Matching mode of the API request path. The value can be <b>SWA</b> or <b>NORMAL</b> . <ul style="list-style-type: none"> <li>• <b>SWA</b>: prefix match. For example, if the request path is set to <b>/test/AA</b> and the matching mode is set to <b>SWA</b>, the API can be accessed using <b>/test/AA/BB</b> or <b>/test/AA/CC</b> but cannot be accessed using <b>/test/AACC</b>.</li> <li>• <b>NORMAL</b>: exact match.</li> </ul> |

### 4: x-apigateway-cors

**Meaning:** Specifies whether APIs defined by ROMA Connect support CORS.

**Scope of effect:** [Operation Object](#)

**Example**

```
paths:
  '/path':
    get:
      x-apigateway-cors: true
```

**Table 7-36** Parameter description

| Parameter         | Mandatory | Type    | Description                |
|-------------------|-----------|---------|----------------------------|
| x-apigateway-cors | No        | Boolean | Whether CORS is supported. |

For the API request for enabling CORS, the headers listed in the following table will be added to the response.

| Parameter Name               | Parameter Value                                                                                                                                                  | Description                                                                                        |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
| Access-Control-Max-Age       | 172800                                                                                                                                                           | Maximum time the response of a preflight request can be cached.                                    |
| Access-Control-Allow-Origin  | *                                                                                                                                                                | Domain that can be accessed. The asterisk (*) indicates that requests from any domain are allowed. |
| Access-Control-Allow-Headers | X-Sdk-Date, X-Sdk-Nonce, X-Proxy-Signed-Headers, X-Sdk-Content-Sha256, X-Forwarded-For, Authorization, Content-Type, Accept, Accept-Ranges, Cache-Control, Range | Header information fields that can be used in API requests.                                        |
| Access-Control-Allow-Methods | GET, POST, PUT, DELETE, HEAD, OPTIONS, PATCH                                                                                                                     | HTTP methods allowed by the API request.                                                           |

## 5: x-apigateway-any-method

**Meaning:** API request method used by default if no HTTP request method is specified.

**Scope of effect:** [Path Item Object](#)

### Example

```
paths:
  '/path':
    get:
      produces:
        - application/json
```

```

responses:
  "200":
    description: "get response"
x-apigateway-any-method:
  produces:
    - application/json
  responses:
    "200":
      description: "any response"

```

## 6: x-apigateway-backend

**Meaning:** API backend definition.

**Scope of effect:** [Operation Object](#)

### Example

```

paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "backend endpoint type"

```

**Table 7-37** Parameter description

| Parameter         | Mandatory | Type                                                   | Description                                                                          |
|-------------------|-----------|--------------------------------------------------------|--------------------------------------------------------------------------------------|
| type              | Yes       | String                                                 | Backend service type. <b>HTTP</b> , <b>HTTP-VPC</b> , and <b>MOCK</b> are supported. |
| parameters        | No        | <a href="#">x-apigateway-backend.parameters</a>        | Backend parameters.                                                                  |
| httpEndpoints     | No        | <a href="#">x-apigateway-backend.httpEndpoints</a>     | HTTP backend service definition.                                                     |
| httpVpcEndpoints  | No        | <a href="#">x-apigateway-backend.httpVpcEndpoints</a>  | HTTP-VPC backend service definition.                                                 |
| functionEndpoints | No        | <a href="#">x-apigateway-backend.functionEndpoints</a> | Function backend service definition.                                                 |

| Parameter     | Mandatory | Type                                               | Description                      |
|---------------|-----------|----------------------------------------------------|----------------------------------|
| mockEndpoints | No        | <a href="#">x-apigateway-backend.mockEndpoints</a> | Mock backend service definition. |

## 6.1: x-apigateway-backend.parameters

**Meaning:** API backend definition.

**Scope of effect:** [x-apigateway-backend](#)

### Example

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      parameters:
        - name: "X-Auth-Token"
          description: "authorization token"
          type: "string"
          in: "header"
          required: true
        - name: "userId"
          description: "user name"
          type: "string"
          in: "path"
          required: true
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "HTTP"
        parameters:
          - name: "userId"
            value: "userId"
            in: "query"
            origin: "REQUEST"
            description: "user name"
          - name: "X-Invoke-User"
            value: "apigateway"
            in: "header"
            origin: "CONSTANT"
            description: "invoke user"
```

**Table 7-38** Parameter description

| Parameter   | Mandatory | Type   | Description                                                                                                                                                                                                                     |
|-------------|-----------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name        | Yes       | String | Parameter name, which consists of a maximum of 32 bytes, starting with a letter. Only letters, digits, periods (.), hyphens (-), and underscores (_) are allowed.<br><br>The names of header parameters are not case-sensitive. |
| value       | Yes       | String | Parameter value, which is a parameter name if the parameter comes from a request.                                                                                                                                               |
| in          | Yes       | String | Parameter location, which can be <b>header</b> , <b>query</b> , or <b>path</b> .                                                                                                                                                |
| origin      | Yes       | String | Parameter mapping source. <b>REQUEST</b> and <b>CONSTANT</b> are supported.                                                                                                                                                     |
| description | No        | String | Parameter description.                                                                                                                                                                                                          |

## 6.2: x-apigateway-backend.httpEndpoints

**Meaning:** HTTP backend service definition.

**Scope of effect:** [x-apigateway-backend](#)

### Example

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      parameters:
        - name: "X-Auth-Token"
          description: "authorization token"
          type: "string"
          in: "header"
          required: true
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "HTTP"
      httpEndpoints:
        address: "www.example.com"
        scheme: "http"
        method: "GET"
        path: "/users"
        timeout: 30000
```



**Table 7-39** Parameter description

| Parameter | Man<br>dator<br>y | Type   | Description                                                                                                        |
|-----------|-------------------|--------|--------------------------------------------------------------------------------------------------------------------|
| address   | Yes               | Array  | Backend service address. The format is <i>&lt;Domain name or IP address&gt;.[Port number]</i>                      |
| scheme    | Yes               | String | Backend request protocol. <b>HTTP</b> and <b>HTTPS</b> are supported.                                              |
| method    | Yes               | String | Backend request method. <b>GET, POST, PUT, DELETE, HEAD, OPTIONS, PATCH,</b> and <b>ANY</b> are supported.         |
| path      | Yes               | String | Backend request path, which can contain variables.                                                                 |
| timeout   | No                | Number | Backend request timeout in milliseconds. The value ranges from 1 to 60,000, and the default value is <b>5000</b> . |

### 6.3: x-apigateway-backend.httpVpcEndpoints

**Meaning:** HTTP-VPC backend service definition.

**Scope of effect:** [x-apigateway-backend](#)

#### Example

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      parameters:
        - name: "X-Auth-Token"
          description: "authorization token"
          type: "string"
          in: "header"
          required: true
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "HTTP-VPC"
      httpVpcEndpoints:
        name: "vpc-test-1"
        scheme: "http"
        method: "GET"
        path: "/users"
        timeout: 30000
```

**Table 7-40** Parameter description

| Parameter | Mandatory | Type   | Description                                                                                                        |
|-----------|-----------|--------|--------------------------------------------------------------------------------------------------------------------|
| name      | Yes       | Array  | VPC channel name.                                                                                                  |
| scheme    | Yes       | String | Backend request protocol. <b>HTTP</b> and <b>HTTPS</b> are supported.                                              |
| method    | Yes       | String | Backend request method. <b>GET, POST, PUT, DELETE, HEAD, OPTIONS, PATCH,</b> and <b>ANY</b> are supported.         |
| path      | Yes       | String | Backend request path, which can contain variables.                                                                 |
| timeout   | No        | Number | Backend request timeout in milliseconds. The value ranges from 1 to 60,000, and the default value is <b>5000</b> . |

## 6.4: x-apigateway-backend.functionEndpoints

**Meaning:** FUNCTION backend service definition.

**Scope of effect:** [x-apigateway-backend](#)

### Example

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      parameters:
        - name: "X-Auth-Token"
          description: "authorization token"
          type: "string"
          in: "header"
          required: true
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "FUNCTION"
        functionEndpoints:
          version: "v1"
          function-urn: ""
          invocation-type: "synchronous"
          timeout: 30000
```

**Table 7-41** Parameter description

| Parameter       | Mandatory | Type   | Description                                                                                                 |
|-----------------|-----------|--------|-------------------------------------------------------------------------------------------------------------|
| function-urn    | Yes       | String | Function URN.                                                                                               |
| version         | Yes       | String | Function version.                                                                                           |
| invocation-type | Yes       | String | Function invocation type. <b>async</b> or <b>sync</b> are supported.                                        |
| timeout         | No        | Number | Function timeout in milliseconds. The value ranges from 1 to 60,000, and the default value is <b>5000</b> . |

## 6.5: x-apigateway-backend.mockEndpoints

**Meaning:** Mock backend service definition.

**Scope of effect:** [x-apigateway-backend](#)

### Example

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      parameters:
        - name: "X-Auth-Token"
          description: "authorization token"
          type: "string"
          in: "header"
          required: true
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "MOCK"
        mockEndpoints:
          result-content: "mocked"
```

**Table 7-42** Parameter description

| Parameter      | Mandatory | Type   | Description    |
|----------------|-----------|--------|----------------|
| result-content | Yes       | String | Mock response. |

## 7: x-apigateway-backend-policies

**Meaning:** API backend policy.

**Scope of effect: Operation Object**

**Example**

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      responses:
        default:
          description: "default response"
      x-apigateway-request-type: "public"
      x-apigateway-backend:
        type: "backend endpoint type"
      x-apigateway-backend-policies:
        - type: "backend endpoint type"
          name: "backend policy name"
      conditions:
        - type: "equal/enum/pattern",
          value: "string",
          origin: "source/request_parameter",
          parameter_name: "string"
```

**Table 7-43** Parameter description

| Parameter             | Man<br>dator<br>y | Type                                                              | Description                                                                          |
|-----------------------|-------------------|-------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| type                  | Yes               | String                                                            | Backend service type. <b>HTTP</b> , <b>HTTP-VPC</b> , and <b>MOCK</b> are supported. |
| name                  | Yes               | String                                                            | Name                                                                                 |
| parameters            | No                | <b>x-<br/>apigateway-<br/>backend.parameters</b>                  | Backend parameters.                                                                  |
| httpEndpoints         | No                | <b>x-<br/>apigateway-<br/>backend.http<br/>Endpoints</b>          | HTTP service definition.                                                             |
| httpVpcEndpo<br>ints  | No                | <b>x-<br/>apigateway-<br/>backend.http<br/>VpcEndpoints</b>       | HTTP-VPC service definition.                                                         |
| functionEndp<br>oints | No                | <b>x-<br/>apigateway-<br/>backend.func<br/>tionEndpoint<br/>s</b> | Function service definition.                                                         |
| mockEndpoint<br>s     | No                | <b>x-<br/>apigateway-<br/>backend.moc<br/>kEndpoints</b>          | Mock service definition.                                                             |

| Parameter  | Mandatory | Type                                            | Description               |
|------------|-----------|-------------------------------------------------|---------------------------|
| conditions | Yes       | <b>x-apigateway-backend-policies.conditions</b> | Backend policy condition. |

## 7.1: x-apigateway-backend-policies.conditions

**Meaning:** API backend policy.

**Scope of effect:** **x-apigateway-backend-policies**

### Example

```
paths:
  '/users/{userId}':
    get:
      produces:
        - "application/json"
      responses:
        default:
          description: "default response"
          x-apigateway-request-type: "public"
          x-apigateway-backend:
            type: "backend endpoint type"
          x-apigateway-backend-policies:
            - type: "backend endpoint type"
              name: "backend policy name"
              conditions:
                - type: "equal/enum/pattern",
                  value: "string",
                  origin: "source/request_parameter",
                  parameter_name: "string"
```

**Table 7-44** Parameter description

| Parameter | Mandatory | Type   | Description                                                                           |
|-----------|-----------|--------|---------------------------------------------------------------------------------------|
| type      | Yes       | String | Policy condition type. <b>equal</b> , <b>enum</b> , and <b>pattern</b> are supported. |
| value     | Yes       | String | Policy condition value.                                                               |
| origin    | Yes       | String | Policy condition source. <b>source</b> and <b>request</b> are supported.              |
| parameter | No        | String | Input parameter name if the <b>origin</b> parameter is set to <b>request</b> .        |

## 8: x-apigateway-ratelimit

**Meaning:** Request throttling policy.

**Scope of effect:** [Operation Object](#)

### Example

```
paths:
  '/path':
    get:
      x-apigateway-ratelimit: 'customRatelimitName'
```

**Table 7-45** Parameter description

| Parameter              | Mandatory | Type   | Description                                                                                          |
|------------------------|-----------|--------|------------------------------------------------------------------------------------------------------|
| x-apigateway-ratelimit | No        | String | Name of the referenced request throttling policy. Set this parameter to <b>customRatelimitName</b> . |

## 9: x-apigateway-ratelimits

**Meaning:** Mapping between a request throttling policy name and limit values.

**Scope of effect:** [Swagger Object](#)

### Example

```
x-apigateway-ratelimits:
  customRatelimitName:
    api-limit: 200
    app-limit: 200
    user-limit: 200
    ip-limit: 200
    interval: 1
    unit: second/minute/hour
    shared: true
    special:
      - type: APP
        limit: 100
    instance: xxxxxxxx
```

**Table 7-46** Parameter description

| Parameter           | Mandatory | Type                                           | Description                                                                                                                                 |
|---------------------|-----------|------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| customRatelimitName | No        | <a href="#">x-apigateway-ratelimits.policy</a> | Custom request throttling policy. To use a request throttling policy, set <a href="#">x-apigateway-ratelimit</a> to the name of the policy. |

### 9.1: x-apigateway-ratelimits.policy

**Meaning:** Definition of a request throttling policy.

**Scope of effect:** [x-apigateway-ratelimits](#)

**Example**

```
x-apigateway-ratelimits:
  customRatelimitName:
    api-limit: 200
    app-limit: 200
    user-limit: 200
    ip-limit: 200
    interval: 1
    unit: MINUTE
    shared: false
    special:
      - type: USER
        limit: 100
      instance: xxxxxxx
```

**Table 7-47** Parameter description

| Parameter  | Man<br>dator<br>y | Type                                                   | Description                                                                                  |
|------------|-------------------|--------------------------------------------------------|----------------------------------------------------------------------------------------------|
| api-limit  | Yes               | Number                                                 | Maximum number of times an API can be called.                                                |
| user-limit | No                | Number                                                 | Number of times the API can be called by a user.                                             |
| app-limit  | No                | Number                                                 | Number of times the API can be called by an app.                                             |
| ip-limit   | No                | Number                                                 | Number of times the API can be called by an IP address.                                      |
| interval   | Yes               | Number                                                 | Throttling period.                                                                           |
| unit       | Yes               | String                                                 | Throttling unit, which can be <b>SECOND</b> , <b>MINUTE</b> , <b>HOURL</b> , or <b>DAY</b> . |
| shared     | No                | Boolean                                                | Whether to share the throttling limits among APIs.                                           |
| special    | No                | <a href="#">x-apigateway-ratelimits.policy.special</a> | Special request throttling policy.                                                           |

**9.2: x-apigateway-ratelimits.policy.special**

**Meaning:** Definition of a special request throttling policy.

**Scope of effect:** [x-apigateway-ratelimits.policy](#)

**Example**

```
x-apigateway-ratelimits:
  customRatelimitName:
```

```
api-limit: 200
app-limit: 200
user-limit: 200
ip-limit: 200
interval: 1
unit: MINUTE
shared: false
special:
- type: USER
  limit: 100
  instance: xxxxxxxx
```

**Table 7-48** Parameter description

| Parameter | Mandator y | Type   | Description                                                                |
|-----------|------------|--------|----------------------------------------------------------------------------|
| type      | Yes        | String | Excluded request throttling type, which can be <b>APP</b> or <b>USER</b> . |
| limit     | Yes        | Number | Access limit.                                                              |
| instance  | Yes        | String | Excluded app or user                                                       |

## 10: x-apigateway-access-control

**Meaning:** Access control policy.

**Scope of effect:** [Operation Object](#)

### Example

```
paths:
  '/path':
    get:
      x-apigateway-access-control: 'customAccessControlName'
```

**Table 7-49** Parameter description

| Parameter                   | Mandat ory | Type   | Description                                                                                          |
|-----------------------------|------------|--------|------------------------------------------------------------------------------------------------------|
| x-apigateway-access-control | No         | String | Name of the referenced access control policy. Set this parameter to <b>customAccessControlName</b> . |

## 11: x-apigateway-access-controls

**Meaning:** Mapping between an access control policy name and limit settings.

**Scope of effect:** [Swagger Object](#)

### Example

```
x-apigateway-access-controls:
  customAccessControlName:
```



```
acl-type: "DENY"
entity-type: "IP"
value: 127.0.0.1,192.168.0.1/16
```

**Table 7-50** Parameter description

| Parameter               | Mandatory | Type                                                | Description                                                                                                                               |
|-------------------------|-----------|-----------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| customAccessControlName | No        | <a href="#">x-apigateway-access-controls.policy</a> | Custom access control policy. To use an access control policy, set <a href="#">x-apigateway-access-control</a> to the name of the policy. |

## 11.1: x-apigateway-access-controls.policy

**Meaning:** Definition of an access control policy.

**Scope of effect:** [x-apigateway-access-controls](#)

### Example

```
x-apigateway-access-controls:
  customAccessControlName:
    acl-type: "DENY"
    entity-type: "IP"
    value: 127.0.0.1,192.168.0.1/16
```

**Table 7-51** Parameter description

| Parameter   | Mandatory | Type   | Description                                                                |
|-------------|-----------|--------|----------------------------------------------------------------------------|
| acl-type    | Yes       | String | Access control effect. The options include <b>PERMIT</b> and <b>DENY</b> . |
| entity-type | Yes       | String | Access control object. Only IP addresses are supported.                    |
| value       | Yes       | String | Access control policy values. Use commas (,) to separate multiple values.  |

## 12: x-apigateway-roma-app

**Meaning:** Integration application bound to the API.

**Scope of effect:** [Operation Object](#)

### Example

```
paths:
  '/path':
```

```
get:
  x-apigateway-roma-app: 'romaAppName'
```

**Table 7-52** Parameter description

| Parameter             | Mandatory | Type   | Description                                           |
|-----------------------|-----------|--------|-------------------------------------------------------|
| x-apigateway-roma-app | Yes       | String | Name of the integration application bound to the API. |

## 7.6.7 Appendix: Extended Definition of Swagger for Custom Backends

On the basis of the basic Swagger definition, ROMA Connect defines the extended definition of APIs, such as the authentication mode and function script definition. This section describes the extended definitions of custom backends.

### 1: x-livedata-auth-type

**Meaning:** Swagger-based apiKey authentication format, which defines an authentication mode provided by the custom backend.

**Scope of effect:** [Security Scheme Object](#)

#### Example

```
securityDefinitions:
  customize-name-signature:
    type: "apiKey"
    name: "Authorization"
    in: "header"
    x-livedata-auth-type: "SIGNATURE"
  x-livedata-signature:
    key: "signatureKey"
    secret: "signatureSecret"
```

**Table 7-53** Parameter description

| Parameter   | Mandatory | Type   | Description                                                                                 |
|-------------|-----------|--------|---------------------------------------------------------------------------------------------|
| type        | Yes       | String | Authentication type. Only <b>apiKey</b> is supported.                                       |
| name        | Yes       | String | Name of the parameter used for authentication. Set this parameter to <b>Authorization</b> . |
| in          | Yes       | String | Location of the parameter. Only <b>header</b> is supported.                                 |
| description | No        | String | Description of the parameter.                                                               |

| Parameter                   | Mandatory | Type   | Description                                                                  |
|-----------------------------|-----------|--------|------------------------------------------------------------------------------|
| x-livodata-auth-type        | Yes       | String | Custom backend authentication mode. The value can only be <b>SIGNATURE</b> . |
| x-livodata-signature.key    | No        | String | Key required for signature.                                                  |
| x-livodata-signature.secret | No        | String | Secret required for signature.                                               |

## 2: x-livodata-version

**Meaning:** Version number of the custom backend.

**Scope of effect:** [Operation Object](#)

### Example

```
paths:
  '/path':
    get:
      x-livodata-version: '1.0.1'
```

**Table 7-54** Parameter description

| Parameter          | Mandatory | Type   | Description  |
|--------------------|-----------|--------|--------------|
| x-livodata-version | Yes       | String | API version. |

## 3: x-livodata-status

**Meaning:** Custom backend status.

**Scope of effect:** [Operation Object](#)

### Example

```
paths:
  '/path':
    get:
      x-livodata-status: 'DESIGNED'
```

**Table 7-55** Parameter description

| Parameter         | Mandatory | Type   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x-livedata-status | Yes       | String | <p>Status of the custom backend. The options are <b>DESIGNED</b>, <b>DEVELOPED</b>, <b>TESTED</b>, and <b>DEPLOYED</b>.</p> <ul style="list-style-type: none"> <li>• <b>DESIGNED</b>: The custom backend has been designed and is to be developed.</li> <li>• <b>DEVELOPED</b>: The custom backend has been developed and is to be tested.</li> <li>• <b>TESTED</b>: The custom backend has been tested and is to be deployed.</li> <li>• <b>DEPLOYED</b>: The custom backend has been deployed.</li> </ul> |

#### 4: x-livedata-roma-app

**Meaning:** Integration application bound to the custom backend.

**Scope of effect:** [Operation Object](#)

##### Example

```
paths:
  '/path':
    get:
      x-livedata-roma-app: 'romaAppName'
```

**Table 7-56** Parameter description

| Parameter           | Mandatory | Type   | Description                                          |
|---------------------|-----------|--------|------------------------------------------------------|
| x-livedata-roma-app | Yes       | String | Integration application bound to the custom backend. |

#### 5: x-livedata-scripts

**Meaning:** Script of the custom backend definition.

**Scope of effect:** [Operation Object](#)

##### Example

```
paths:
  '/path':
    get:
```

```

produces:
  - "application/json"
responses:
  default:
    description: "default response"
x-livadata-scripts:
  - type: "function"
    content: "custom-script-content"
    result: "func"
    
```

**Table 7-57** Parameter description

| Parameter   | Man<br>dator<br>y | Type                                       | Description                                                                 |
|-------------|-------------------|--------------------------------------------|-----------------------------------------------------------------------------|
| content     | Yes               | String                                     | Script statement.                                                           |
| result      | Yes               | String                                     | Response.                                                                   |
| type        | Yes               | String                                     | Script type. The options are <b>Function</b> , <b>SQL</b> , and <b>SP</b> . |
| datasources | No                | <b>x-livadata-<br/>scripts.datasources</b> | Data source definition.                                                     |

## 5.1 x-livadata-scripts.datasources

**Meaning:** data source definition of the custom backend.

**Scope of effect:** **x-livadata-scripts**

### Example

```

paths:
  '/users':
    get:
      produces:
        - "application/json"
      responses:
        default:
          description: "default response"
      x-livadata-scripts:
        - type: "function"
          content: "custom-script-content"
          result: "custom-script-result"
          datasource:
            name: "custom-datasource-name"
    
```

**Table 7-58** Parameter description

| Parameter | Man<br>dator<br>y | Type   | Description       |
|-----------|-------------------|--------|-------------------|
| name      | Yes               | String | Data source name. |

## 7.7 Control Policy Management

### 7.7.1 Configuring a Request Throttling Policy

#### Overview

Request throttling controls the number of times an API can be called within a period to protect the backend service. To provide continuous and stable services, create request throttling policies to control the number of calls made to your APIs.

A request throttling policy and an API are independent of each other. A request throttling policy takes effect for an API only after it is bound to the API.

#### NOTE

An API can be bound to only one request throttling policy in an environment, but each request throttling policy can be bound to multiple APIs.

#### Creating a Request Throttling Policy

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Request Throttling Policies** tab page, click **Create**.
3. In the **Create Request Throttling Policy** dialog box, configure policy information.

**Table 7-59** Parameters for creating a request throttling policy

| Parameter | Description                                                                                                                                                                                                                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name      | Enter a request throttling policy name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                               |
| Type      | Select the type of the request throttling policy. <ul style="list-style-type: none"> <li>• <b>API-based:</b> The API bound to a throttling policy consumes the throttling value of the policy.</li> <li>• <b>API-shared:</b> All APIs bound to a throttling policy consume the throttling value of the policy.</li> </ul> |

| Parameter                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Period                   | Enter the request throttling duration. The unit can be seconds, minutes, hours, or days. This parameter must be used together with the following request throttling parameters: <ul style="list-style-type: none"><li>• <b>Max. API Requests:</b> used to limit the total number of times an API can be called within a period.</li><li>• <b>Max. User Requests:</b> used to limit the number of times an API can be called by a user within a period.</li><li>• <b>Max. App Requests:</b> used to limit the total number of times an API can be called by an application within a period.</li><li>• <b>Max. IP Address Requests:</b> used to limit the total number of times an API can be called by an IP address within a period.</li></ul> |
| Max. API Requests        | Enter the maximum number of times that an API can be called. This parameter is used along with <b>Period</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Max. User Requests       | Enter the maximum number of times that an API can be called by a user. This parameter is used along with <b>Period</b> . The value of this parameter cannot be greater than that of <b>Max. API Requests</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Max. App Requests        | Enter the maximum number of times that an API can be called by an application. This parameter is used along with <b>Period</b> . The value of this parameter cannot be greater than that of <b>Max. API Requests</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Max. IP Address Requests | Enter the maximum number of times that an API can be called by an IP address. This parameter is used along with <b>Period</b> . The value of this parameter cannot be greater than that of <b>Max. API Requests</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Description              | Enter the descriptive information of the request throttling policy.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

4. Click **OK**.

After the request throttling policy is created, you also need to perform the operations described in [Binding a Request Throttling Policy to an API](#) to make the policy take effect for the API.

## Binding a Request Throttling Policy to an API

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Request Throttling Policies** tab page, click **Bind to API**.
3. On the **Bind to API** page, click **Select API**.
4. In the **Select API** dialog box, select the APIs to which the request throttling policy is to be bound in the specified environment.

You can filter the required APIs by API group, environment, or API name.

5. Click **OK** to bind the request throttling policy to the selected API.

## Binding a Request Throttling Policy to an Application

If you want to throttle requests for an integration application, you can add an excluded application to the request throttling policy. After an integration application is added to the request throttling policy, **Max. App Requests** of the application is restricted by the threshold of the excluded application, and **Max. API Requests** and **Max. User Requests** are restricted by the throttling policy.

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Request Throttling Policies** tab page, click the name of the request throttling policy that you want to bind to an application.
3. Click the **Excluded Apps** tab and click **Select Excluded App**.
4. In the **Select Excluded App** dialog box, configure application information.

**Table 7-60** Excluded app parameters

| Parameter | Description                                                                                                                                                                                                                                 |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| App       | Select the integration application to which the request throttling policy is to be bound.                                                                                                                                                   |
| Threshold | Enter the maximum number of times that an API can be called by the integration application within a specified period. The value of this parameter cannot be greater than that of <b>Max. API Requests</b> in the request throttling policy. |

5. Click **OK**.

## Binding a Request Throttling Policy to a Tenant

If you want to throttle requests for a tenant, you can add an excluded tenant to the request throttling policy. After the tenant is added to the request throttling policy, **Max. User Requests** of the tenant is limited by the threshold of the excluded tenant, and **Max. API Requests** and **Max. App Requests** are limited by the throttling policy.

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Request Throttling Policies** tab page, click the name of the request throttling policy that you want to bind to an application.
3. Click the **Excluded Tenants** tab and click **Select Excluded Tenant**.
4. In the **Select Excluded Tenant** dialog box, configure tenant information.



**Table 7-61** Parameters for configuring an excluded tenant

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tenant ID | <p>Enter the ID of the tenant to which the request throttling policy is to be bound.</p> <ul style="list-style-type: none"><li>• If the App authentication mode is used to call APIs, the tenant ID is the project ID of the user to which the integration application belongs.</li><li>• If the IAM authentication mode is used to call APIs, the tenant ID is the account ID of the caller.</li></ul> <p>You can click the username in the upper right corner of the console and choose <b>My Credentials</b> to obtain the project ID and account ID of the user on the <b>My Credentials</b> page.</p> |
| Threshold | <p>Enter the maximum number of times that an API can be called by the integration application within a specified period. The value of this parameter cannot be greater than that of <b>Max. API Requests</b> in the request throttling policy.</p>                                                                                                                                                                                                                                                                                                                                                         |

5. Click **OK**.

## 7.7.2 Configuring an Access Control Policy

### Overview

Access control allows you to control the IP addresses and accounts used to access APIs, protecting backend services. You can create an access control policy to allow or deny the access of certain IP addresses or accounts to an API.

An access control policy and an API are independent of each other. An access control policy takes effect for an API only after it is bound to the API.

#### NOTE

An API can be bound to only one access control policy in an environment, but each access control policy can be bound to multiple APIs.

### Creating an Access Control Policy

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Access Control Policies** tab page, click **Create**.
3. In the **Create Access Control Policy** dialog box, configure policy information.

**Table 7-62** Parameters for creating an access control policy

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name             | Enter an access control policy name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                         |
| Restriction Type | Select the restriction type of the access control policy. <ul style="list-style-type: none"> <li>• <b>IP address:</b> API calling is restricted based on IP addresses.</li> <li>• <b>Account name:</b> API calling is restricted based on account names. This option applies only to APIs that use the IAM authentication mode. The restriction also applies to the IAM users under the specified accounts. IAM users cannot be specified separately.</li> </ul> |
| Effect           | Select the action of access control. This parameter is used along with <b>Restriction Type</b> . <ul style="list-style-type: none"> <li>• <b>Allow:</b> Only specified IP addresses or accounts are allowed to call APIs.</li> <li>• <b>Deny:</b> Specified IP addresses or accounts are not allowed to call APIs.</li> </ul>                                                                                                                                    |
| IP Address       | This parameter is mandatory only if <b>Restriction Type</b> is set to <b>IP address</b> .<br>Click <b>Add IP Address</b> to add the IP addresses or IP address segments that are allowed or forbidden to call an API.                                                                                                                                                                                                                                            |
| Account Names    | This parameter is mandatory only if <b>Restriction Type</b> is set to <b>Account name</b> .<br>Enter the account names that are allowed or forbidden to call an API. Use commas (,) to separate multiple account names.<br>You can click the username in the upper right corner of the console and choose <b>My Credentials</b> to obtain the account name of the user on the <b>My Credentials</b> page.                                                        |

4. Click **OK**.

After the access control policy is created, you also need to perform the operations described in [Binding an Access Control Policy to an API](#) to make the policy take effect for the API.

### Binding an Access Control Policy to an API

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Access Control Policies** tab page, click **Bind to API**.
3. On the **Bind to API** page, click **Select API**.

4. In the **Select API** dialog box, select the APIs to which the access control policy is to be bound in the specified environment.  
You can filter the required APIs by API group, environment, and API name.
5. Click **OK**.

## 7.7.3 Configuring a Client Quota Policy

### Overview

A client quota limits the total number of times that an API can be called by a client in a specified period to protect backend services. You can create a client quota policy to limit the number of callings made by clients bound to the policy.

A client quota policy and a client are independent of each other. The client quota policy takes effect on the client only after the client is bound to the client quota policy.

Only users with the Tenant Administrator permission can view and configure client quota policies.

#### NOTE

One client can be bound to only one client quota policy, but one client quota policy can be bound to multiple clients.

### Creating a Client Quota Policy

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Calling**. On the **Client Quota Policies** tab page, click **Create**.
3. In the **Create Client Quota Policy** dialog box, configure policy information.

**Table 7-63** Parameters for creating a client quota policy

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                     |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                   | Enter a client quota policy name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                           |
| Initial Effective Time | Select the start time for the quota policy to take effect. For example, if Initial Effective Time is set to <b>2020/08/08 05:05:00</b> and <b>Period</b> is set to 1 hour, the client quota policy takes effect starting from 2020/08/08 05:05:00. The period from the fifth minute of an hour to the fifth minute of the next hour is a cycle, for example, 05:05:00-06:05:00. |

| Parameter         | Description                                                                                                                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Period            | Enter the period in which the quota policy is applied. The unit can be second, minute, hour, or day. This parameter must be used along with <b>Max. API Requests</b> to limit the total number of times an API can be called by a client within the specified period. |
| Max. API Callings | Enter the maximum number of times that an API can be called by a client. This parameter is used along with <b>Period</b> .                                                                                                                                            |
| Description       | Enter a brief description of the client quota policy.                                                                                                                                                                                                                 |

4. Click **OK**.

After the client quota policy is created, you also need to perform the operation described in [Binding a Quota Policy to a Client](#) to make the policy take effect for APIs.

## Binding a Quota Policy to a Client

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Calling**. On the **Client Quota Policies** tab page, click **Bind to Client** in the **Operation** column of a policy.
3. On the **Bind to Client** page, click **Select Client**.
4. In the **Select Client** dialog box, select the client to which the client quota policy is to be bound.

You can enter a client name to filter the required client.

5. Click **Bind**.

A client can be bound to only one quota policy. If a client is bound to a quota policy repeatedly, the original quota policy will be unbound.

## 7.7.4 Configuring a Client Access Control Policy

### Overview

Client access control controls the client IP addresses that access APIs to protect backend services. You can configure an access control policy to allow or forbid a client with the specified IP address to access an API.

### Configuring an Access Control Policy

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Calling**. On the **Clients** tab page, click **Set Access Control** in the **Operation** column of a client.

3. In the **Set Access Control Policy** dialog box, configure policy information.

**Table 7-64** Parameters for setting an access control policy

| Parameter  | Description                                                                                                                                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Effect     | Select an access control action. <ul style="list-style-type: none"> <li>• <b>Allow:</b> Only clients with specified IP addresses are allowed to call APIs.</li> <li>• <b>Deny:</b> Clients with specified IP addresses are not allowed to call APIs.</li> </ul> |
| IP Address | Click <b>Add IP Address</b> to add the client IP addresses or IP address segments that are allowed or forbidden to call an API.                                                                                                                                 |

4. Click **OK**.

After configuring an access control policy, you can click **Reset** in the **Set Access Control Policy** dialog box to clear all access control policies.

## 7.8 Configuring a Custom Authorizer

### 7.8.1 Creating a Frontend Custom Authorizer

#### Overview

If you need to use your own authentication system for API calling authentication, you can create a custom authorizer.

Custom authorizers are classified into the following types:

- Frontend custom authorizer: ROMA Connect uses a custom authentication function to authenticating received API requests.
- Backend custom authorizer: The backend service of an API uses a custom authentication function to authenticating backend service requests forwarded by ROMA Connect.

This section describes how to create a frontend custom authorizer. You need to create a function backend as the authentication function, and use the function backend as the authentication backend in custom authentication.

#### Creating a Function Backend for Frontend Authentication

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Create**.
3. On the **Create Backend** page, set backend parameters and click **Create**.
  - **Backend Request Method** must be set to **POST**.

- You do not need to set input parameters. The function backend used for custom authentication will obtain the values of Header and Query parameters from API requests.
- For details about the settings of other parameters, see [Creating a Function API](#).

After the backend is created, the online IDE page is automatically displayed.

#### 4. Develop a function backend.

In the upper left corner of the online IDE, choose **File > Create Function Backend > Blank Template**. In the dialog box displayed, click **Yes**. Compile a function script for security authentication and click **Save**.

The function script used for frontend custom authentication must meet the following conditions:

- For the request parameters obtained using the function script:
  - Header parameter: indicates the identity source parameter in Header defined in custom authentication. The parameter value is transferred from the API request that uses the frontend custom authentication. The format of the invoked parameter in the function script is as follows: `body["headers"]["Parameter name"]`.
  - Query parameter: indicates the identity source parameter in Query defined in custom authentication. The parameter value is transferred from the API request that uses the frontend custom authentication. The format of the invoked parameter in the function script is as follows: `body["queryStringParameters"]["Parameter name"]`.
  - Body parameter: indicates the user data defined in custom authentication. The parameter value is set when the custom authentication is created. The format of the invoked parameter in the function script is as follows: `body["user_data"]`.
- For the response message defined by the function script:

The response body cannot be greater than 1 MB. The response content must meet the following format:

```
{
  "status": "allow/deny",
  "context": {
    "user": "abc"
  }
}
```

- **status**: identifies the authentication result. This field is mandatory. Only **allow** or **deny** is supported. **allow** indicates that the authentication is successful, and **deny** indicates that the authentication fails.
- **context**: indicates the authentication response result. This field is mandatory. Only key-value pairs of the string type are supported. The key value does not support JSON objects or arrays.  
The data in the context is user-defined. After the authentication is successful, the data can be used as a **system parameter** (frontend authentication parameter) and mapped to the backend request parameter of the API. The system parameter name set in the API backend service must be the same as the parameter name in the

context. The parameter name is case sensitive. The parameter name in **context** must start with a letter and contain 1 to 32 characters, including uppercase letters, lowercase letters, digits, underscores (\_), and hyphens (-).

The following is an example of the header parameter definition script:

```
function execute(data){
  data=JSON.parse(data)
  body=data.body
  if(body["headers"]["test"]=='abc'){
    return{
      "status": "allow",
      "context": {
        "user": "abcd"
      }
    }
  }else{
    return{
      "status": "deny"
    }
  }
}
```

The following is an example of the query parameter definition script:

```
function execute(data){
  data=JSON.parse(data)
  body=data.body
  if(body["queryStringParameters"]["test"]=='abc'){
    return{
      "status": "allow",
      "context": {
        "user": "abcd"
      }
    }
  }else{
    return{
      "status": "deny"
    }
  }
}
```

The following is an example of the user data definition script:

```
function execute(data){
  data=JSON.parse(data)
  body=data.body
  if(body["user_data"]=='abc'){
    return{
      "status": "allow",
      "context": {
        "user": "abcd"
      }
    }
  }else{
    return{
      "status": "deny"
    }
  }
}
```

5. Test the function backend.

In the upper right corner of the page, click **Test**. In the **Test Parameters** area, add request parameters required for authentication based on the definition of the function backend and click **Test** to send the request.

The user data definition script example in the preceding step is used as an example. You need to enter the request content {"user\_data": "abc"} in the Body parameter as the authentication parameter of the backend request.

If the value of status in the test result is **allow**, the test is successful.

6. Deploy the function backend.

After the backend test is complete, click **Deploy** in the upper right corner of the page. In the dialog box displayed, click **Yes** to deploy the function backend.

## Creating a Frontend Custom Authorizer

Before creating a frontend custom authorizer, ensure that the function backend used for frontend custom authentication has been created. Otherwise, create a function backend first. For details, see [Creating a Function Backend for Frontend Authentication](#).

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Custom Authorizers** tab page, click **Create**.
3. In the **Create Custom Authorizer** dialog box, configure custom authorizer information and click **Create**.

**Table 7-65** Parameters for creating a frontend custom authorizer

| Parameter               | Description                                                                                                                                                                                                                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a custom authorizer name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                           |
| Type                    | Select <b>Frontend</b> .                                                                                                                                                                                                                                                                                      |
| Integration Application | Select the integration application to which the custom authorizer belongs.                                                                                                                                                                                                                                    |
| Function URN            | Select the function backend used for frontend custom authentication. Only function backends in the <b>Deployed</b> state can be selected.                                                                                                                                                                     |
| Identity Sources        | Add request parameters used for authentication. The Header and Query parameters can be added.<br>If the value of <b>Cache Duration</b> is not <b>0</b> , a request parameter must be added. When the authentication result is cached, this parameter is used as the cache index of the authentication result. |
| Cache Duration (s)      | Enter the cache time of the authentication result. If it is set to <b>0</b> , the authentication result is not cached. The maximum cache duration can be set to 3600 seconds.                                                                                                                                 |
| Send Request Body       | Determine whether to send the API request body to the authentication function.                                                                                                                                                                                                                                |
| User Data               | Enter the user-defined authentication request parameter, which is used together with <b>Identity Sources</b> as the authentication request parameter.                                                                                                                                                         |



## 7.8.2 Creating a Backend Custom Authorizer

### Overview

If you need to use your own authentication system for authenticating backend service requests, you can create a custom authorizer.

Custom authorizers are classified into the following types:

- Frontend custom authorizer: ROMA Connect uses a custom authentication function to authenticating received API requests.
- Backend custom authorizer: The backend service of an API uses a custom authentication function to authenticating backend service requests forwarded by ROMA Connect.

This section describes how to create a backend custom authorizer. You need to create a function backend as the authentication function, and use the function backend as the authentication backend in custom authentication.

### Creating a Function Backend for Backend Authentication

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > Custom Backend**. On the **Backends** tab page, click **Create**.
3. On the **Create Backend** page, set backend parameters and click **Create**.
  - **Backend Request Method** must be set to **POST**.
  - You do not need to set input parameters. The Header and Query parameters are invalid in the function backend used for backend custom authentication.
  - For details about the settings of other parameters, see [Creating a Function API](#).

After the backend is created, the online IDE page is automatically displayed.

4. Develop a function backend.

In the upper left corner of the online IDE, choose **File > Create Function Backend > Blank Template**. In the dialog box displayed, click **Yes**. Compile a function script for security authentication and click **Save**.

The function script used for backend custom authentication must meet the following conditions:

- For the request parameters invoked using the function script:
  - Body parameter: indicates the user data defined in a custom authorizer. The parameter value is specified when a custom authorizer is created. The format of the invoked parameter in the function script is as follows: `body["user_data"]`.
- For the response message defined by the function script:
  - The response body cannot be greater than 1 MB. The response content must meet the following format:

```
{
  "status": "allow/deny",
  "context": {
    "user": "abc"
  }
}
```

- **status:** identifies the authentication result. This field is mandatory. Only **allow** or **deny** is supported. **allow** indicates that the authentication is successful, and **deny** indicates that the authentication fails.
- **context:** indicates the authentication response result. This field is mandatory. Only key-value pairs of the string type are supported. The key value does not support JSON objects or arrays.  
The data in the context is user-defined. After the authentication is successful, the data can be used as a **system parameter** (backend authentication parameter) and mapped to the backend request parameter of the API. The system parameter name set in the API backend service must be the same as the parameter name in the context. The parameter name is case sensitive. The parameter name in **context** must start with a letter and contain 1 to 32 characters, including letters, digits, underscores (\_), and hyphens (-).

The following is an example of the user data definition script:

```
function execute(data){
  data=JSON.parse(data)
  body=data.body
  if(body["user_data"]=='abc'){
    return{
      "status": "allow",
      "context": {
        "user": "abcd"
      }
    }
  }
  }else{
    return{
      "status": "deny"
    }
  }
}
```

5. Test the function backend.

In the upper right corner of the page, click **Test**. In the **Test Parameters** area, add request parameters required for authentication based on the definition of the function backend and click **Test** to send the request.

The user data definition script example in the preceding step is used as an example. You need to enter the request content {"user\_data": "abc"} in the Body parameter as the authentication parameter of the backend request.

If the value of status in the test result is **allow**, the test is successful.

6. Deploy the function backend.

After the backend test is complete, click **Deploy** in the upper right corner of the page. In the dialog box displayed, click **Yes** to deploy the function backend.

## Creating a Backend Custom Authorizer

Before creating a backend custom authorizer, ensure that the function backend used for backend custom authentication has been created. Otherwise, create a

function API first. For details, see [Creating a Function Backend for Backend Authentication](#).

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Custom Authorizers** tab page, click **Create**.
3. In the **Create Custom Authorizer** dialog box, configure custom authorizer information and click **Create**.

**Table 7-66** Parameters for creating a backend custom authorizer

| Parameter               | Description                                                                                                                                                                   |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a custom authorizer name. It is recommended that you enter a name based on naming rules to facilitate search.                                                           |
| Type                    | Select <b>Backend</b> .                                                                                                                                                       |
| Integration Application | Select the integration application to which the custom authorizer belongs.                                                                                                    |
| Function URN            | Select the function backend used for backend custom authentication. Only function backends in the <b>Deployed</b> state can be selected.                                      |
| Cache Duration (s)      | Enter the cache time of the authentication result. If it is set to <b>0</b> , the authentication result is not cached. The maximum cache duration can be set to 3600 seconds. |
| Send Request Body       | Determine whether to send the API request body to the authentication function.                                                                                                |
| User Data               | Enter the user-defined authentication request parameter.                                                                                                                      |

## 7.9 Configuring Signature Verification for Backend Services

### Overview

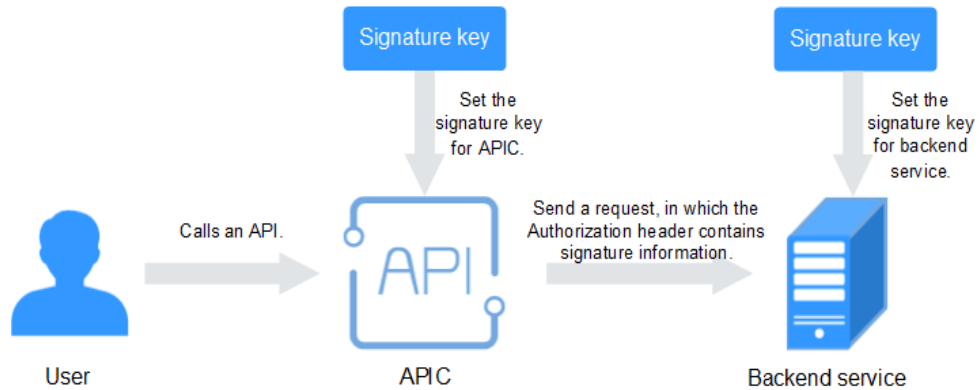
Signature keys are used by backend services to verify the identity of ROMA Connect to ensure secure access.

A signature key consists of a key and a secret. The signature key takes effect only after it is bound to an API.

#### NOTE

An API can be bound to only one signature key in an environment, but a signature key can be bound to multiple APIs.

After a signature key is bound to an API, ROMA Connect uses the key and secret in the signature key to add signature information to requests sent to the backend service of the API. In this case, the backend service needs to sign the request in the same way. If the signature is the same as that carried in the Authorization header of the request, the backend service determines that the request sent by ROMA Connect is valid.



### Creating a Signature Key

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Signature Keys** tab page, click **Create**.
3. In the **Create Signature Key** dialog box, configure signature key information.

**Table 7-67** Parameters for creating a signature key

| Parameter      | Description                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name           | Enter a signature key name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                               |
| Type           | Select the type of the signature key. The value can be <b>hmac</b> or <b>basic</b> .                                                                                                                                                                                                                                          |
| Key            | Enter the key information based on the value of <b>Type</b> . <ul style="list-style-type: none"> <li>• If <b>Type</b> is set to <b>hmac</b>, enter the key in the key pair used for HMAC authentication.</li> <li>• If <b>Type</b> is set to <b>basic</b>, enter the username used for basic authentication.</li> </ul>       |
| Secret         | Enter the secret information based on the value of <b>Type</b> . <ul style="list-style-type: none"> <li>• If <b>Type</b> is set to <b>hmac</b>, enter the secret in the key pair used for HMAC authentication.</li> <li>• If <b>Type</b> is set to <b>basic</b>, enter the password used for basic authentication.</li> </ul> |
| Confirm Secret | Enter the same secret again.                                                                                                                                                                                                                                                                                                  |

4. Click **OK**.

After the signature key is created, you also need to perform the operations described in [Binding a Signature Key to an API](#) to make the signature key take effect for the API.

## Binding a Signature Key to an API

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **API Connect > API Management**. On the **Signature Keys** tab page, click **Bind to API**.
3. On the **Bind to API** page, click **Select API**.
4. In the **Select API** dialog box, select the APIs to which the signature key is to be bound in the specified environment.  
You can filter the required APIs by API group, environment, and API name.
5. Click **OK**.

After the API is bound with a signature key, sign the backend service and check whether the signature result is consistent with the signature carried in the Authorization header of the request. For details, see [Signing Backend Services](#).

## 7.10 Configuring API Cascading

### Overview

API cascading allows you to cascade two ROMA Connect instances in the same region or different regions so that one instance can use an API of the other instance as its backend service, thereby implementing cross-instance API calling. A dedicated authentication channel will be used for API calling between the two instances that establish the cascading relationship. This can prevent an authentication conflict in API cascading scenarios.


- The instance that uses an API of the other instance as its backend service can be called a cascading instance.
- The instance that provides its API to the other instance as a backend service can be called a cascaded instance.

With the API cascading function, API providers can provide APIs in one instance to another instance to improve the reuse capability of API assets and avoid repeatedly deploying backend services in different instances.

### Prerequisites


- The cascading and cascaded instances can communicate with each other.
- If the cascading and cascaded instances are located in different networks and they communicate with each other through an air wall, the IP address and port number of the instances must be configured on the air wall. In addition, the TCP protocol must be used on the air wall to establish network paths in ferry mode. A dedicated VPN or tunnel can also be used to implement cross-network interworking.

## Procedure

1. Enable the cascading function on the cascaded instance.
  - a. Log in to the ROMA Connect console on which the cascaded instance is located. On the **Instances** page, click **View Console**.
  - b. On the **Instance Information** page, click the **Configuration Parameters** tab and locate the **cascade** parameter.
  - c. Click **Edit** on the right of the parameter, set **Current Value** to **on**, and click **Save**.
  - d. Click  on the left of the parameter to expand and set the parameters related to the cascading function.

**Table 7-68** Parameters related to the cascading function

| Parameter            | Description                                                                                                                                                                                                                                        |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cascade_auth_key     | Encryption key used for authentication between APIs in the cascading relationship. The <b>cascade_auth_key</b> value of the cascaded instance must be the same as that of the cascading instance.                                                  |
| cascade_instance_ids | ID list of cascading instances. Only instances specified by this parameter can establish the cascading relationship with the current instance. Use commas (,) to separate multiple instance IDs. A maximum of five instance IDs can be configured. |

2. Enable the cascading function on the cascading instance.
  - a. Log in to the ROMA Connect console on which the cascading instance is located. On the **Instances** page, click **View Console**.
  - b. On the **Instance Information** page, click the **Configuration Parameters** tab and locate the **cascade** parameter.
  - c. Click **Edit** on the right of the parameter, set **Current Value** to **on**, and click **Save**.
  - d. Click  on the left of the parameter to expand and set the cascading parameters.

**Table 7-69** Parameters related to the cascading function

| Parameter            | Description                                                                                                                                                                                           |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cascade_auth_key     | Encryption key used for calling authentication between APIs in cascading relationship. The <b>cascade_auth_key</b> value of the cascaded instance must be the same as that of the cascading instance. |
| cascade_instance_ids | ID list of cascading instances. You do not need to set this parameter in the cascading instance.                                                                                                      |

3. Create a load balance channel from the cascading instance to the cascaded instance.
  - a. In the navigation pane of the cascading instance console, choose **API Connect > API Management**. On the **Load Balance Channels** tab page, click **Create**.
  - b. On the **Create Load Balance Channel** page, configure basic information and health check information and click **Next**.

**Table 7-70** Parameters in Basic Information and Health Check Configuration

| Parameter              | Description                                                                                                                                                                                                                                                      |
|------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                   | Enter a load balance channel name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                           |
| Port                   | Enter the access port number of the ECS in the load balance channel. You can determine the port number based on the protocol used by the API in the cascaded instance. For HTTP, set this parameter to <b>80</b> . For HTTPS, set this parameter to <b>443</b> . |
| Member Type            | Select the member type of a server in the load balance channel. To access an API of the cascaded instance, select <b>IP address</b> for <b>Member Type</b> .                                                                                                     |
| Distribution Algorithm | Select a distribution algorithm for backend service requests. The load balance channel determines the server to which the request is to be sent based on the distribution algorithm.                                                                             |
| Protocol               | Select the protocol used for the health check. The value can be <b>TCP</b> , <b>HTTP</b> , or <b>HTTPS</b> .                                                                                                                                                     |
| Path                   | This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTP</b> or <b>HTTPS</b> .<br>Enter the health check URL.                                                                                                                                       |
| Method                 | This parameter is available only if <b>Protocol</b> is set to <b>HTTP</b> or <b>HTTPS</b> .<br>Select the HTTP request method used for the health check. The value can be <b>GET</b> or <b>HEAD</b> .                                                            |
| Advanced Settings      |                                                                                                                                                                                                                                                                  |
| Health Check Port      | Enter the destination port of the health check. By default, the port number configured for the load balance channel is used.                                                                                                                                     |

| Parameter           | Description                                                                                                                                                                                                                                                        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Healthy Threshold   | Number of consecutive successful checks required for an ECS to be considered healthy. For example, if <b>Healthy Threshold</b> is set to <b>2</b> , ROMA Connect considers the ECS status as healthy when the check is successful for two consecutive times.       |
| Unhealthy Threshold | Number of consecutive failed checks required for an ECS to be considered unhealthy. For example, if <b>Unhealthy Threshold</b> is set to <b>5</b> , ROMA Connect considers the ECS status as abnormal when the check fails for five consecutive times.             |
| Timeout (s)         | Enter the response timeout duration of a health check, in the unit of seconds. If no response is received within the specified duration, the health check fails.                                                                                                   |
| Interval (s)        | Interval between consecutive checks.                                                                                                                                                                                                                               |
| Response Codes      | This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTP</b> or <b>HTTPS</b> .<br>When the server returns a specified HTTP response code, the server considers the response as successful. Multiple response codes can be specified at the same time. |

- c. Click **Add Backend Server Address**. On the page displayed, configure the IP address of the cascaded instance.

**Table 7-71** Backend server information

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backend Server Address | Enter the API access address of the cascaded instance. <ul style="list-style-type: none"> <li>• If you use the IP address format: <ul style="list-style-type: none"> <li>– Set this parameter to the EIP of the cascaded instance if the cascading and cascaded instances communicate with each other over a public network based on the EIP.</li> <li>– Set this parameter to the APIC connection address of the cascaded instance if the cascading and cascaded instances communicate with each other over an intranet.</li> </ul> </li> <li>• If you use the domain name format, enter the access domain name of the API.</li> </ul> |



| Parameter | Description                                                                                                                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Weight    | <p>This parameter is mandatory only if <b>WRR</b> or <b>WLC</b> is selected for <b>Distribution Algorithm</b> of the load balance channel.</p> <p>Enter the weight of the backend server. A larger weight indicates that more requests will be forwarded to the backend server.</p> |

- d. Click **Finish**.
4. On the cascading instance, create an API and set the backend address to the API in the cascaded instance.

For details about how to create an API, see [Creating an API](#). Only the configuration of defining the backend service is different. The following table describes the different configuration.

**Table 7-72** Backend service access parameters

| Parameter            | Description                                                                                                                                                                                                                                                         |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Backend Type         | Select a backend service type. When the API of the cascaded instance is used as the backend service, select <b>HTTP/HTTPS</b> .                                                                                                                                     |
| Protocol             | Set this parameter based on the request protocol of the API in the cascaded instance.                                                                                                                                                                               |
| Request Mode         | Set this parameter based on the API request method used in the cascaded instance.                                                                                                                                                                                   |
| Load Balancing       | Determine whether to use a load balance channel to access backend services. When the API of the cascaded instance is used as the backend service, select <b>Configure now</b> .                                                                                     |
| Load Balance Channel | Select the load balance channel created in <a href="#">3</a> .                                                                                                                                                                                                      |
| Cascading Flag       | Determine whether to use the cascading mode to access backend services. Enable this option.                                                                                                                                                                         |
| Host Header          | Define the Host header field carried in the backend service request. If you have specified <b>Backend Server Address</b> with an IP address when creating a load balance channel in, set <b>Host Header</b> to the domain name of the API of the cascaded instance. |

| Parameter              | Description                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Path                   | <p>Enter the request path of the backend service in the <i>/getUserInfo/{userId}</i> format. The request path can contain the path parameter in the format of <i>{Parameter name}</i>.</p> <p>If the path needs to contain an environment variable, enclose the environment variable in number signs (#), for example, <i>/#path#</i>. Multiple environment variables can be added, for example, <i>/#path##request#</i>.</p> |
| Timeout (ms)           | <p>Enter the timeout interval of a backend service request. The default value is <b>5000</b>.</p>                                                                                                                                                                                                                                                                                                                             |
| Two-way Authentication | <p>This parameter is mandatory only if <b>Protocol</b> is set to <b>HTTPS</b>.</p> <p>Determine whether to enable two-way authentication between ROMA Connect and backend services. When the API of the cascaded instance is used as the backend service, do not enable two-way authentication.</p>                                                                                                                           |
| Backend Authentication | <p>Determine whether to enable backend authentication. When the API of the cascaded instance is used as the backend service, do not enable backend authentication.</p>                                                                                                                                                                                                                                                        |

# 8 Message Integration Guide

---

[Usage Introduction](#)

[Creating a Topic](#)

[\(Optional\) Granting Permissions for Topics](#)

[Connecting to a Topic](#)

[Topic Management](#)

## 8.1 Usage Introduction

### Function Description

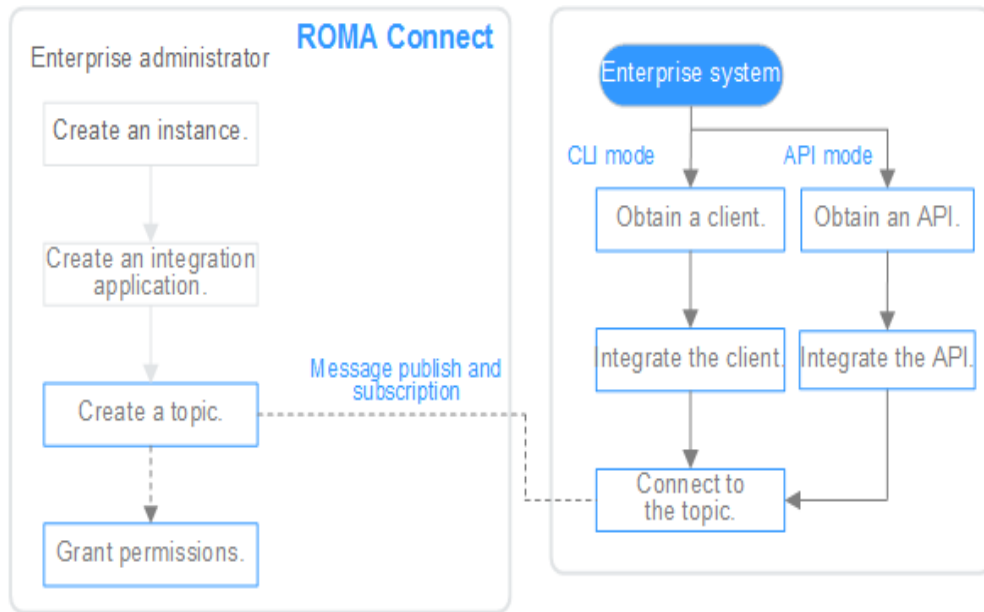
MQS is a message integration component of ROMA Connect. It uses a unified message access mechanism to provide secure and standard message channels for cross-network access. ROMA Connect has the following advantages for message integration:

- **Support for Kafka-native features**  
DMS for Kafka is compatible with open-source Kafka APIs and supports all message processing functions of open-source Kafka.
- **Secure message transmission**  
Operations on RabbitMQ instances are recorded and can be audited. Messages can be encrypted before storage. SASL authentication and security groups are used to enhance network access control.
- **High reliability of message data**  
MQS instances support data persistence and replication. Messages can be replicated synchronously or asynchronously between replicas.

### Process Flow

The following figure shows the process of using ROMA Connect for message integration.

**Figure 8-1** Process of using ROMA Connect for message integration



1. You have **created an instance and integration application.**
2. **Create a topic.**  
Create a topic for storing messages so that message producers can publish messages and message consumers can subscribe to messages.
3. (Optional) **Grant permissions for topics.**  
If SASL\_SSL of MQS is enabled in the ROMA Connect instance, the client needs to use the key and secret of the authorized integration application for security authentication when publishing and subscribing to messages to the topic.
4. **Connect to the topic.**  
The open-source Kafka client needs to be integrated to the system. Then, messages can be produced and consumed through the command lines provided by the client.

## 8.2 Creating a Topic

### Overview

Create a topic for storing messages so that producers can publish messages and consumers can subscribe to messages in it.

### Prerequisites

Each topic must belong to an integration application. Before creating a topic, ensure that an integration application is available. Otherwise, **create an integration application** first.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **Message Queue Service > Topic Management**. On the page displayed, click **Create Topic** in the upper right corner.
3. In the dialog box displayed, configure topic information and click **OK**.

**Table 8-1** Parameters for creating a topic

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Topic Name              | Enter a topic name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                     |
| Integration Application | Select the integration application to which the topic belongs.                                                                                                                                                                                                                                                                                                                              |
| Permission              | Select the permission on a topic for the integration application to which the topic belongs. <ul style="list-style-type: none"> <li>● <b>Publish/Subscribe</b>: Messages can be published to and subscribed from the topic.</li> <li>● <b>Publish</b>: Messages can only be published to the topic.</li> <li>● <b>Subscribe</b>: Messages can only subscribed from in the topic.</li> </ul> |
| Partitions              | Set the number of partitions of a topic to improve the concurrent performance of message production and consumption.<br>If this parameter is set to <b>1</b> , messages will be consumed in the FIFO order.                                                                                                                                                                                 |
| Replicas                | Set the number of replicas of a topic. ROMA Connect automatically backs up data on each replica. If one replica is faulty, data will still be available. The reliability increases with the number of replicas of a topic.                                                                                                                                                                  |
| Aging Time (h)          | Set the aging time of messages stored in the topic. Messages that exceed the aging time will be deleted.                                                                                                                                                                                                                                                                                    |
| Synchronous Replication | When a client creates a message to a topic, ROMA Connect determines whether to replicate the message to all replicas and then returns a response to the client.<br>After synchronous replication is enabled, you also need to set <b>acks</b> to <b>-1</b> on the client. Otherwise, synchronous replication will not take effect.                                                          |
| Synchronous Flushing    | This parameter specifies whether each message created by a client to the topic is immediately written to the disk. When synchronous flushing is enabled, the reliability is enhanced.                                                                                                                                                                                                       |

| Parameter        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tag Name         | Add message publishing and subscription tags for the integration application to which a topic belongs. Generally, you do not need to add the tags. The tags are used only in special service scenarios.<br><br>After a tag is added, the client needs to transmit the tag information when publishing and subscribing to messages in the topic. If there are multiple tags, the tags transmitted by the client must be a subset of the tags contained in the topic. |
| Sensitive Fields | Enter sensitive fields to messages in a topic. If a message sent to the topic contains sensitive fields, the sensitive fields will be shielded.                                                                                                                                                                                                                                                                                                                     |
| Description      | Enter a brief description of the topic.                                                                                                                                                                                                                                                                                                                                                                                                                             |

## 8.3 (Optional) Granting Permissions for Topics

### Overview

You need to grant permissions to integration applications only if MQS SASL\_SSL is enabled for the ROMA Connect instance so that the integration applications can send messages to or receive messages from topics. When a client publishes messages into a topic and subscribes to a topic to obtain messages, the key and secret of the authorized integration application must be used for security authentication.

By default, the integration application to which a topic belongs has the permissions to publish and subscribe to messages in the topic.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Message Queue Service > Topic Management**. On the page displayed, click **Grant Integration Application Permission** in the **Operation** column of the topic to which you want to grant permissions.
3. In the **Grant Integration Application Permission** dialog box, grant permissions for integration applications.

In the **Available** area, select the integration applications to be authorized. In the **Selected/All** area, configure the topic permissions for the selected integration applications.

**Table 8-2** Authorization configuration

| Parameter  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Permission | Select the topic permissions of the integration application. <ul style="list-style-type: none"><li>• <b>Publish/Subscribe:</b> Messages can be published to and subscribed from a topic.</li><li>• <b>Publish:</b> Messages can be only published to a topic.</li><li>• <b>Subscribe:</b> Messages can be only subscribed from a topic.</li></ul>                                                                                          |
| Tag        | Add tags for publishing and subscribing to messages of a topic. Tags are used only in special service scenarios and do not need to be configured in general service scenarios.<br><br>After a tag is added, the client needs to transmit the tag information when publishing and subscribing to messages in the topic. If there are multiple tags, the tags transmitted by the client must be a subset of the tags contained in the topic. |

4. Click **OK**.

## 8.4 Connecting to a Topic

### Overview

After a topic is created, you can use the CLI or open-source Kafka client to connect to the topic and produce and consume messages in the topic.

This section describes how to connect to a topic using the CLI. If you use an open-source Kafka client, see [Recommendations for Client Usage](#).

### Prerequisites

- A topic is available. If no topic is available, [create a topic](#) first.
- The open-source Kafka command-line tool of the corresponding version has been downloaded based on the Kafka version of the ROMA Connect instance. You can view the Kafka version in the **MQS Information** area on the **Instance Information** page of the ROMA Connect console.
  - [Kafka command line tool 1.1.0](#)
  - [Kafka command line tool 2.3.0](#)
- If **MQS SASL\_SSL** has been enabled for the ROMA Connect instance, ensure that the client certificate has been downloaded. To download the client certificate, log in to the ROMA Connect console, choose **Message Queue Service > Topic Management**, and click **Download SSL Certificate**.
- **Java JDK** has been installed in the environment where the Kafka command-line tool is used, and related environment variables have been configured.

 NOTE

- The Kafka server version of ROMA Connect can be 1.1.0 or 2.3.0. Therefore, you must use the Kafka command-line tool or client of the same version as the server to avoid unpredictable problems.
- If both SASL\_SSL and intra-VPC plaintext access are enabled for MQS of the ROMA Connect instance, the SASL mode cannot be used for connecting to MQS topics in the VPC.
- If the SASL mode is used for connecting to MQS topics, you are advised to configure the mapping between the host and IP address in the `/etc/hosts` file on the host where the client is located. Otherwise, network delay will occur.

Set the IP address to the connection address of MQS. Set the host to the name of each instance host. Ensure that the name of each host is unique. For example:

```
10.10.10.11 host01
10.10.10.12 host02
10.10.10.13 host03
```

## Using SASL Authentication

If SASL\_SSL access is enabled for the ROMA Connect instance, messages produced and consumed by the client to the topic are encrypted for transmission, which is more secure. The following uses Linux as an example.

1. Decompress the Kafka command-line tool and client certificates.

Access the directory where the tool package is stored and run the following command to decompress the package:

- Run the following command to decompress the command-line tool package:  

```
tar -zxf kafka_tar
```

In the preceding command, **kafka\_tar** indicates the name of the Kafka command-line tool package.

- Run the following command to decompress the client certificate file package:  

```
unzip cert_zip
```

In the preceding command, **cert\_zip** indicates the name of the client certificate file package.

2. Modify the configuration file of the Kafka command-line tool.

Find the **consumer.properties** and **producer.properties** files in the `/config` directory of the Kafka command-line tool and add the following content to the files:

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule required \
username="*****" \
password="*****",
sasl.mechanism=PLAIN
security.protocol=SASL_SSL
ssl.truststore.location=/cert/client.truststore.jks
ssl.truststore.password=dms@kafka
ssl.endpoint.identification.algorithm=
```

- The values of **username** and **password** are the key and secret of the integration application to which the topic belongs. For details on how to obtain the key and secret, see [Viewing and Editing an Integration Application](#).



- The value of **ssl.truststore.location** is the path for storing the client certificate obtained after decompression in **1**. Replace it with the actual path. Note that the certificate path in Windows must contain slashes (/).
  - **ssl.truststore.password** indicates the server certificate password, which must be set to **dms@kafka** and cannot be changed.
3. Access the **/bin** directory of the Kafka command-line tool.  
In the Windows operating system, you need to access the **/bin/windows** directory.
  4. Produce a message to the topic.
    - a. Run the following command to create a connection with the topic for producing messages:  

```
./kafka-console-producer.sh --broker-list Address --topic TopicName --producer.config ../config/producer.properties
```

In the preceding command:
      - **Address** indicates the MQS address of ROMA Connect. For details about how to obtain the MQS address, see [Viewing Details of an Instance](#). If you access the topic from a public network, use the public connection address. If you access the topic using the intra-VPC mode, use the intranet connection address.
      - **TopicName** indicates the name of the topic where the message is to be produced.
      - **../config/producer.properties** indicates the relative path of the configuration file.
    - b. Enter the message contents and send the messages to the topic.

```
>Message1  
>Message2  
>Message3
```

In the preceding information, **Message1**, **Message2**, and **Message3** indicate the actual message contents sent to the topic. Each row indicates a message.
    - c. To disconnect from the topic, press **Ctrl+C**.
  5. Consume messages from a topic.

 **NOTE**

When a consumer consumes messages from multiple partitions of a topic, messages in only one partition can be consumed at a time. Messages in multiple partitions are consumed in multiple times.

- a. Run the following command to create a connection with the topic for reading messages:  

```
./kafka-console-consumer.sh --bootstrap-server Address --topic TopicName --from-beginning --consumer.config ../config/consumer.properties
```

In the preceding command:
  - **Address** indicates the MQS address of ROMA Connect. For details about how to obtain the MQS address, see [Viewing Details of an Instance](#). If you access the topic from a public network, use the public connection address. If you access the topic using the intra-VPC mode, use the intranet connection address.

- **TopicName** indicates the name of the topic where the message is to be consumed.
  - **../config/consumer.properties** indicates the relative path of the configuration file.
- b. After the command is executed, the system is continuously connected to the topic and reads messages. To disconnect from the topic, press **Ctrl+C**.

## Without Using SASL Authentication

If SASL\_SSL access is not enabled for the ROMA Connect instance, the client does not need to load the certificate. Messages produced and consumed by the topic are not encrypted. The following uses Linux as an example.

1. Decompress the Kafka command-line tool.

Access the directory where the tool package is stored and run the following command to decompress the package:

```
tar -zxf kafka_tar
```

In the preceding command, **kafka\_tar** indicates the name of the Kafka command-line tool package.

2. Access the **/bin** directory of the Kafka command-line tool.

In the Windows operating system, you need to access the **/bin/windows** directory.

3. Produce a message to the topic.

- a. Run the following command to create a connection with the topic for producing messages:

```
./kafka-console-producer.sh --broker-list Address --topic TopicName
```

In the preceding command:

- **Address** indicates the MQS address of ROMA Connect. For details about how to obtain the MQS address, see [Viewing Details of an Instance](#). If you access the topic from a public network, use the public connection address. If you access the topic using the intra-VPC mode, use the intranet connection address.
  - **TopicName** indicates the name of the topic where the message is to be produced.
- b. Enter the message contents and send the messages to the topic.

```
>Message1  
>Message2  
>Message3
```

In the preceding information, **Message1**, **Message2**, and **Message3** indicate the actual message contents sent to the topic. Each row indicates a message.

- c. To disconnect from the topic, press **Ctrl+C**.

4. Consume messages from a topic.

### NOTE

When a consumer consumes messages from multiple partitions of a topic, messages in only one partition can be consumed at a time. Messages in multiple partitions are consumed in multiple times.

- a. Run the following command to create a connection with the topic for reading messages:  

```
./kafka-console-consumer.sh --bootstrap-server Address --topic TopicName --from-beginning
```

In the preceding command:

  - **Address** indicates the MQS address of ROMA Connect. For details about how to obtain the MQS address, see [Viewing Details of an Instance](#). If you access the topic from a public network, use the public connection address. If you access the topic using the intra-VPC mode, use the intranet connection address.
  - **TopicName** indicates the name of the topic where the message is to be produced.
- b. After the command is executed, the system is continuously connected to the topic and reads messages. To disconnect from the topic, press **Ctrl+C**.

## 8.5 Topic Management

### 8.5.1 Querying Topic Messages

#### Overview

ROMA Connect provides a visualized message query function, which allows you to view the message data stored in topics on the console and view the message body more intuitively and conveniently.

Messages of only one topic can be queried at a time.

#### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Message Queue Service > Message Query**.
3. In the upper right corner of the page, select the topic whose messages are to be queried. The messages stored in the topic are automatically displayed.  
You can also click **Advanced Search**, set search criteria, and click **Search** to search for messages.
  - **Search By**: method used for message query.
    - **Creation time**: Messages are queried by creation time.
    - **Offset**: Messages are queried by the location where the message is recorded in a partition.
  - **Topic Name**: name of the topic whose messages are to be queried.
  - **Partition**: ID of the partition in which messages are to be queried. The partition ID starts from 0. This parameter is mandatory if **Search By** is set to **Offset**.
  - **Tag**: Tags are carried when a client produces messages to a topic. Generally, tags are not used.

- **Message ID:** unique ID of a message.
- **Business Key:** business key carried in a message, which identifies the service to which the message belongs.

## 8.5.2 Importing and Exporting Topics

### Overview

ROMA Connect allows you to export topics as a file to the local directory or import a local topic file to ROMA Connect for batch migration of topics.

### Prerequisites

- Before importing a topic, ensure that the integration application to which the topic belongs has been created. Otherwise, [create an integration application](#) first.
- Before importing a topic, check whether a duplicate topic name exists in the instance to which the topic is to be imported. If there are duplicate topic names, the topic will fail to be imported.
- Ensure that the quota of topics meets the requirements before importing the topics.
- If the description of a topic contains a newline character, the newline character is escaped to `\n` in the exported CSV file of the topic. Therefore, if you import topics using such exported CSV files, manually modify the escape character `\n` back to a newline character.

### Importing Topics

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **Message Queue Service > Topic Management**. On the page displayed, click **Import**.
3. In the dialog box displayed, select a local topic file and import it.
4. After the import is successful, you can view the imported topics in the topic list.

### Exporting Topics

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane, choose **Message Queue Service > Topic Management**.
3. Export topics.
  - Exporting specified topics: Select the topics to be exported and choose **More > Export** to export a topic file to the local directory.
  - Exporting all topics: Click **Export All** to export a topic file to the local directory.

# 9 Device Integration Guide

---

- [Usage Introduction](#)
- [Creating a Product](#)
- [Registering a Device](#)
- [Connecting Devices to ROMA Connect](#)
- [Product Management](#)
- [Device Management](#)
- [Rule Engine](#)
- [Subscription Management](#)

## 9.1 Usage Introduction

### Function Description

Device Integration is a component of ROMA Connect, which uses the standard MQTT protocol to connect to devices and quickly manage devices on the cloud. ROMA Connect has the following advantages for device integration:

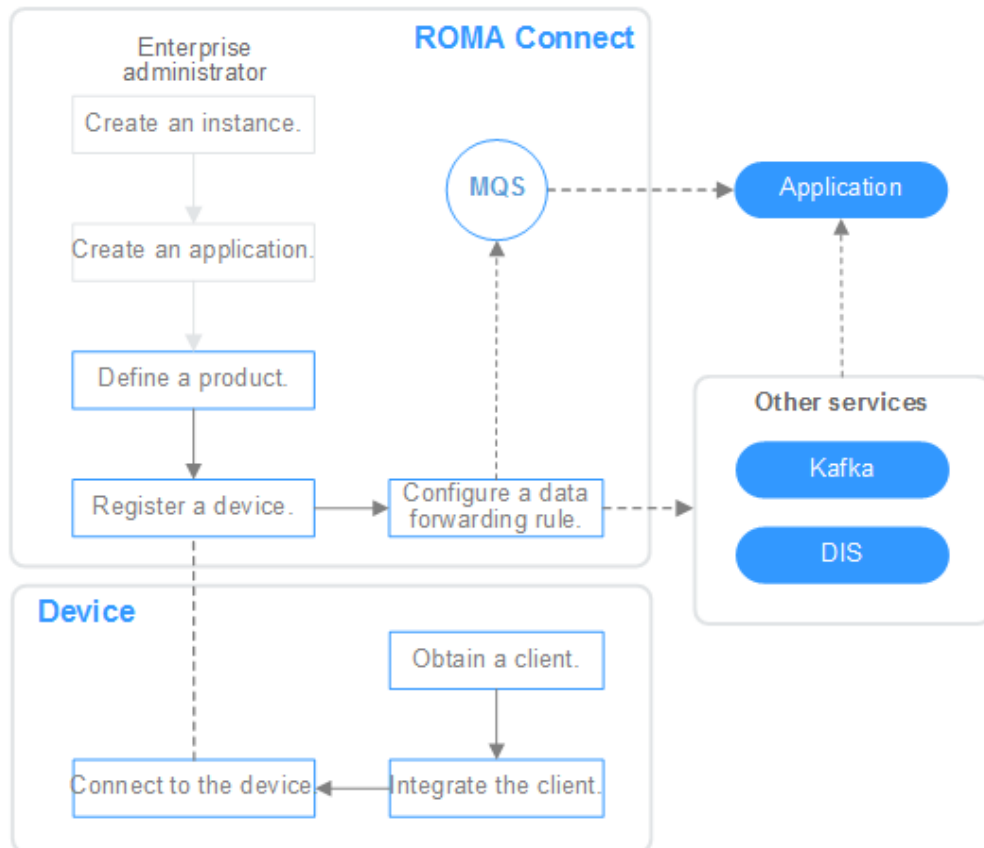
- **Standard MQTT protocols**  
The open-source standard MQTT SDK is used to connect devices to the cloud for message publishing and subscription.
- **Low-latency access for massive numbers of devices**  
LINK supports horizontal expansion of Broker and long connections of millions of devices.
- **Two-way synchronization between devices and applications**  
LINK implements two-way synchronization of configuration and status data between devices and applications.
- **Secure device information transmission**  
LINK provides authorization certification for devices and applications and bidirectional binding authorization for topics to ensure device security and

uniqueness. It provides TLS-based data transmission channels for secure message transmission.

## Process Flow

The following figure shows the process of using ROMA Connect for device integration.

**Figure 9-1** Process of using ROMA for device integration



1. You have **created an instance and an integration application.**
2. **Create a product.**  
A product is a device model. You can define a product to determine the function attributes of devices. Each device belongs to a product.
3. **Register a device.**  
After a device is registered with ROMA Connect, a unique ID and key are allocated to the device so that the device can connect to Device Integration.
4. **Configure a data forwarding rule.**  
ROMA Connect does not directly store data reported by devices. You must configure data forwarding rules to forward device data to other services for storage. Data can be forwarded to ROMA MQS or other services such as Kafka and DIS.
5. **Connect to devices.**  
After being integrated with an MQTT client, a device can connect to ROMA Connect for reporting data and delivering commands.

## 9.2 Creating a Product

### Overview

A product is a device model, which is a set of service attributes of devices of the same type. Each device belongs to a product and inherits all service attributes of the product.

Product information includes basic information and thing model. It describes what a product is, what functions the product can provide, and what services the product can provide for external systems.

- Basic information

Describes basic information about a device, including the manufacturer name, manufacturer ID, product type, product model, protocol type, and device type. The product model and manufacturer ID are used together to identify a product.

For example, the product model of a water meter is **NBLoTDevice**, the manufacturer name is **HZYB**, the manufacturer ID is **TestUtf8Manuld**, and the device type is **WaterMeter**.

- Thing model

Service capabilities of a device need to be defined. After the capabilities of a device are divided into multiple thing model services, define the attributes, commands, and command parameters of each thing model service.

Take a water meter as an example. It has multiple capabilities, such as reporting data about the water flow, alarms, power, and connections, and receiving various commands delivered by a server. When describing the capabilities of a water meter, you can divide the capabilities into five thing model services, and define the attributes reported by each thing model service and their supported commands, as listed in the following table.

**Table 9-1** Service description

| Thing Model Service | Description                                                                                                                                                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WaterMeterBasic     | Used to define parameters reported by the water meter, such as the water flow, temperature, and pressure. If these parameters need to be controlled or modified using commands, you also need to define parameters in the commands. |
| WaterMeterAlarm     | Used to define data reported by the water meter in various alarm scenarios. Commands need to be defined if necessary.                                                                                                               |
| Battery             | Used to define data including the voltage and current intensity of the water meter.                                                                                                                                                 |
| DeliverySchedule    | Used to define transmission rules for the water meter. Commands need to be defined if necessary.                                                                                                                                    |

| Thing Model Service | Description                                               |
|---------------------|-----------------------------------------------------------|
| Connectivity        | Used to define connection parameters for the water meter. |

 **NOTE**

The number of thing model services defined depends on the requirements. In the preceding example, the alarm thing model service can be split into the water pressure alarm service and traffic alarm service, or the alarm service can be integrated into the basic water meter service.

### Prerequisites

- Each product must belong to an integration application. Before creating a product, ensure that an integration application is available. Otherwise, [create an integration application](#) first.
- If you need to create a product by using a product template, [create a product template](#) first.

### Creating a Product

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Product Management**. On the page displayed, click **Create Product** in the upper right corner.
3. In the **Create Product** dialog box, set product parameters and click **OK**.

**Table 9-2** Parameters for creating a product

| Parameter               | Description                                                                                                                                                                                                                                                                                                                  |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product Type            | Select a product type. <ul style="list-style-type: none"> <li>• <b>Common</b>: indicates a directly connected device or a gateway subdevice.</li> <li>• <b>Gateway</b>: indicates a gateway product of the Edgeshell type.</li> </ul>                                                                                        |
| Protocol Type           | Select the protocol type used by the product. <ul style="list-style-type: none"> <li>• <b>MQTT</b>: a publish-subscribe network protocol that transports messages between devices.</li> <li>• <b>Modbus</b>: a serial communication protocol.</li> <li>• <b>OPC UA</b>: an Ethernet-based communication protocol.</li> </ul> |
| Integration Application | Select the integration application to which the product belongs.                                                                                                                                                                                                                                                             |



| Parameter         | Description                                                                                                                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product Template  | After a product template is selected, the created product has all services and attributes of the product template.<br>Only product templates in the <b>Enabled</b> state can be selected.                                                                                       |
| Product Name      | Enter a product name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                       |
| Manufacturer Name | Enter the manufacturer name of the device in use.                                                                                                                                                                                                                               |
| Manufacturer ID   | Enter the manufacturer ID of the device in use.                                                                                                                                                                                                                                 |
| Product Model     | Enter the product model of the device in use.<br>The product model and manufacturer ID collectively identify a product. The combination of product model and manufacturer ID must be unique.                                                                                    |
| Device Type       | Select the type of the device to be connected. You can select <b>Default Type</b> or add a device type that is added in <b>DEVICE_TYPE</b> on the <b>Data Dictionaries</b> tab page of the <b>Instances</b> page. For details, see <a href="#">Creating a Data Dictionary</a> . |
| Model Version     | Enter a custom model version of the device.                                                                                                                                                                                                                                     |
| Description       | Enter a brief description of the product.                                                                                                                                                                                                                                       |

### (Optional) Adding a Thing Model Service for a Product

1. On the ROMA Connect console, choose **LINK > Product Management** and click a product name to access the product details page.
2. Add thing model services for the product.
  - a. On the **Thing Model** tab page, click **Create**.
  - b. In the **Create Thing Model Service** dialog box, set thing model service parameters and click **OK**.

**Table 9-3** Parameters for creating a thing model service

| Parameter   | Description                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------------------|
| Name        | Enter a thing model service name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Status      | This parameter specifies whether to enable the thing model service. The thing model service is enabled by default.    |
| Description | Enter a brief description of the thing model service.                                                                 |

3. Add attributes for the thing model service.
  - a. On the **Thing Model** tab page, select the thing model service for which you want to add attributes.
  - b. On the **Attributes** tab page, click **Create**.
  - c. In the **Create Attribute** dialog box, set attribute parameters and click **OK**.

**Table 9-4** Parameters for creating an attribute


| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name        | Enter an attribute name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Data Type   | <p>Select the data type of the attribute.</p> <ul style="list-style-type: none"> <li>• <b>Int</b>: integer. If you select <b>Int</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, <b>Step</b>, and <b>Unit</b>.</li> <li>• <b>Number</b>: number. If you select <b>Number</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, <b>Step</b>, and <b>Unit</b>.</li> <li>• <b>String</b>: a character string. If you select <b>String</b>, you also need to specify <b>Data Length</b> and <b>Enumerated Value</b>.</li> <li>• <b>Bool</b>: Boolean. If you select <b>Bool</b>, you also need to specify <b>0</b> and <b>1</b>.</li> <li>• <b>Datetime</b>: date. If you select <b>Datetime</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>JsonObject</b>: JSON object. If you select <b>JsonObject</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>Array</b>: array.</li> </ul> <p>The following uses the <b>Int</b> type as an example:<br/>When defining the temperature attribute for thermometers, set <b>Type</b> to <b>Int</b>, <b>Min. Value</b> to <b>0</b>, <b>Max. Value</b> to <b>100</b>, <b>Step</b> to <b>2</b>, and <b>Unit</b> to <b>°C</b>. This indicates that the device reports the temperature value (for example, 0°C, 2°C, 4°C, 6°C and 8°C) each time the temperature changes by two degrees.</p> |
| Mandatory   | This parameter specifies whether the attribute must be reported by a device. This parameter is enabled by default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Description | Enter a brief description of the attribute.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Min. Value  | Enter the minimum value of the attribute. The value must be of the Int type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Max. Value  | Enter the maximum value of the attribute. The value must be of the Int type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| Parameter | Description                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------|
| Step      | Enter the step of the attribute. The value must be of the Int type.                                                                   |
| Unit      | Enter the unit of the attribute. The value cannot exceed 50 characters. For example, the unit of the temperature attribute can be °C. |

4. Add commands for the thing model service.
  - a. On the **Thing Model** tab page, select the thing model service for which you want to add commands.
  - b. On the **Commands** tab page, click **Create**.
  - c. In the **Create Command** dialog box, set command parameters and click **OK**.

**Table 9-5** Parameters for creating a command

| Parameter    | Description                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------------------|
| Command Name | Enter a command name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Description  | Enter a brief description of the command.                                                                 |

- d. Locate the created command in the command list and click  next to the command name to expand the command field list.  
Click **Delivery Command Fields** or **Response Command Fields** on the right to view the delivery command fields and response command fields of the command.
- e. Click **Create Field**.
- f. In the **Create Delivery Command Field** or **Create Response Command Field** dialog box, set field parameters and click **OK**.

**Table 9-6** Parameters for creating a field

| Parameter  | Description                                                                                             |
|------------|---------------------------------------------------------------------------------------------------------|
| Field Name | Enter a field name. It is recommended that you enter a name based on naming rules to facilitate search. |

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Type   | <p>Select the data type of the command field.</p> <ul style="list-style-type: none"> <li>• <b>Int</b>: integer. If you select <b>Int</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, <b>Step</b>, and <b>Unit</b>.</li> <li>• <b>Number</b>: number. If you select <b>Number</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, <b>Step</b>, and <b>Unit</b>.</li> <li>• <b>String</b>: a character string. If you select <b>String</b>, you also need to specify <b>Data Length</b> and <b>Enumerated Value</b>.</li> <li>• <b>Bool</b>: Boolean. If you select <b>Bool</b>, you also need to specify <b>0</b> and <b>1</b>.</li> <li>• <b>Datetime</b>: date. If you select <b>Datetime</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>JsonObject</b>: JSON object. If you select <b>JsonObject</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>Array</b>: array.</li> </ul> |
| Mandatory   | This parameter specifies whether the field must be carried in the delivery command. By default, the field must be carried.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Description | Enter a brief description of the command field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Min. Value  | Enter the minimum value of the delivery command field. The value must be of the Int type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Max. Value  | Enter the maximum value of the delivery command field. The value must be of the Int type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Step        | Enter the step of the delivery command field. The value must be of the Int type.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Unit        | Enter the unit of the delivery command field. The value cannot exceed 50 characters. For example, the unit of the voltage field can be V.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

### (Optional) Adding a Customized Topic for a Product

If the basic topics of a device cannot meet your requirements, you can customize a topic for the device. After the customized topic is added to a product, the topic is inherited by all devices under the product.

1. On the ROMA Connect console, choose **LINK > Product Management** and click a product name to access the product details page.
2. On the **Topic Classes** tab page, click **Create Topic**.
3. In the dialog box displayed, configure topic information and click **OK**.

**Table 9-7** Parameters for creating a topic

| Parameter            | Description                                                                                                                                                                                                                                                                                                              |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operation Permission | Select the topic permission granted for the device. <ul style="list-style-type: none"> <li>• <b>Publishing:</b> The device has the permission to publish messages into the topic.</li> <li>• <b>Subscription:</b> The device has the permission to subscribe to the topic and obtain messages from the topic.</li> </ul> |
| Topic Class Name     | Enter the custom field in the name of the custom topic. The version number, device ID, and customized fields form a custom topic name: /{Version number}/{Device ID}/{Custom field}.                                                                                                                                     |
| Version Number       | Enter the version number of the topic, for example, <b>V1.0</b> .                                                                                                                                                                                                                                                        |
| Description          | Enter the brief description of the custom topic.                                                                                                                                                                                                                                                                         |

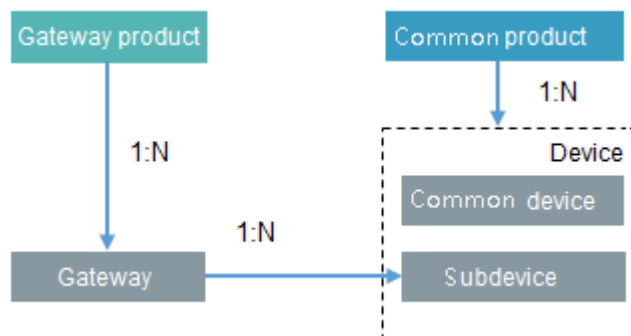
## 9.3 Registering a Device

### Overview

After a device is registered with ROMA Connect, a unique ID and password are allocated to the device in the system. The device uses the ID and password to access ROMA Connect.

A device that is directly connected to ROMA Connect is called common device, and a device that is connected to ROMA Connect through a gateway is called subdevice.

The following figure shows the relationship between devices and products. Both common devices and subdevices belong to common products. Gateways belong to gateway products. A gateway can connect to multiple subdevices.



### Prerequisites

- Each device must belong to an integration application. Before registering a device, ensure that an integration application is available. Otherwise, **create an integration application** first.

- Each device must belong to a product. Before registering a device, ensure that a product is available. Otherwise, [create a product](#) first.

## Creating a Device

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Device Management**. On the page displayed, click **Create Device** in the upper right corner.
3. In the **Create Device** dialog box, set device parameters and click **OK**.

**Table 9-8** Parameters for creating a device

| Parameter               | Description                                                                                                                                                                                                                              |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Integration Application | Select the integration application to which a device belongs.                                                                                                                                                                            |
| Associated Product      | Select the product to which the device belongs. The device inherits all thing model service capabilities defined for the product.<br>For common devices and subdevices, select a common product. For gateways, select a gateway product. |
| Device ID               | Enter the unique identifier of the device, such as the IMEI and MAC address. The device ID corresponds to the deviceId of a directly connected device or a subdevice under a gateway or corresponds to the gatewayId of a gateway.       |
| Device Name             | Enter a device name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                 |
| Password                | Enter the password for connecting the device to ROMA Connect. The password is user-defined.                                                                                                                                              |
| Confirm Password        | Enter the confirm password, which must be the same as the value of <b>Password</b> .                                                                                                                                                     |
| Status                  | This parameter specifies whether to enable the device. The device can connect to ROMA Connect only after being enabled. By default, the device is enabled.                                                                               |
| Device Tag              | Add tags for the device to facilitate search.                                                                                                                                                                                            |
| Description             | Enter a brief description of the device.                                                                                                                                                                                                 |

## (Optional) Adding a Custom Topic for a Device

If the basic topics of a device cannot meet your requirements, you can customize a topic for the device.

1. On the ROMA Connect console, choose **LINK > Device Management** and click a device name to access the device details page.

2. On the **Topics** tab page, select **Custom Topics** and click **Create** on the right.
3. In the dialog box displayed, set custom topic parameters and click **OK**.

**Table 9-9** Parameters for creating a custom topic

| Parameter   | Description                                                                                                                                                                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name        | Enter a custom topic name.                                                                                                                                                                                                                                                                             |
| Permission  | Select the topic permission granted for the device. <ul style="list-style-type: none"><li>• <b>Publish:</b> The device has the permission to publish messages into the topic.</li><li>• <b>Subscribe:</b> The device has the permission to subscribe to the topic and read messages from it.</li></ul> |
| Description | Enter a brief description of the custom topic.                                                                                                                                                                                                                                                         |

## 9.4 Connecting Devices to ROMA Connect

### Overview

A device can access ROMA Connect by integrating with the MQTT.fx client.

After a device is connected to ROMA Connect, it can report data to ROMA Connect. You can also use ROMA Connect to deliver control commands to the device.

ROMA Connect does not directly store data reported by devices. You must [configure data forwarding rules](#) to forward device data to other services for storage.

### Connecting to Devices

A device can access ROMA Connect by using the open-source Eclipse Paho MQTT Client. You can also use the [MQTT.fx client](#) to verify device access.

1. Obtain the MQTT client.  
Obtain the [Eclipse Paho MQTT Client](#) based on your programming language.
2. Obtain device access information.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
  - b. In the navigation tree on the left, choose **LINK > Device Management**.
  - c. Obtain device connection information.
    - MQTT/MQTTS connection address: Obtain the access address of the device from the upper part of the **Device Management** page and then download the SSL certificate. If MQTT is used for access, obtain the MQTT connection address. If MQTTS is used for access, obtain the MQTTS connection address.

- Client ID/Username/Password: Locate the corresponding device on the **Devices** tab page and obtain the client ID, username, and password of the device.
- d. Obtain topic information of the device.

Click the device name to go to the device details page. On the **Topics** tab page, obtain the topic information about the messages reported (published) and received (subscribed) by the device.

After the device is created, the system automatically generates five basic topics. For details about the topics, see [Table 9-10](#). For details about how to use a topic, see "MQTT Topic Specifications" in the *ROMA Connect Developer Guide*.

**Table 9-10** Basic topic description

| Topic Name                              | Topic Class | Description                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {Product ID}/out/{Device ID}            | Publish     | Used by devices to publish messages. Messages are processed based on the rule engine configuration.                                                                                                                                                                                                                                       |
| {Product ID}/in/{Device ID}             | Subscribe   | Used by devices to subscribe to messages. Messages are processed based on the rule engine configuration.                                                                                                                                                                                                                                  |
| /v1/devices/{Device ID}/datas           | Publish     | Used by devices to report messages complying with the MQTT-based communication protocol for IoT in power distribution. Messages are filtered based on the protocol and then processed based on the rule engine configuration. When the device shadow function is enabled, the data reported by this topic will be recorded in the shadow. |
| /v1/devices/{Device ID}/command         | Subscribe   | Used by devices to subscribe to commands delivered by the IoT platform. Commands delivered by the IoT platform are published to this topic by default.                                                                                                                                                                                    |
| /v1/devices/{Device ID}/commandResponse | Publish     | Used by devices to publish responses. After a device receives a command from the IoT platform, it responds to the command through this topic.                                                                                                                                                                                             |

3. Integrate the device with the MQTT client.



Develop and integrate the device with Eclipse Paho MQTT Client, and write the device access information during the integration. For details, see [Device Integration Development](#).

4. Connect the device to ROMA Connect.

After the integration development is complete, power on the device to take it online and connect it to ROMA Connect.

### Commissioning a Device

- Reporting data to ROMA Connect  
By default, ROMA Connect has subscribed to topics of the publish type. After a device sends a message to such a topic, the topic can report data to ROMA Connect.
- Receiving commands from ROMA Connect  
After ROMA Connect sends a control command to the command delivery topic, the topic delivers the command to the device for device control.

## 9.5 Product Management

### 9.5.1 Viewing Products


#### Overview

After a product is created, you can view and reset the product certificate, and import and export the product.

#### Functions

After a product is created, you can view and edit the product as required by referring to the following table.

**Table 9-11** Function description

| Function                      | Description                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Viewing a product certificate | <p>A product certificate can be used as a credential for accessing the device to which a product belongs. All devices under a product can use one product certificate for authentication.</p> <p>Click  in the <b>Password</b> column to view the product certificate.</p> |

| Function                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Importing products                  | <p>If a file already exists on the local host, you can click <b>Import</b> on the <b>Products</b> tab page to import the file that contains product information.</p> <p>To obtain a file template, create a product on the console, select the product, and click <b>Export</b> to export a CSV file. Then, you can modify the CSV file based on the product to be imported.</p> <p><b>NOTE</b><br/>The imported product information does not contain attributes and commands of the thing model and thing model service. You can manually create them after the import.</p> |
| Exporting specified or all products | <p>Export information about one or more products to a CSV file.</p> <p>To export one or more products, select the products to be exported and click <b>Export</b>. To export all products, click <b>Export All</b>.</p> <p><b>NOTE</b><br/>The exported product information does not contain the thing model and the attributes and commands of the thing model service.</p>                                                                                                                                                                                                 |
| Resetting a password                | <p>To change the password (product certificate), click <b>Reset Password</b> in the <b>Operation</b> column of the target product.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Editing a product                   | <p>Click <b>Edit</b> in the <b>Operation</b> column of the target product. <b>Product Type</b>, <b>Protocol Type</b>, and <b>Application</b> cannot be modified.</p>                                                                                                                                                                                                                                                                                                                                                                                                         |
| Deleting one or more products       | <p>Delete a created product.</p> <p>To delete one or more products, select the required products and click <b>Delete</b>. You can also click <b>Delete</b> in the <b>Operation</b> column of a specific product to delete it.</p>                                                                                                                                                                                                                                                                                                                                            |

## 9.5.2 Importing and Exporting Products

### Overview

ROMA Connect supports product import and export. You can export products as a file to the local host or import a local product file to ROMA Connect to migrate product information in batches.

### Prerequisites

- Before importing a product, ensure that the integration application to which the product belongs has been created. Otherwise, [create the integration application](#) in advance.
- Before importing a product, check whether duplicate product names exist in the instance to which the product is to be imported. If duplicate product names exist, the product fails to be imported.

- Ensure that the quota of products meets the requirements before importing the products.

## Importing Products

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Product Management**. On the **Products** tab page, click **Import**.
3. In the dialog box displayed, select a local product file and import it.
4. After the import is successful, you can view the imported products in the product list.

## Exporting Products

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **LINK > Product Management**.
3. Export specified or all products.
  - Exporting specified products: Select the products to be exported and click **Export** to export the product file to the local host.
  - Exporting all products: Click **Export All** to export the product file to the local host.

## 9.5.3 Creating a Product Template

### Overview

You can save service capabilities of a product as a product template. When creating a product, you can select and inherit the service capabilities of the product template to quickly create the product.

### Creating a Product Template

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Product Management**. On the page displayed, click **Create Product Template** in the upper right corner.
3. In the **Create Product Template** dialog box, set product template parameters and click **OK**.

**Table 9-12** Parameters for creating a product template

| Parameter | Description                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------|
| Name      | Enter a product template name. It is recommended that you enter a name based on naming rules to facilitate search. |

| Parameter   | Description                                                                                                                                                                                                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description | Enter the descriptive information of the product template.                                                                                                                                                                                                                                  |
| Status      | This parameter specifies whether to enable the product template. The product template is enabled by default. Services and attributes cannot be added to a product template in the <b>Disabled</b> state. In addition, such a product template cannot be selected when a product is created. |

### Adding a Thing Model Service to a Product Template

1. Log in to the ROMA Connect console, choose **LINK > Product Management**, and click the **Product Templates** tab.
2. Click the product template name to go to the product template details page.
3. Add a thing model service to a product template.
  - a. On the **Thing Model** tab page, click **Create**.
  - b. In the **Create Thing Model Service** dialog box, set thing model service parameters and click **OK**.

**Table 9-13** Parameters for creating a thing model service

| Parameter   | Description                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------------------|
| Name        | Enter a thing model service name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Status      | This parameter specifies whether to enable the thing model service. The thing model service is enabled by default.    |
| Description | Enter a brief description of the thing model service.                                                                 |

4. Add attributes for the thing model service.
  - a. On the **Thing Model** tab page, select the thing model service for which you want to add attributes.
  - b. On the **Attributes** tab page on the right, click **Create**.
  - c. In the **Create Attribute** dialog box, set attribute parameters and click **OK**.


**Table 9-14** Parameters for creating an attribute

| Parameter      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attribute Name | Enter an attribute name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Data Type      | <p>Select the data type of the attribute.</p> <ul style="list-style-type: none"> <li>• <b>Int</b>: integer. If you select <b>Int</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, and <b>Unit</b>.</li> <li>• <b>Number</b>: number. If you select <b>Number</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, and <b>Unit</b>.</li> <li>• <b>String</b>: a character string. If you select <b>String</b>, you also need to specify <b>Data Length</b> and <b>Enumerated Value</b>.</li> <li>• <b>Bool</b>: Boolean. If you select <b>Bool</b>, you also need to specify <b>0</b> and <b>1</b>.</li> <li>• <b>Datetime</b>: date. If you select <b>Datetime</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>JsonObject</b>: JSON object. If you select <b>JsonObject</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>Array</b>: array.</li> </ul> <p>The following uses the <b>Int</b> type as an example:<br/>For example, when defining the temperature attribute for thermometers, set <b>Type</b> to <b>Int</b>, <b>Min. Value</b> to <b>0</b>, <b>Max. Value</b> to <b>100</b>, <b>Step</b> to <b>2</b>, and <b>Unit</b> to <b>°C</b>. This indicates that the device reports the temperature value (for example, 0°C, 2°C, 4°C, 6°C and 8°C) each time the temperature changes by two degrees.</p> |
| Mandatory      | This parameter specifies whether a device must report this attribute. This parameter is enabled by default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Description    | Enter a description of the attribute.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

5. Add commands for the thing model service.
  - a. On the **Thing Model** tab page, select the thing model service for which you want to add commands.
  - b. On the **Commands** tab page, click **Create**.
  - c. In the **Create Command** dialog box, set command parameters and click **OK**.

**Table 9-15** Parameters for creating a command

| Parameter    | Description                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------------------|
| Command Name | Enter a command name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Description  | Enter a brief description of the command.                                                                 |

- d. Locate the created command in the command list and click  next to the command name to expand the command field list.

Click **Delivery Command Fields** or **Response Command Fields** on the right to view the delivery command fields and response command fields of the command.

- e. Click **Create Field**.
- f. In the **Create Field** dialog box, set command field parameters and click **OK**.

**Table 9-16** Parameters for creating a command field

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Field Name  | Enter a field name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Data Type   | Select the data type of the command field. <ul style="list-style-type: none"> <li>• <b>Int</b>: integer. If you select <b>Int</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, and <b>Unit</b>.</li> <li>• <b>Number</b>: number. If you select <b>Number</b>, you also need to specify <b>Min. Value</b>, <b>Max. Value</b>, and <b>Unit</b>.</li> <li>• <b>String</b>: a character string. If you select <b>String</b>, you also need to specify <b>Data Length</b> and <b>Enumerated Value</b>.</li> <li>• <b>Bool</b>: Boolean. If you select <b>Bool</b>, you also need to specify <b>0</b> and <b>1</b>.</li> <li>• <b>Datetime</b>: date. If you select <b>Datetime</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>JsonObject</b>: JSON object. If you select <b>JsonObject</b>, you also need to specify <b>Data Length</b>.</li> <li>• <b>Array</b>: array.</li> </ul> |
| Mandatory   | This parameter specifies whether the field must be carried in the delivery command. By default, the field must be carried.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Description | Enter the descriptive information of the command field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

## 9.6 Device Management

### 9.6.1 Viewing Devices


#### Overview

After a device is created, you can view the device certificate and details, force the device to go offline, and edit device information.

#### Functions

After a device is created, you can view or operate the device by referring to the following table.

**Table 9-17** Functions

| Function                     | Description                                                                                                                                                                                                                                                  |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Viewing a device certificate | A device certificate can be used as a credential for accessing a device.<br><br>You can click  on the right of the <b>Password</b> column to view the device certificate. |
| Editing a device             | Modify information of a created device, except the product name and device ID.                                                                                                                                                                               |
| More > Force Offline         | Forcibly bring online devices offline.<br><b>NOTE</b><br>This function is applicable only to the online devices that use the MQTT protocol.                                                                                                                  |

| Function                                                                                                                                                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>More &gt; Deliver Command</p> <p><b>NOTE</b><br/>Commands can be delivered only when the protocol type of the product to which the device belongs is MQTT.</p> | <p>LINK can effectively manage devices. The product thing model of a device defines the commands that LINK can deliver to the device. An application can deliver commands to a single device through LINK to remotely control the device. The command delivery configuration is as follows:</p> <ul style="list-style-type: none"> <li>• Select a thing model service that is created in the product to which the device belongs. If no thing model service is available, click <b>Create</b> on the right to create one.</li> <li>• Determine whether to enable <b>Synchronous Communication</b>. If this parameter is enabled, a success response is returned after the entire process is complete or a failure response is returned when the timeout period expires. If this parameter is disabled, LINK returns a response immediately after the command is delivered, regardless of the subsequent process. If the device is offline when the command is delivered, a success response is also returned. After the device goes online, LINK forwards the cached command to the device.</li> <li>• Select a command under the thing model service of the product.</li> <li>• (Optional) Set the delivered command field. This field is displayed based on the field configured in the command. If the field is not configured in the command, it is not displayed. For example, if the delivered field is temperature, a text box is provided to enter the temperature value, such as 37°C.</li> </ul> <p><b>NOTE</b><br/>An application can also call open northbound APIs of LINK to deliver commands to a single device or a batch of devices.</p> |
| More > Reset Password                                                                                                                                             | To change the password (that is, device certificate), click <b>Reset Password</b> in the <b>Operation</b> column of the target device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| More > Delete                                                                                                                                                     | Delete a created device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

The content on the device details page varies depending on the product protocol type selected during device creation. For details, see [Table 9-18](#).

**Table 9-18** Description of the device details page

| Proto col Type | Tab Page          | Description                                                                                     |
|----------------|-------------------|-------------------------------------------------------------------------------------------------|
| MQTT           | Basic Information | You can view basic information, including the device ID, online status, and connection address. |



| Protocol Type | Tab Page          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | Topics            | <p>The system automatically generates default topics, that is, basic topics, based on the product ID and device ID. You can also customize topics as required.</p> <ol style="list-style-type: none"> <li>1. Click a device name to access the device details page. Click the <b>Topics</b> tab page and then select <b>Custom Topics</b>.</li> <li>2. Click <b>Create</b> and configure topic information. <ul style="list-style-type: none"> <li>• Enter a topic name.</li> <li>• Select the topic permission. The value can be <b>PUB</b> or <b>SUB</b>.</li> <li>• (Optional) Enter a description of the topic.</li> </ul> </li> <li>3. Click <b>OK</b>.</li> </ol> |
|               | Device Shadows    | <p>A device shadow is used to store the device status reported by devices and the expected status of applications. You can view the information in a list or a JSON file.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| ModBus        | Basic Information | <p>You can view the basic information, including the device ID, online status, connection address, edge connection configuration, and downloaded EdgeShell.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|               | Topics            | <p>The system automatically generates default topics, that is, basic topics, based on the product ID and device ID. You can also customize topics as required. The procedure is the same as that when MQTT is used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|               | Edge Details      | <p>You can view the resource status of the host where the device is located, whether the device is online, and the plug-in status.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

| Proto<br>col<br>Type | Tab Page                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | Modbus<br>Configuration | <p>Create a Modbus connection point and policy as follows:</p> <ol style="list-style-type: none"> <li>1. Click the device name to go to the device details page. Click the <b>Modbus Configuration</b> tab.</li> <li>2. Click <b>Create Connection Point</b> and configure connection point information. <ul style="list-style-type: none"> <li>• Specify the connection point name in <b>Name</b>.</li> <li>• Select the connection point type. <b>TCP/IP</b> and <b>RS232</b> are supported.</li> <li>• Enter the timeout interval of a configuration request, in milliseconds.</li> <li>• Enter an IP address of the connection point.</li> <li>• Enter an IP port of the connection point.</li> </ul> </li> <li>3. Click <b>OK</b>.</li> <li>4. After the connection point is created, click the drop-down list button and click <b>Create Policy</b>. <ul style="list-style-type: none"> <li>• Enter a policy name.</li> <li>• Enter a value greater than 200 as the collection interval. The maximum polling interval is 86400000 (one day), in the unit of ms.</li> <li>• Enter a value ranging from 1 to 125 to as the slave site ID.</li> <li>• Enter a value ranging from 0 to 1000000 as the start address.</li> <li>• Set the length of the data to be read. The value ranges from 1 to 1000.</li> <li>• Enter a function code. Currently, only <b>01</b>, <b>02</b>, <b>03</b>, and <b>04</b> are supported.</li> <li>• Click <b>Add Point</b>, configure the point mapping, and enter the attribute name of the point to be mapped. For example, if the attribute of collection point 0 is set to temperature, the 0-bit register value (temperature) is reported. This parameter is optional.</li> </ul> </li> <li>5. Click <b>OK</b>.</li> </ol> <p>After the policy is created, connect to a device and click <b>Deliver Policy</b> to deliver the policy to the device.</p> |
|                      | Plug-in<br>Deployment   | <p>Before using this function, you need to install the EdgeShell client and connect it to LINK.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| Protocol Type | Tab Page          | Description                                                                                                                                                                                                      |
|---------------|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OpcUA         | Basic Information | You can view basic information, including the device ID, online status, connection address, and downloaded EdgeShell.                                                                                            |
|               | Topics            | The system automatically generates default topics, that is, basic topics, based on the product ID and device ID. You can also customize topics as required. The procedure is the same as that when MQTT is used. |
|               | Edge Details      | You can view the resource status of the host where the device is located, whether the device is online, and the plug-in status.                                                                                  |

| Protocol Type | Tab Page             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | OPC UA Configuration | <p>Create an OPC UA connection point and policy as follows:</p> <ol style="list-style-type: none"> <li>1. Click the device name to access the device details page. Click the <b>OPC UA Configuration</b> tab.</li> <li>2. Click <b>Create Connection Point</b> and configure connection point information. <ul style="list-style-type: none"> <li>• Specify the connection point name in <b>Name</b>.</li> <li>• Enter the TCP connection address, which must start with <b>opc.tcp:</b> and can contain only letters, digits, colons (:), and slashes (/). Example: <code>opc.tcp://127.0.0.1:34561/DataTransferServer</code></li> <li>• Enter a value greater than 200 as the collection interval. The maximum polling interval is 86400000 (one day), in the unit of ms.</li> <li>• Determine whether to enable regular collection. If this parameter is set to <b>Yes</b>, the collection start time is a multiple of the collection interval.</li> <li>• Set a security policy.</li> <li>• Select the security mode. If you select <b>User Authentication</b>, you also need to specify <b>Username</b> and <b>Password</b>.</li> <li>• Enter the timeout interval of a request, in the unit of ms.</li> <li>• Select a topic for the device to publish messages.</li> </ul> </li> <li>3. Click <b>OK</b>.</li> <li>4. After the connection point is created, click the drop-down list button and click <b>Create Policy</b>. <ul style="list-style-type: none"> <li>• Enter a policy name.</li> <li>• Enter a value greater than 200 as the collection interval. The maximum polling interval is 86400000 (one day), in the unit of ms.</li> <li>• Enter a value ranging from 1 to 125 to as the slave site ID.</li> <li>• Enter a value ranging from 0 to 1000000 as the start address.</li> <li>• Set the length of the data to be read. The value ranges from 1 to 1000.</li> <li>• Enter a function code. Currently, only <b>01</b>, <b>02</b>, <b>03</b>, and <b>04</b> are supported.</li> <li>• Click <b>Add Point</b>, configure the point mapping, and enter the attribute name of the point to be mapped. For example, if the attribute of</li> </ul> </li> </ol> |

| Protocol Type | Tab Page           | Description                                                                                                                                                                                                                                                                  |
|---------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                    | <p>collection point 0 is set to temperature, the 0-bit register value (temperature) is reported. This parameter is optional.</p> <p>5. Click <b>OK</b>.</p> <p>After the policy is created, connect to the device and click <b>Deliver Policy</b> to deliver the policy.</p> |
|               | Plug-in Deployment | Before using this function, you need to install the EdgeShell client and connect it to LINK.                                                                                                                                                                                 |

## 9.6.2 Importing and Exporting Devices

### Overview

ROMA Connect supports device import and export. You can export device as a file to the local host or import a local product file to ROMA Connect to migrate device information in batches.

### Prerequisites

- Before importing a device, ensure that the integration application to which the device belongs has been created. Otherwise, [create the integration application](#) in advance.
- Before importing a device, ensure that the product to which the device belongs has been created. Otherwise, [create the product](#) in advance.
- Before importing a device, check whether a duplicate device name exists in the instance to which the device is to be imported. If there are duplicate device names, the device will fail to be imported.
- Ensure that the quota of devices meets the requirements before importing the devices.

### Importing Devices

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Device Management**. On the **Devices** tab page, click **Import Device**.
3. In the dialog box displayed, select a local device file and import it.
4. After the import is successful, you can view the imported devices in the device list.

### Exporting Devices

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.

2. In the navigation tree on the left, choose **LINK > Device Management**.
3. Export devices.
  - Exporting specified devices: Select the devices to be exported and click **Export** to export a device file to a local directory.
  - Exporting all devices: Click **Export All Devices** to export a device file to a local directory.

## 9.6.3 Creating a Device Group

### Overview

If there are a large number of devices, you can manage these devices by creating groups, which helps you quickly search for you desired devices.

### Prerequisites

Each device group must belong to an integration application. Before creating a device group, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Creating a Root Group

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Device Management**. On the **Device Groups** tab page, click **Create Root Group**.
3. In the **Create Root Group** dialog box, set group parameters and click **OK**.

**Table 9-19** Parameters for creating a root group

| Parameter         | Description                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------|
| Group Name        | Enter a device group name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Group Description | Enter a brief description of the device group.                                                                 |
| Application Name  | Select the integration application to which the device group belongs.                                          |

### Creating a Subgroup

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Device Management**. On the **Device Groups** tab page, click a device group for which you want to create a subgroup.

The created subgroup and the devices to be added to the subgroup must belong to the same application.

3. Click + on the right of the device group name.
4. In the **Create Group** dialog box, set subgroup parameters and click **OK**.

**Table 9-20** Parameters for creating a subgroup

| Parameter         | Description                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------|
| Parent Group      | Parent group to which a subgroup belongs.                                                                  |
| Group Name        | Enter a subgroup name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Group Description | Enter a brief description of the device group.                                                             |

### Adding a Device to a Device Group

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Device Management**. On the **Device Groups** tab page, click a device group for which you want to add a device.
3. Click **Add Device to Group**.
4. In the **Add Device to Group** dialog box, select the devices to be added and click **OK**.

## 9.6.4 Appendix: EdgeShell Access Description

EdgeShell is used to collect data from edge devices and send the data to the edge gateway program of ROMA Connect. Currently, EdgeShell supports the access of devices using the Modbus and OPC UA protocols. The procedure for using EdgeShell is as follows:

1. Create a product.  
On the **Product Management** page, click **Create Product**, set **Product Type** to **Common**, and set **Protocol Type** to **ModBus** or **OPC UA**.

## Create Product

Product Type ?  Common  Gateway

Protocol Type MQTT ModBus **OpcUA**

\* Application  ✕ 🔍

Product Template  ▾

\* Product Name

Enter a string of 1 to 64 characters. Only letters, digits, underscores (\_), and hyphens (-) are allowed.

2. Create a device

On the **Device Management** page, click **Create Device** and select the product created in 1.

Create Device ✕

---

\* Application Name  ✕

\* Associated Product  ▾

\* Device ID

Physical Identifier of a device, such as the IMEI and MAC address.

\* Device Name

3. Download the EdgeShell installation package.

Click the device name to access the device details page. On the **Basic Information** tab page, click **Download Installation Package** next to EdgeShell to download the EdgeShell installation package.

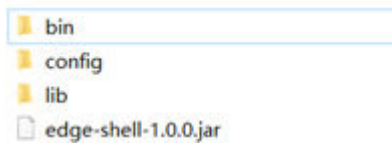


| Basic Information       | Topics                                        | Edge Details | OPC UA Configuration | Plug-in Deployment |
|-------------------------|-----------------------------------------------|--------------|----------------------|--------------------|
| Device ID               | qqtwat010203                                  |              |                      |                    |
| Integration Application | app-market01                                  |              |                      |                    |
| Creator                 | [REDACTED]                                    |              |                      |                    |
| Connection Address      | [REDACTED]                                    |              |                      |                    |
| Description             | --                                            |              |                      |                    |
| EdgeShell               | <a href="#">Download Installation Package</a> |              |                      |                    |

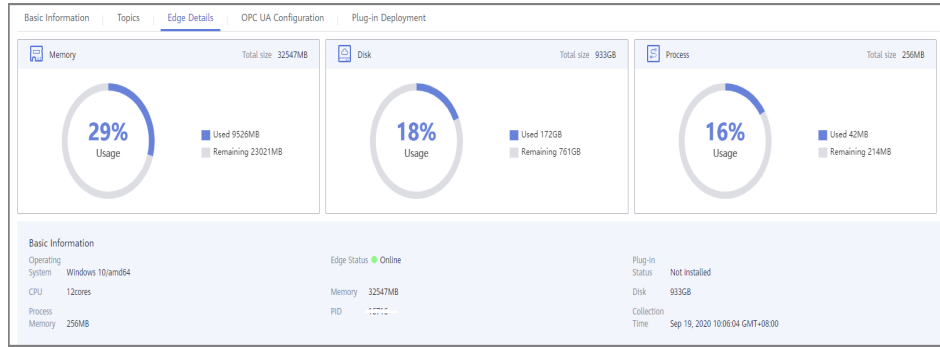
4. Start EdgeShell.

EdgeShell can run in both the Windows and Linux environments. The startup script varies according to the environment.

- a. EdgeShell depends on the Java running environment. Before running EdgeShell, ensure that the Java running environment has been installed and the JAVA\_HOME environment variable has been configured.
- b. Decompress the downloaded **edge-shell.zip** package. The following figure shows the directory structure obtained after the decompression.



- bin: startup and stop scripts
  - config: configuration parameters
  - lib: dependent jar package
  - edge-shell-1.0.0.jar: main program
- c. On the **Basic Information** tab page of the device details page, copy **SSL Connection Address (IPv4)** to the **mqtt.properties** file in the **config** directory.
  - d. Run the startup script.
    - In Windows, run **bin/windows/start.bat**.
    - In Linux, run **bin/linux/start.sh**.
  - e. After the device is started, the EdgeShell is connected to LINK. In this case, the device is displayed as online, and data is displayed in the **Edge Details** tab page of the device details page.



**NOTE**

After the script is started, the **log** directory is generated, which records startup and run logs. The **edge-shell-error.log** and **edge-shell-info.log** files exist in the directory. If the device is not connected after the startup, you can view the error logs to quickly locate the fault.

On Windows, if the log directory is not generated after you click **Start**, modify the following statement in the **start.bat** file.

Before the change:

```
start edge-shell %JAVA_OPT% -jar edge-shell-1.0.0.jar
```

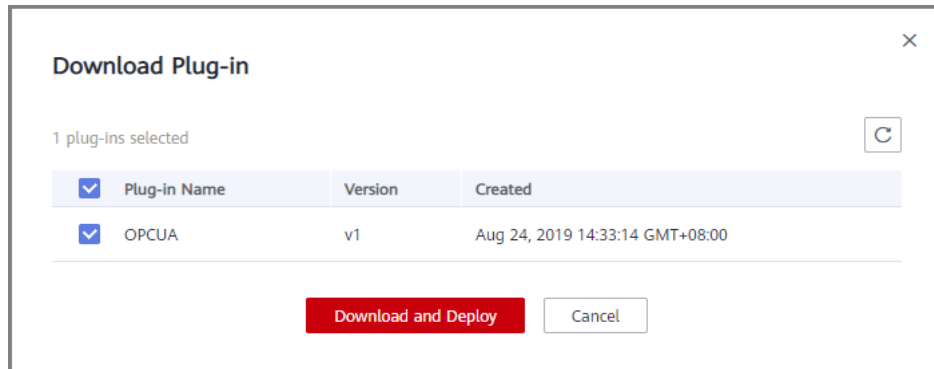
pause

After the change:

```
java %JAVA_OPT% -jar edge-shell-1.0.0.jar
```

- Download the plug-in.

On the **Plug-in Deployment** page, click **Download Plug-in**. In the displayed dialog box, click **Download and Deploy**.



After the plug-in is installed, click **Start** in the **Operation** column to start the plug-in.

| Plug-in Name | Version | Status     | Deployment Time                 | Operation             |
|--------------|---------|------------|---------------------------------|-----------------------|
| OPCUA        | v1      | Downloaded | Sep 24, 2020 15:16:13 GMT+08:00 | Start   Stop   Delete |

- Configure the Modbus/Opcua/connection point and collection policy.

- Modbus:

- If the created device is a Modbus device, **Modbus Configuration** is displayed on the device details page. The Modbus plug-in supports TCP/IP and serial port connection modes. The configuration is as follows:

- To create a Modbus connection point with the TCP/IP type, set the parameters, as shown in the following figure.

The screenshot shows a dialog box titled "Create Modbus Connection Point" with a close button (X) in the top right corner. The fields are as follows:

- Name: modbus\_connect-7mnn
- Connection Type: TCP/IP (selected), RS232
- Timeout Interval (ms): 300
- IP Address: 127 . 0 . 0 . 1
- IP Port: 502

Buttons: OK (red), Cancel (white).

- To create a Modbus connection point with the serial type, set the parameters, as shown in the following figure.

The screenshot shows a dialog box titled "Create Modbus Connection Point" with a close button (X) in the top right corner. The fields are as follows:

- Name: modbus\_connect-pbw3
- Connection Type: TCP/IP, RS232 (selected)
- Timeout Interval (ms): 300
- Serial Port (?): COM3
- Baud Rate: 9600
- Transmission Mode: RTU (selected), ASCII
- Data Bit: 8 (selected), 7
- Parity Bit: No parity (selected), Odd parity, Even parity
- Stop Bit: 1 (selected), 2

Buttons: OK (red), Cancel (white).

- ii. Create a collection policy under the created connection point.

The screenshot shows a dialog box titled "Create Modbus Policy" with a close button (X) in the top right corner. The fields are as follows:

- Associated Connection Point: modbus\_connect-dfqh
- Policy Name: strategy-k5bk
- Collection Interval (ms) (?): 5000
- Slave Site ID (?): 1
- Start Address (?): 5
- Read Length (?): Enter the read length.
- Function Code (?): 03

Point Mapping (?): Edit with JSON strings

| Register Point | Attribute Name | Opera... |
|----------------|----------------|----------|
| + Add Point    |                |          |

Buttons: OK (red), Cancel (white).

- OPC UA:
  - i. Create an OPC UA connection point. Currently, **Security Policy** of the OPC UA connection point supports only **None**. **Security Mode** supports **None** and **Username/Password**.

- ii. Create a node under the created connection point and enter the node ID.

7. Deliver the collection policy.

The following uses OPC UA as an example to describe how to deliver a collection policy to EdgeShell.

On the **OPC UA Configuration** tab page of the device details page, select the target OPC UA connection point and click **Deliver Policy**.

| Name                | Connection Point Address | Timeout Interval (ms) | Security Policy | Username | Collection Interval (ms) | Regular Collection | Topic Name          | Collection Node | Operation   |
|---------------------|--------------------------|-----------------------|-----------------|----------|--------------------------|--------------------|---------------------|-----------------|-------------|
| opcua_connect-ec... | opc.tcp://               | 200                   | None            | --       | 5000                     | No                 | 85jR9Fo72262/out... | ['atadada']     | Edit Delete |

8. Configure the rule engine.

Configure data for the gateway to forward collected device data to MQS.

- a. On the **Rule Engine** page, click **Create Rule**. In the dialog box displayed, select the same application as that of the gateway product.

**Create Rule** ×

Integration Application  × | Q

\* Rule Name   
Enter 1 to 64 characters. Only letters, digits, hyphens (-), and underscores (\_) are allowed.

Description   
0/200

Status  Enabled

OK Cancel

- b. Click the rule engine name to access the rule engine details page and configure the source and destination.
  - Select the created product and device as the data source.
  - Select MQS in the same instance as the destination.

**Source** ● After a source is created or deleted, the message forwarding rule takes effect within 1 minute.

| Product Name   | Device Name | Topic Name        | Topic Level   | Base64 Encoding | Include Device Data | Operation |
|----------------|-------------|-------------------|---------------|-----------------|---------------------|-----------|
| product        | ---         | XSKy41INRp3/out/← | Product-level | No              | No                  | Delete    |
| product-modbus | ---         | Wl8AhlEwKXK/out/← | Product-level | No              | No                  | Delete    |

+ Create Source

**Data Parsing** ● The Base64 Encoding and Include Device Data options will become invalid after the data parsing settings are applied. Disabled

**Destination** ● After a destination is created or deleted, the message forwarding rule will take effect within 1 minute.

| Destination | Connection Address | Topic Name | Operation |
|-------------|--------------------|------------|-----------|
| ROMA_MQS    | .....              | topic      | Delete    |

## 9.7 Rule Engine

### 9.7.1 Configuring Data Forwarding Rules

#### Overview

A rule engine can subscribe to device topics, obtain data reported by devices, and send parsed data to other cloud services for use. For example, a user can define a rule that requires a device to report its temperature every hour. If the device temperature exceeds the normal range, the device will be shut down and an alarm notification will be sent to the user. The device integration component transmits collected data to the big data analysis platform to evaluate the risks of other devices.

#### Prerequisites

Each data forwarding rule must belong to an integration application. Before creating a rule, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

## Creating a Rule

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Rule Engine**. On the page displayed, click **Create Rule** in the upper right corner.
3. In the **Create Rule** dialog box, set rule parameters and click **OK**.

**Table 9-21** Parameters for creating a rule

| Parameter        | Description                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| Application Name | Select the integration application to which the rule belongs.                                                                        |
| Rule Name        | Enter a rule name. It is recommended that you enter a name based on naming rules to facilitate search.                               |
| Description      | Enter a brief description of the rule.                                                                                               |
| Status           | This parameter specifies whether to enable the rule. The rule is enabled by default. The rule takes effect only after being enabled. |

4. After the rule is created, click its name in the rule list to access the rule details page.
5. Configure a source for the rule.
  - a. Click **Create Source** in the **Source** area.
  - b. Configure source parameters and click **Save**.

**Table 9-22** Parameters for creating a source

| Parameter           | Description                                                                                                                                                                                                                                                                                                   |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Product Name        | Select the product to which a device belongs.                                                                                                                                                                                                                                                                 |
| Device Name         | Select the device that needs to forward data. You can select a specified device or all devices.                                                                                                                                                                                                               |
| Topic Name          | Select the topic used by the device to report messages.                                                                                                                                                                                                                                                       |
| Topic Level         | Select the topic level. The value is automatically adapted based on the value of <b>Device Name</b> . If you do not specify <b>Device Name</b> , <b>Topic Level</b> is <b>Product</b> . If you select a specific device from the <b>Device Name</b> drop-down list box, <b>Topic Level</b> is <b>Device</b> . |
| Base64 Encoding     | This parameter specifies whether Base64 encoding is performed on the forwarded device data.                                                                                                                                                                                                                   |
| Include Device Data | This parameter specifies whether the forwarded device information contains device data.                                                                                                                                                                                                                       |

6. (Optional) Configure data parsing to filter the data to be forwarded.  
For details about the SQL parsing configuration, see [SQL Parsing](#).

 **NOTE**

After data parsing is applied, the **Base64 Encoding** and **Include Device Data** parameters do not take effect.

7. Configure a destination for the rule.
  - a. Click **Create Destination** in the **Destination** area.
  - b. Configure destination parameters and click **Save**.

**Table 9-23** Parameters for creating a destination

| Destination Type | Parameter          | Description                                                                                                                                                                                                                                              |
|------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ROMA MQS         | Connection Address | Select the connection address of ROMA MQS.                                                                                                                                                                                                               |
|                  | Topic Name         | Select the name of the topic to which data is to be forwarded.                                                                                                                                                                                           |
|                  | Username           | This parameter is available only if <b>MQS SASL_SSL</b> is enabled for the ROMA Connect instance.<br>Enter the key of the integration application to which the topic defined in <b>Topic Name</b> belongs.                                               |
|                  | Password           | This parameter is available only if <b>MQS SASL_SSL</b> is enabled for the ROMA Connect instance.<br>Enter the secret of the integration application to which the topic defined in <b>Topic Name</b> belongs.                                            |
| DIS              | Stream             | Select the name of the DIS stream to which data is to be forwarded. A DIS stream is a logical unit created by tenants to distinguish real-time data of different tenants. When DIS is used to send or receive data, you need to specify the stream name. |
|                  | Service            | Select an entrusted service. The entrusted service allows a user to create an entrustment on IAM and grant ROMA Connect the permission to access DIS.                                                                                                    |
| Device topic     | Product            | Select the product to which a device belongs.                                                                                                                                                                                                            |
|                  | Device             | Select the name of the device to which data is to be forwarded.                                                                                                                                                                                          |

| Destination Type | Parameter | Description                                                    |
|------------------|-----------|----------------------------------------------------------------|
|                  | Topic     | Select the name of the topic to which data is to be forwarded. |

## SQL Parsing

### Concepts

After a device is connected to ROMA Connect, the device encapsulates data into a message in JSON format and sends the message to ROMA Connect. JSON contains keys and values. To facilitate understanding, a rule can be considered as an SQL statement and JSON can be considered as a table. **Key** is the column of the table, and **Value** is the column value of the table. Device messages are filtered using the SQL statement and then sent to other services.

For example, a temperature sensor is used to control the temperature of devices. It can collect the device type, ambient temperature, ambient humidity, and current time. The format and content of the reported information is shown in the example.

```
{
  "device":camera,
  "temperature":30,
  "humidity":65,
  "time":"xxx,xxx"
}
```

If you want to make a rule that requires an alarm message to be sent when the temperature is higher than 20 degrees Celsius or lower than 15 degrees Celsius, enter the following SQL statement: If the preceding conditions are met, LINK reports the device type, absolute value of the device temperature, device humidity, and time for further processing.

```
SELECT
  device, abs(temperature),
  humidity ,
  TIME
FROM
  mcxeSR187154/OUT/test
WHERE
  temperature > 20
OR   temperature < 15
```

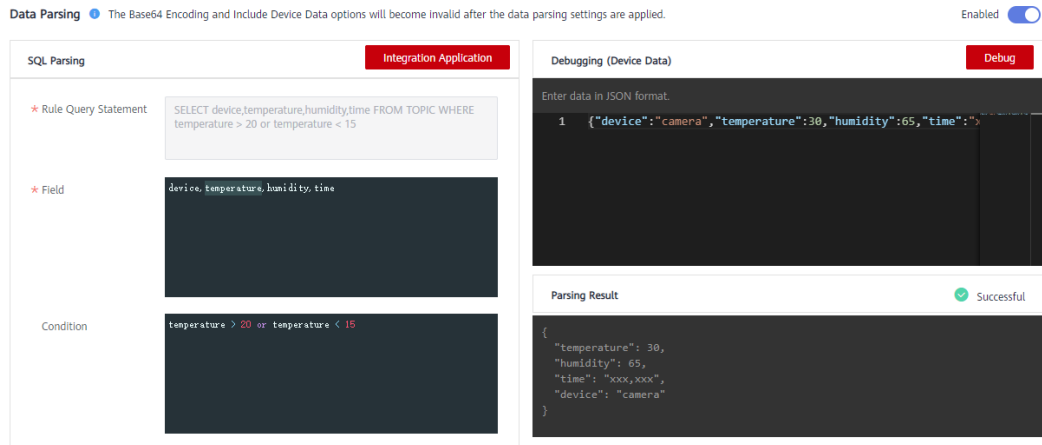
#### NOTE

In the **FROM** statement, **mcxeSR187154/out/test** indicates that the rule engine accepts only messages from the device named **test**. Your device may be different from that in the example. Modify the device information based on the site requirements.

This rule is triggered when the temperature in reported data is greater than 20°C or less than 15°C. In addition, the temperature, device name, and location in the data will be parsed for further processing. [Figure 9-2](#) shows the data parsing result.



**Figure 9-2** Data parsing result



### Usage Description

An SQL statement consists of three parts: SELECT statement, From statement, and Where statement. Data in JSON format is classified into two types: constant data with single or double quotation marks and variable data without any quotation marks.

The field in the **SELECT** statement is the value of the key in the JSON message. Built-in SQL functions are supported. For details about how to use other SQL functions, see [Table 9-25](#). In addition, the SELECT statement supports not only the combination of asterisks (\*) and functions, but also arrays and JSON with nested values. For example, you can use `a.color` to obtain `"color":"red"` in `{"a":{"temperature":29, "color":"red"}}`. Note that the fields without quotation marks are variables, and the fields with single quotation marks and double quotation marks are constants.

The temperature can be a positive number, 0, or negative number. To facilitate management, **abs(temperature)** in the preceding example uses the `abs()` function to output the absolute value of the temperature.

The **FROM** statement contains the device name. You can specify a single device or all devices of a product to report messages. *Product name\_out\_Device name* indicates a single device. After the rule is executed, it applies only to this device. *Product name\_out\_+* can match all devices under this product because the plus sign (+) represents all items in this level. After the rule is executed, it applies to all devices under the product.

The **WHERE** statement contains condition expressions and is used to filter fields and messages that meet conditions. In the preceding example, **WHERE temperature > 20 or temperature < 15** is the filter criterion. Messages will be filtered out only when the temperature is higher than 20°C or lower than 15°C. For details about the condition expressions supported by the **WHERE** statement, see [Table 9-24](#).

**Table 9-24** Condition expression

| Operator                                        | Description                                               | Example                                                                                                                   |
|-------------------------------------------------|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| =                                               | Equal                                                     | color = 'red'                                                                                                             |
| <>                                              | Not equal to                                              | color <>'red'                                                                                                             |
| and                                             | And                                                       | color = 'red' and switch = 'on'                                                                                           |
| or                                              | Or                                                        | color = 'red' or switch = 'on'                                                                                            |
| ( )                                             | Represent a whole.                                        | a>1 and (b<1 or b>5). The judgment of the OR logic is executed first, and then the judgment of the AND logic is executed. |
| in                                              | Only enumeration is supported. Subquery is not supported. | where a in(1,2,3). The example of <b>where a in(select xxx)</b> is not supported.                                         |
| +                                               | Addition                                                  | a in (3,2,3+8)                                                                                                            |
| -                                               | Subtraction                                               | 13 - 2                                                                                                                    |
| /                                               | Divide                                                    | 25 / 5                                                                                                                    |
| *                                               | Multiple                                                  | 2 * 8                                                                                                                     |
| %                                               | Obtains the remainder.                                    | 10 % 2                                                                                                                    |
| <                                               | Less than                                                 | 1 < 3                                                                                                                     |
| <=                                              | Less than or equal to                                     | 1 <= 3                                                                                                                    |
| >                                               | Greater than                                              | 8 > 3                                                                                                                     |
| >=                                              | Greater than or equal to                                  | 8 >= 3                                                                                                                    |
| CASE ...<br>WHEN ...<br>THEN ...<br>ELSE ...END | Case expression (Nested expression is not supported.)     | <b>case a when 3 then 'hello' when 4 then 'bye' end</b> FROM item WHERE a >= b+c"                                         |

A rule engine also provides a variety of functions that you can use when you write SQL statements to diversify your data processing. You can use functions in SQL statements to obtain or process data. For example, "**SELECT** service, abs(temperature)" uses the abs(number) function. For details about the condition expression, see [Table 9-25](#).

**Table 9-25** SQL functions

| Function                                | Description                                                                                                                                                                                |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| abs(number)                             | Returns the absolute value.                                                                                                                                                                |
| sin(n)                                  | Returns the arc sine of the n value.                                                                                                                                                       |
| cos(number)                             | Returns the arc sine of the number value.                                                                                                                                                  |
| asin(number)                            | Returns the arc sine of the number value.                                                                                                                                                  |
| sinh(n)                                 | Returns the hyperbolic cosine of the n value.                                                                                                                                              |
| cosh(number)                            | Returns the hyperbolic cosine of the number value.                                                                                                                                         |
| tan(n)                                  | Returns the tangent of the n value.                                                                                                                                                        |
| tanh(n)                                 | Returns the hyperbolic tangent of the n value.                                                                                                                                             |
| lower(string)                           | Returns the lowercase character string.                                                                                                                                                    |
| upper(string)                           | Returns the uppercase letters.                                                                                                                                                             |
| power(n,m)                              | Return the value of <b>n</b> raised to the power of <b>m</b> .                                                                                                                             |
| rand()                                  | Returns a random number ranging from 0 to 1.                                                                                                                                               |
| mod(n, m)                               | Returns the remainder of n%m.                                                                                                                                                              |
| log(n, m)                               | Returns the natural logarithm. If the value of m does not need to be transferred, log(n) is returned.                                                                                      |
| exp(number)                             | Returns the specified power of a number.                                                                                                                                                   |
| floor(number)                           | Returns the closest integer, whose value is less than or equal to the floating point number.                                                                                               |
| concat(string1, string2)                | String concatenation. For example, concat(field,a) outputs fielda.                                                                                                                         |
| replace(source, substring, replacement) | Replaces the value of a target column. For example, replace(field,'iel','oo') outputs <b>food</b> .                                                                                        |
| topic()                                 | Returns the information about the entire topic. For example, Topic: /abcdef/ghi. Use the topic() function to return / <b>abcdef/ghi</b> .                                                  |
| endswith(input, suffix)                 | Checks whether the input value ends with the suffix.                                                                                                                                       |
| timestamp(format)                       | If no parameter is specified, the default timestamp is returned. If a parameter is specified, the timestamp in the specified format is returned. For example, timestamp() = 1553572557420. |

| Function       | Description                                                                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timestampUtc() | Obtains the UTC time of the system. If no parameter is specified, the current system time in milliseconds is returned. If one parameter is specified, the parameter is used as the format parameter for time formatting. |
| serviceId()    | Returns the service ID corresponding to the message. The parameter in this function must be left unspecified.                                                                                                            |
| clientId()     | Obtains the client ID of the current topic. The parameter in this function must be left unspecified.                                                                                                                     |

## 9.7.2 Importing or Exporting Rules

### Overview

ROMA Connect supports rule import and export. You can export rules as a file to the local host or import a local rule file to ROMA Connect to migrate rule information in batches.

### Prerequisites

- Before importing a rule, ensure that the integration application to which the rule belongs has been created. Otherwise, [create an integration application](#) first.
- Before importing a rule, ensure that the product to which the rule belongs has been created. Otherwise, [create the product](#) in advance.
- Before importing a rule, ensure that the device to which the rule belongs has been created. Otherwise, [Registering a Device](#) in advance.
- Before importing a rule, check whether a duplicate rule name exists in the instance to which the rule is to be imported. If there are duplicate rule names, the rule will fail to be imported.
- Ensure that the quota of rules meets the requirements before importing the rules.

### Importing Rules

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **LINK > Rule Engine**. On the **Rule Engine** page, click **Import**.
3. In the dialog box displayed, select a local rule file and import it.
4. After the import is successful, you can view the imported rules in the rule list.

### Exporting Rules

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.

2. In the navigation tree on the left, choose **LINK > Rule Engine**.
3. Export rules.
  - To export specified rules, select the rules and click **Export**. The system will export the rule file to the local PC.
  - To export all rules, click **Export All**. The system will export the rule file to the local PC.

## 9.8 Subscription Management

### 9.8.1 Subscribing to Device Notifications

#### Overview

ROMA Connect provides the device change notification function. When the status of a subscribed device changes, for example, the device goes online, goes offline, or is deleted, ROMA Connect sends a message to the corresponding message integration topic to obtain the latest device status in real time.

#### Prerequisites

A topic is available for receiving messages, and the topic is in the same application as a device. Otherwise, [create a topic](#) first.

#### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation tree on the left, choose **LINK > Subscription Management**.
3. On the **Subscription Management** page, select the integration application to which the topic that needs to receive notifications belongs.
4. Under the notification type to be subscribed to, select a topic to receive notifications and enable the notification function.
  - After a notification function is enabled or disabled, it takes a maximum of 30 seconds to take effect.
  - A notification is triggered when any of the following operations is performed:
    - The device type is changed. For example, a change from a common device to a subdevice.
    - The device name is changed.
    - The device status is changed.
    - The device description is changed.
  - For details about the notification message example, see [Appendix: Packets of Subscribed Notification Messages](#).

## 9.8.2 Appendix: Packets of Subscribed Notification Messages

This section uses a common device as an example to describe sample MQS packets related to subscription management.

- Message attributes

```
{
  "deviceIdentifier":"iottest2", //Device identifier
  "eventTime":"1588843914884", //Message time
  "gatewayIdentifier":"null", //Gateway identifier
  "deviceId":"D114VKx7a", //Device ID
  "gatewayId":"null", //Gateway ID
  "TAGS":"null" //Message tag
}
```

 NOTE

- For a common device or gateway, set **gatewayId** and **gatewayIdentifier** to **null**.
  - For a subdevice, **gatewayId** and **gatewayIdentifier** are **deviceId** and **deviceIdentifier** of the gateway respectively.
- Message content

- a. Adding a device

```
{
  'deviceType':'Default', //Device type
  'instanceOid':'40', //Instance OID
  'manufacturerName':'iotcompro1', //Manufacturer name
  'productOid':'1', //Product OID
  'manufacturerId':'iotcompro1', //Manufacturer ID
  'remark':'', //Device description
  'deviceId':'D114VKx7a', //Device ID
  'deviceName':'iottest2', //Device name
  'productName':'iotcompro1', //Product name
  'notifyType':'deviceAdded', //Message type
  'deviceIdentifier':'iottest2', //Device identifier
  'deviceOid':'14', //Device OID
  'createdBy':'admin', //Creator
  'eventTime':'1588843914884', //Message time
  'model':'iotcompro1', //Product model
  'productType':'0', //Product type. 1: gateway product; 0: common product.
  'status':'0', //Device status. 0: enabled; 1: disabled.
  'parentDeviceOid':'null' //Parent device OID
}
```

- b. Deleting a device

```
{
  'notifyType':'deviceDeleted', //Message type
  'deviceIdentifier':'iottest2', //Device identifier
  'eventTime':'1588843914884', //Message time
  'deviceId':'D114VKx7a', //Device ID
}
```

- c. Taking a device online

```
{
  'notifyType':'deviceOnline', //Message type: device online
  'deviceIdentifier':'iottest2', //Device identifier
  'eventTime':'1588843914884', //Message time
  'deviceId':'D114VKx7a', //Device ID
}
```

- d. Taking a device offline

```
{
  'notifyType':'deviceDeleted', //Message type
  'deviceIdentifier':'iottest2', //Device identifier
  'eventTime':'1588843914884', //Message time
  'deviceId':'D114VKx7a', //Device ID
}
```

e. Changing the device type

```
{
  'instanceOid':'40', //Instance OID
  'productOid':'1', //Product OID
  'remark':'', //Device description
  'deviceId':'D114VKx7a', //Device ID
  'deviceName':'iottest1', //Device name
  'notifyType':'deviceChange', //Message type
  'deviceIdentifier':'iottest1', //Device identifier
  'deviceOid':'1', //Device OID
  'createdBy':'admin', //Creator
  'eventTime':'1588843914884', //Message time
  'status':'0', //Device status. 0: enabled; 1: disabled.
  'parentDeviceOid':'null' //Parent device OID
}
```

# 10 Data Source Management

---

[Data Sources Supported by ROMA Connect](#)

[Connecting to an API Data Source](#)

[Connecting to an ActiveMQ Data Source](#)

[Connecting to an ArtemisMQ Data Source](#)

[Connecting to a DB2 Data Source](#)

[Connecting to a DIS Data Source](#)

[Connecting to a DWS Data Source](#)

[Connecting to a GaussDB 100 Data Source](#)

[Connecting to a GaussDB 200 Data Source](#)

[Connecting to an FTP Data Source](#)

[Connecting to an FI HDFS Data Source](#)

[Connecting to an FI Hive Data Source](#)

[Connecting to an FI Kafka Data Source](#)

[Connecting to an HL7 Data Source](#)

[Connecting to a HANA Data Source](#)

[Connecting to a HIVE Data Source](#)

[Connecting to an LDAP Data Source](#)

[Connecting to an IBM MQ Data Source](#)

[Connecting to a Kafka Data Source](#)

[Connecting to a MySQL Data Source](#)

[Connecting to a MongoDB Data Source](#)

[Connecting to an MRS Hive Data Source](#)

[Connecting to an MRS HDFS Data Source](#)



- [Connecting to an MRS HBase Data Source](#)
- [Connecting to an MRS Kafka Data Source](#)
- [Connecting to an OBS Data Source](#)
- [Connecting to an Oracle Data Source](#)
- [Connecting to a PostgreSQL Data Source](#)
- [Connecting to a Redis Data Source](#)
- [Connecting to a RabbitMQ Data Source](#)
- [Connecting to an SAP Data Source](#)
- [Connecting to an SNMP Data Source](#)
- [Connecting to a SQL Server Data Source](#)
- [Connecting to a TaurusDB Data Source](#)
- [Connecting to a WebSocket Data Source](#)
- [Connecting to a Custom Data Source](#)

## 10.1 Data Sources Supported by ROMA Connect

### Data Sources Supported by Data Integration Task

**Table 10-1** Data sources supported for data integration

| Data Source Type | Compatibility                                                          |
|------------------|------------------------------------------------------------------------|
| API              | N/A                                                                    |
| ActiveMQ         | 5.15.9                                                                 |
| ArtemisMQ        | 2.9.0                                                                  |
| DB2              | 9.7                                                                    |
| DIS              | N/A                                                                    |
| DWS              | 1.3.4                                                                  |
| FI HDFS          | FusionInsight HD V100R002C80U20,<br>FusionInsight HD 6.5.1             |
| FI Hive          | FusionInsight HD V100R002C80U20,<br>FusionInsight HD 6.5.1             |
| FI Kafka         | FusionInsight HD V100R002C80U20,<br>FusionInsight HD 6.5.1             |
| FTP              | N/A                                                                    |
| GaussDB 100      | FusionInsight_LibrA_V100R003C20 and<br>FusionInsight_LibrA_V300R001C00 |

| Data Source Type    | Compatibility                                                                     |
|---------------------|-----------------------------------------------------------------------------------|
| GaussDB 200         | FusionInsight_LibrA_V100R002C70SPC 100,<br>FusionInsight_LibrA_V100R002C80SPC 300 |
| HL7                 | 2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.6, 2.7, 2.8, or 2.8.1                           |
| HANA                | 1.0                                                                               |
| IBM MQ              | 9.1                                                                               |
| Kafka               | 1.1.0 and 2.3.0                                                                   |
| LDAP                | N/A                                                                               |
| MongoDB             | 3.4                                                                               |
| MRS Hbase           | MRS 1.8.*                                                                         |
| MRS HDFS            | MRS 1.8.*                                                                         |
| MRS Hive            | MRS 2.1.*                                                                         |
| MRS Kafka           | MRS 1.8.*                                                                         |
| MySQL               | 5.7, 8.0                                                                          |
| OBS                 | 3                                                                                 |
| Oracle              | 11g, 12g                                                                          |
| PostgreSQL          | 11                                                                                |
| RabbitMQ            | 3.6.10                                                                            |
| Redis               | 3.0.7 or 4.0.11                                                                   |
| SAP                 | SAP Java Connector 3.0.19                                                         |
| SNMP                | v1, v2, or v3                                                                     |
| SQL Server          | 2014                                                                              |
| TaurusDB            | 2.5.6                                                                             |
| WebSocket           | N/A                                                                               |
| Custom data sources | N/A                                                                               |

## Data Sources Supported by Data APIs

**Table 10-2** Data sources supported by data APIs

| Data Source Type | Compatibility                                                                     |
|------------------|-----------------------------------------------------------------------------------|
| DWS              | 1.3.4                                                                             |
| GaussDB 100      | FusionInsight_LibrA_V100R003C20 and FusionInsight_LibrA_V300R001C00               |
| GaussDB 200      | FusionInsight_LibrA_V100R002C70SPC 100,<br>FusionInsight_LibrA_V100R002C80SPC 300 |
| HIVE             | 2.3.2                                                                             |
| MongoDB          | 3.4                                                                               |
| MySQL            | 5.6, 5.7, 8.0                                                                     |
| Oracle           | 11g                                                                               |
| PostgreSQL       | 11.0                                                                              |
| Redis            | 2.8.x, 3.x.x                                                                      |
| SQL Server       | 2008R2, 2012, 2014, 2016, 2017                                                    |

## 10.2 Connecting to an API Data Source

### Overview

ROMA Connect can use API as a data source for data integration tasks. Before using the API data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **API** and click **Next**.
4. Configure the data source connection information.

**Table 10-3** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |
| Description             | Enter the descriptive information.                                                                            |
| Address                 | Enter the request URL of the API, for example, <b>https://example.com/test</b> .                              |
| Request Mode            | Select the request method of the API.                                                                         |

| Parameter           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication Mode | <p>Select the authentication method of the API.</p> <ul style="list-style-type: none"> <li>● <b>None:</b> No authentication is required. Any user can call the API.</li> <li>● <b>Basic Auth:</b> The username and password are used for authentication. APIs can be called only after the authentication is successful.<br/>If you select <b>Basic Auth</b>, you also need to set <b>Username</b> and <b>Password</b> for authentication.</li> <li>● <b>AppKey Auth:</b> The AppKey and AppSecret are used to sign the request. The API can be called only after the signature verification is successful.<br/>If you select <b>AppKey Auth</b>, you also need to select an app authentication type. The app authentication type can be: <ul style="list-style-type: none"> <li>– <b>Default:</b> The AppKey and AppSecret are used to sign the request. The API can be called only after the signature verification is successful. This mode is used by default for APIs of APIC.</li> <li>– <b>Secret:</b> The AppKey and AppSecret are used to authenticate the request.</li> <li>– <b>Jwt:</b> The AppKey, AppSecret, and the API calling timestamp are used to generate signature information and authenticate the request.</li> </ul> </li> <li>● <b>OAuth2.0:</b> User information is obtained through open APIs under permission. The request header carries the authorization information. The API for obtaining the secret information in the header can be called only after being verified.<br/>If you select OAuth2.0 authentication, you also need to select the authorization type and enter the access token URL, client ID, client key, and scope.</li> <li>● <b>Secret:</b> A password is used for authentication. The request header carries the secret information. The API for obtaining the secret information in the header can be called only after being verified.<br/>If you select secret authorization, you need to enter <b>Secret</b>.</li> <li>● <b>MD5:</b> The verification information signature is obtained after the body parameters are encrypted. The request body carries the signature information. The API for obtaining the signature information can be called only after being verified.<br/>If you select secret authorization, you need to enter <b>Secret</b>.</li> <li>● <b>HMAC:</b> The verification information signature is obtained after the body parameters are encrypted. The request body carries the signature information.</li> </ul> |

| Parameter | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <p>The API for obtaining the signature information can be called only after being verified.<br/>If you select secret authorization, you need to enter <b>Secret</b>.</p> <ul style="list-style-type: none"> <li>Other authentication modes: These authentication modes are customized by business partners, such as <b>Key Top Auth</b>, <b>Hik Vision Auth</b>, and <b>li He Auth</b>. Set the authentication parameters based on the customized authentication mode you select.</li> </ul> |

The following is an example of connecting to an API released by ROMA Connect. The API authentication mode is **App authentication**.

Set the authentication mode to **APPKey Auth**, and retain **Default** for the APP authentication type. AppKey and AppSecret are the key and secret of the integration application authorized by the API.

**Figure 10-1** API data source configuration example

The screenshot shows a configuration form with the following fields:

- Address:** https://aa8adbee8cbe470f852f8ed9f6c50663.apic.cn-nor
- Request Mode:** GET
- Authentication Mode:** AppKey Auth
- App Authentication Type:** Default
- AppKey:** a362bce3-e567-4799-a826-18c5acee38fc
- AppSecret:** (masked with dots)

- Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, click **Create**.
  - If the test result is **Failed to connect to the data source**, check whether the address, request mode, authentication mode, AppKey, and AppSecret are correct, and whether the API can be accessed. Then, click **Recheck** until the connection is successful.
- Click **Create**.

## 10.3 Connecting to an ActiveMQ Data Source

### Overview

ROMA Connect can use ActiveMQ as a data source for data integration tasks. Before using the ActiveMQ data source, you need to connect it to ROMA Connect.

## Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **ActiveMQ** and click **Next**.
4. Configure the data source connection information.

**Table 10-4** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                      |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                               |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                 |
| Broker List             | Enter the broker connection address of ActiveMQ, that is, the IP address and port number of the host. Click <b>Add Address</b> to add addresses.                                                                                                                                                                                                                                   |
| Username                | Enter the username for connecting to ActiveMQ.                                                                                                                                                                                                                                                                                                                                     |
| Password                | Enter the password for connecting to ActiveMQ.                                                                                                                                                                                                                                                                                                                                     |
| SSL Authentication      | Determine whether to use SSL authentication for the connection between ROMA Connect and ActiveMQ.                                                                                                                                                                                                                                                                                  |
| Authentication Mode     | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Select the SSL authentication mode. <ul style="list-style-type: none"> <li>• <b>One-way SSL</b>: Only the client (ROMA Connect) authenticates the server (ActiveMQ).</li> <li>• <b>Two-way SSL</b>: The client (ROMA Connect) and server (ActiveMQ) authenticate each other.</li> </ul> |
| Trust Store             | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>This parameter specifies the trust store used by the client during SSL authentication. It must match the private key used by the server.                                                                                                                                                |

| Parameter             | Description                                                                                                                                                                                                                                       |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Trust Store Password  | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Enter the password of the trust store.                                                                                                                 |
| Keystore              | This parameter is mandatory only if <b>SSL Authentication Mode</b> is set to <b>Two-way SSL</b> .<br>This parameter specifies the keystore used by the client during SSL two-way authentication. It must match the public key used by the server. |
| Keystore Password     | This parameter is mandatory only if <b>SSL Authentication Mode</b> is set to <b>Two-way SSL</b> .<br>Enter the password of the keystore.                                                                                                          |
| Keystore Key Password | This parameter is mandatory only if <b>SSL Authentication Mode</b> is set to <b>Two-way SSL</b> .<br>Enter the password of the keystore key. If this parameter is not specified, the keystore key password is the same as the keystore password.  |

The following is an example of connecting to an ActiveMQ in one-way SSL authentication mode. You need to upload the trust store.

**Figure 10-2** ActiveMQ data source configuration example

The screenshot shows a configuration form for an ActiveMQ data source. The fields and their values are as follows:

- Broker List:** 10.10.10.1 / 3303. There is an "Add Address" button below.
- Username:** admin
- Password:** masked with dots
- SSL Authentication:** Radio buttons for "Disable" and "Enable", with "Enable" selected.
- Authentication Mode:** Two tabs: "One-way SSL" (selected) and "Two-way SSL".
- Trust Store:** A file named "client.truststore.jks (0.00B)" is selected, with a green checkmark and an "Upload" button. Below it, a message says "File uploaded successfully. client.truststore.jks".
- Trust Store Password:** A text input field with the placeholder "Enter the trust store password."

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.



- If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.4 Connecting to an ArtemisMQ Data Source

### Overview

ROMA Connect can use ArtemisMQ as a data source for data integration tasks. Before using the ArtemisMQ data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **ArtemisMQ** and click **Next**.
4. Configure the data source connection information.

**Table 10-5** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |
| Description             | Enter the descriptive information.                                                                            |
| Connection Address      | Enter the IP address and port number of the ArtemisMQ.                                                        |
| Username                | Enter the username for connecting to ArtemisMQ.                                                               |
| Password                | Enter the password for connecting to ArtemisMQ.                                                               |
| SSL Authentication      | Determine whether to use SSL authentication for the connection between ROMA Connect and ArtemisMQ.            |

| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication Mode   | <p>This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b>.</p> <p>Select the SSL authentication mode.</p> <ul style="list-style-type: none"> <li>• <b>One-way SSL</b>: Only the client (ROMA Connect) authenticates the server (ArtemisMQ).</li> <li>• <b>Two-way SSL</b>: The client (ROMA Connect) and server (ArtemisMQ) authenticate each other.</li> </ul> |
| Trust Store           | <p>This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b>.</p> <p>This parameter specifies the trust store used by the client during SSL authentication. It must match the private key used by the server.</p>                                                                                                                                                  |
| Trust Store Password  | <p>This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b>.</p> <p>Enter the password of the trust store.</p>                                                                                                                                                                                                                                                    |
| Keystore              | <p>This parameter is mandatory only if <b>SSL Authentication Mode</b> is set to <b>Two-way SSL</b>.</p> <p>This parameter specifies the keystore used by the client during SSL two-way authentication. It must match the public key used by the server.</p>                                                                                                                                    |
| Keystore Password     | <p>This parameter is mandatory only if <b>SSL Authentication Mode</b> is set to <b>Two-way SSL</b>.</p> <p>Enter the password of the keystore.</p>                                                                                                                                                                                                                                             |
| Keystore Key Password | <p>This parameter is mandatory only if <b>SSL Authentication Mode</b> is set to <b>Two-way SSL</b>.</p> <p>Enter the password of the keystore key. If this parameter is not specified, the keystore key password is the same as the keystore password.</p>                                                                                                                                     |

The connection configuration of the ArtemisMQ data source is similar to that of the ActiveMQ data source. For details about the configuration example, see [Figure 10-2](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.5 Connecting to a DB2 Data Source

### Overview

ROMA Connect can use the DB2 database as a data source for data integration tasks. Before using the DB2 data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **DB2** and click **Next**.
4. Configure the data source connection information.

**Table 10-6** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                         |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                  |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                    |
| Connection Mode         | Select the connection mode for the database. <ul style="list-style-type: none"> <li>• <b>Default:</b> The default mode is used for the database connection.</li> <li>• <b>Professional:</b> The JDBC mode is used for database connection.</li> </ul> |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                            |
| Database                | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                              |

| Parameter         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection String | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the DB2 database, for example, <code>jdbc:db2://{hostname}:{port}/{dbname}</code> . <ul style="list-style-type: none"><li>• <code>{hostname}</code> indicates the connection address of the database.</li><li>• <code>{port}</code> indicates the port of the database.</li><li>• <code>{dbname}</code> indicates the name of the database to be connected.</li></ul> |
| Username          | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Password          | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

The connection configuration of the DB2 data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.6 Connecting to a DIS Data Source

### Overview

ROMA Connect can use the DIS database as a data source for data integration tasks. Before using the DIS data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **DIS** and click **Next**.

4. Configure the data source connection information.

**Table 10-7** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                  |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                         |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                           |
| Stream Name             | Enter the name of the DIS stream to be connected.                                                                                                                                                                                                                                                            |
| Data Type               | Select the source data type of the DIS stream. Currently, only JSON is supported.                                                                                                                                                                                                                            |
| Configuration Type      | Select the configuration type of the DIS stream. <ul style="list-style-type: none"> <li>• <b>Basic:</b> The DIS stream created by the current user in the current region can be called by default.</li> <li>• <b>Advanced:</b> DIS streams of different users in different regions can be called.</li> </ul> |
| AK                      | Enter the AK of the user to which the DIS stream belongs. The AK and SK are long-term identity credentials of the user on the cloud service platform.                                                                                                                                                        |
| SK                      | Enter the SK of the user to which the DIS stream belongs. The AK and SK are long-term identity credentials of the user on the cloud service platform.                                                                                                                                                        |
| Endpoint                | This parameter is mandatory only if <b>Configuration Type</b> is set to <b>Advanced</b> .<br>Enter the endpoint of the DIS stream in the format such as <code>https://dis.region1.example.com</code> .                                                                                                       |
| Region                  | This parameter is mandatory only if <b>Configuration Type</b> is set to <b>Advanced</b> .<br>Enter the region where the DIS stream is located.                                                                                                                                                               |
| Project ID              | This parameter is mandatory only if <b>Configuration Type</b> is set to <b>Advanced</b> .<br>Enter the ID of the project to which the DIS stream belongs.                                                                                                                                                    |

The following is an example of connecting to a DIS in other regions.

**Figure 10-3** DIS data source configuration example

The screenshot shows a configuration form for a DIS data source. It contains the following fields and options:

- Stream Name:** doc-test
- Data Type:** JSON
- Configuration Type:** Basic (selected), Advanced
- Endpoint:** https://dis.region1.example.com
- Region:** region1
- Project ID:** 195e56f...ff98c021d
- AK:** G0WB...RZKSICX
- SK:** .....

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.7 Connecting to a DWS Data Source

### Overview

ROMA Connect can use the DWS database as a data source for data integration tasks. Before using the DWS data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **DWS** and click **Next**.

4. Configure the data source connection information.

**Table 10-8** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |
| Description             | Enter the descriptive information.                                                                            |
| Connection Address      | Enter the IP address and port number for connecting the database in the DWS cluster.                          |
| Database                | Enter the name of the database to be connected in the DWS cluster.                                            |
| Username                | Enter the username used to connect to the database.                                                           |
| Password                | Enter the password used to connect to the database.                                                           |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.8 Connecting to a GaussDB 100 Data Source

### Overview

ROMA Connect can use the GaussDB 100 database as a data source for data integration tasks or data API creation. Before using the GaussDB 100 data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.

2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **GaussDB 100** and click **Next**.
4. Configure the data source connection information.

**Table 10-9** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                         |
| Integration Application | Select the integration application to which the GaussDB 100 data source belongs.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Connection Mode         | Select a database connection mode. <ul style="list-style-type: none"> <li>• <b>Default:</b> The system automatically concatenates data source connection character strings based on configured data.</li> <li>• <b>Professional:</b> You need to specify the data source connection string manually.</li> </ul>                                                                                                                                                                                       |
| Version Number          | Select the version number of the database to be connected. V100R003C20 and V300R001C00 are supported.                                                                                                                                                                                                                                                                                                                                                                                                 |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                                                                                                                                                                                                            |
| Database                | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                              |
| Connection String       | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the GaussDB 100 database, in the format of <i>jdbc:postgresql://host:port/dbname</i> . <ul style="list-style-type: none"> <li>• <b>host</b> indicates the IP address for connecting to the database.</li> <li>• <i>port</i> indicates the port of the database.</li> <li>• <i>dbname</i> indicates the name of the GaussDB 100 database to be connected.</li> </ul> |
| Username                | Enter the username used for logging in to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Password                | Enter the password used for logging in to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                               |



The connection configuration of the GaussDB 100 data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.9 Connecting to a GaussDB 200 Data Source

### Overview

ROMA Connect can use GaussDB 200 as a data source for data integration tasks. Before using the GaussDB 200 data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **GaussDB 200** and click **Next**.
4. Configure the data source connection information.

**Table 10-10** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |
| Description             | Enter the descriptive information.                                                                            |

| Parameter          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection Mode    | Select a database connection mode. <ul style="list-style-type: none"><li>• <b>Default:</b> The system automatically concatenates data source connection character strings based on configured data.</li><li>• <b>Professional:</b> You need to specify the data source connection string manually.</li></ul>                                                                                                                                                                                      |
| Connection Address | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                                                                                                                                                                                                        |
| Database           | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                          |
| Connection String  | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the GaussDB 200 database, in the format of <i>jdbc:postgresql://host:port/dbname</i> . <ul style="list-style-type: none"><li>• <b>host</b> indicates the IP address for connecting to the database.</li><li>• <i>port</i> indicates the port of the database.</li><li>• <i>dbname</i> indicates the name of the GaussDB 200 database to be connected.</li></ul> |
| Username           | Enter the username used for logging in to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Password           | Enter the password used for logging in to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                           |

The connection configuration of the GaussDB 200 data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.10 Connecting to an FTP Data Source

### Overview

ROMA Connect can use FTP as a data source for data integration tasks. Before using the FTP data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **FTP** and click **Next**.
4. Configure the data source connection information.

**Table 10-11** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |
| Description             | Enter the descriptive information.                                                                            |
| Protocol                | Select the protocol for connecting to the FTP data source. The options are <b>FTP</b> and <b>SFTP</b> .       |
| Connection Address      | Enter the IP address and port number for connecting the FTP data source.                                      |
| Username                | Enter the username used for logging in to the FTP data source.                                                |
| Password                | Enter the password used for logging in to the FTP data source.                                                |

| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection Mode | <p>This parameter is mandatory only if <b>Protocol</b> is set to <b>FTP</b>.</p> <p>Select the connection mode of the FTP data source.</p> <ul style="list-style-type: none"><li>• <b>Active:</b> The FTP server actively connects to the data port on the FTP client.</li><li>• <b>Passive:</b> The FTP server waits for the FTP client to connect to its data port.</li></ul> |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.11 Connecting to an FI HDFS Data Source

### Overview

ROMA Connect can use FI HDFS as a data source for data integration tasks. Before using the FI HDFS data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **FI HDFS** and click **Next**.
4. Configure the data source connection information.

**Table 10-12** Data source connection information

| Parameter | Description                                                                                                   |
|-----------|---------------------------------------------------------------------------------------------------------------|
| Name      | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |

| Parameter                | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Integration Application  | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                    |
| Description              | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                      |
| HDFS URL                 | <ul style="list-style-type: none"> <li>• If the root directory is used, set this parameter to <b>hdfs://hacluster</b>. This operation requires the administrator rights.</li> <li>• If a planned directory is used, set this parameter to the planned directory.</li> <li>• If a user database directory is used, for example, <b>/user/hdfs/testdb</b>, the user must have the permission on the directory.</li> </ul> |
| Machine-machine Username | Enter the user authentication name, for example, <b>eip_fdi_hdfs</b> .                                                                                                                                                                                                                                                                                                                                                  |
| Configuration File       | Click <b>Upload File</b> to upload the FI HDFS configuration file. For details, see <b>Obtaining the FusionInsight HDFS Configuration File</b> .                                                                                                                                                                                                                                                                        |

### Obtaining the FusionInsight HDFS Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.
 

Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.
- b. Obtain the **core-site.xml**, **hdfs-site.xml**, and **hosts** files.
 

Download the client configuration file from the MRS console by following the procedure described in **Updating a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:

  - Obtain the **hosts** file from **xxx\_Services\_ClientConfig\_ConfigFiles**.
  - Obtain the **core-site.xml** and **hdfs-site.xml** files from **xxx\_Services\_ClientConfig\_ConfigFiles > HDFS > config**.
- c. Generate the FI HDFS configuration file.
 

Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

#### NOTE

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.

- If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.12 Connecting to an FI Hive Data Source

ROMA Connect can use FI Hive as a data source for data integration tasks. Before using the FI Hive data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **FI Hive** and click **Next**.
4. Configure the data source connection information.

**Table 10-13** Data source connection information

| Parameter                | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                     | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                           |
| Integration Application  | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                    |
| Description              | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                      |
| HDFS URL                 | <ul style="list-style-type: none"> <li>• If the root directory is used, set this parameter to <b>hdfs://hacluster</b>. This operation requires the administrator rights.</li> <li>• If a planned directory is used, set this parameter to the planned directory.</li> <li>• If a user database directory is used, for example, <b>/user/hive/testdb</b>, the user must have the permission on the directory.</li> </ul> |
| Machine-machine Username | Enter the user authentication name, for example, <b>ioc_ipass</b> .                                                                                                                                                                                                                                                                                                                                                     |

| Parameter          | Description                                                                                                                           |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| Configuration File | Click <b>Upload File</b> to upload the FI Hive configuration file. For details, see <b>Obtaining the FI Hive Configuration File</b> . |

### Obtaining the FI Hive Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.  
Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.
- b. Obtain the **hiveclient.properties**, **core-site.xml**, **hdfs-site.xml**, and **hosts** files.  
Download the client configuration file from the MRS console by following the procedure described in **Downloading a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:
  - Obtain the **hosts** file from **xxx\_Services\_ClientConfig\_ConfigFiles**.
  - Obtain the **hiveclient.properties** file from **xxx\_Services\_ClientConfig\_ConfigFiles > Hive > config**.
  - Obtain the **core-site.xml** and **hdfs-site.xml** files from **xxx\_Services\_ClientConfig\_ConfigFiles > HDFS > config**.
- c. Generate the FI Hive configuration file.  
Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

 **NOTE**

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
    - If the test result is **Data source connected successfully**, go to the next step.
    - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
  6. Click **Create**.

## 10.13 Connecting to an FI Kafka Data Source

### Overview

ROMA Connect can use Kafka as a data source for data integration tasks. Before using the FI Kafka data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **FI Kafka** and click **Next**.
4. Configure the data source connection information.

**Table 10-14** Data source connection information

| Parameter                | Description                                                                                                                             |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Name                     | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                           |
| Integration Application  | Select the integration application to which the data source belongs.                                                                    |
| Description              | Enter the descriptive information.                                                                                                      |
| Machine-machine Username | Enter the Kerberos authentication username for accessing the Kafka server, for example, <b>ioc</b> .                                    |
| Configuration File       | Click <b>Upload File</b> to upload the FI Kafka configuration file. For details, see <b>Obtaining the FI Kafka Configuration File</b> . |

#### Obtaining the FI Kafka Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.

Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.



- b. Obtain the **fikafka-producer.properties**, **fikafka-consumer.properties**, and **hosts** files.

Download the client configuration file from the MRS console by following the procedure described in **Downloading a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:

Obtain the **fikafka-producer.properties** and **fikafka-consumer.properties** files from **xxx\_Services\_ClientConfig\_ConfigFiles > Kafka > config**.

Copy the value of **bootstrap.servers** in the **fikafka-producer.properties** file to the **fikafka-consumer.properties** file.

- c. Generate the FI Kafka configuration file.

Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

#### NOTE

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
    - If the test result is **Data source connected successfully**, go to the next step.
    - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
  6. Click **Create**.

## 10.14 Connecting to an HL7 Data Source

### Overview

ROMA Connect can use HL7 (a health information exchange standard) as a data source for data integration tasks. Before using the HL7 data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.

3. On the **Default** tab page, select **HL7** and click **Next**.
4. Configure the data source connection information.

**Table 10-15** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                              |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                       |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                         |
| Direction               | Select the direction of the HL7 data source used in data integration. The options are <b>Source</b> and <b>Destination</b> .                                                                                                                                                                               |
| SSL Authentication      | Determine whether to use SSL authentication for the connection between ROMA Connect and HL7.                                                                                                                                                                                                               |
| Keystore                | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> and <b>Direction</b> is set to <b>Source</b> .<br>This parameter specifies the keystore used by the server (ROMA Connect) during SSL two-way authentication. It must match the public key used by the client (HL7).  |
| Keystore Password       | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> and <b>Direction</b> is set to <b>Source</b> .<br>Enter the password of the keystore.                                                                                                                                |
| Keystore Key Password   | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> and <b>Direction</b> is set to <b>Source</b> .<br>Enter the password of the keystore key.                                                                                                                            |
| Trust Store             | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> and <b>Direction</b> is set to <b>Destination</b> .<br>This parameter specifies the trust store used by the client (ROMA Connect) during SSL authentication. It must match the private key used by the server (HL7). |
| Trust Store Password    | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> and <b>Direction</b> is set to <b>Destination</b> .<br>Enter the password of the trust store.                                                                                                                        |
| Whitelist               | This parameter is mandatory only if <b>Direction</b> is set to <b>Source</b> .<br>Determine whether to allow data to be synchronized to the source HL7 server in the whitelist.                                                                                                                            |
| Whitelist Host Server   | This parameter is mandatory only if <b>Direction</b> is set to <b>Source</b> and <b>Whitelist</b> is set to <b>Enable</b> .<br>Enter the IP address of the HL7 server that is allowed for data synchronization.                                                                                            |

| Parameter          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection Address | <p>This parameter is mandatory only if <b>Direction</b> is set to <b>Destination</b>.</p> <p>Enter the IP address and port number for accessing HL7.</p> <p>To obtain the IP address, locate a data integration task whose source is HL7, view the task details, and view <b>Access Address</b> in the source information.</p> <p>To obtain the port, locate a data integration task whose source is HL7, view the task details, and view <b>Access Port</b> in the source information.</p> |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.15 Connecting to a HANA Data Source

### Overview

ROMA Connect can use the HANA database as a data source for data integration tasks. Before using the HANA data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **HANA** and click **Next**.
4. Configure the data source connection information.

**Table 10-16** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                         |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Connection Mode         | Select the connection mode for the database. <ul style="list-style-type: none"><li>• <b>Default:</b> The default mode is used for the database connection.</li><li>• <b>Professional:</b> The JDBC mode is used for the database connection.</li></ul>                                                                                                                                                                                                                                                                |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                                                                                                                                                                                                                            |
| Database Name           | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                                              |
| Connection String       | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the HANA database, for example, <code>jdbc:sap://{hostname}:{port}?databaseName={dbname}</code> . <ul style="list-style-type: none"><li>• <code>{hostname}</code> indicates the connection address of the database.</li><li>• <code>{port}</code> indicates the port of the database.</li><li>• <code>{dbname}</code> indicates the name of the database to be connected.</li></ul> |
| Username                | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Password                | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

The connection configuration of the HANA data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.

- If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.16 Connecting to a HIVE Data Source

### Overview

ROMA Connect can use HIVE as a data source to create data APIs. Before using the HIVE data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **HIVE** and click **Next**.
4. Configure the data source connection information.

**Table 10-17** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                            |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                          |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                   |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                     |
| Connection Mode         | Select the connection mode for the database. <ul style="list-style-type: none"><li>● <b>Default:</b> The default mode is used for the database connection.</li><li>● <b>Professional:</b> The JDBC mode is used for the database connection.</li></ul> |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                             |

| Parameter         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database Name     | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                                   |
| Connection String | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the HIVE database, for example, <code>jdbc:hive2://{hostname}:{port}/{dbname}</code> . <ul style="list-style-type: none"><li>• <code>{hostname}</code> indicates the connection address of the database.</li><li>• <code>{port}</code> indicates the port of the database.</li><li>• <code>{dbname}</code> indicates the name of the database to be connected.</li></ul> |
| Username          | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Password          | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

The connection configuration of the HIVE data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.17 Connecting to an LDAP Data Source

### Overview

ROMA Connect can use LDAP as a data source for data integration tasks. Before using the LDAP data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.

2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **LDAP** and click **Next**.
4. Configure the data source connection information.

**Table 10-18** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                              |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                       |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                         |
| URL                     | Enter the IP address and port number of the server, in the format of <i>ldap://ip:port</i> . <ul style="list-style-type: none"> <li>• <i>ip</i> indicates the access address of the database.</li> <li>• <i>port</i> indicates the access port of the database.</li> </ul> |
| Authentication Type     | Select the security authentication type of the database. The options are <b>Simple</b> and <b>None</b> .                                                                                                                                                                   |
| Username                | This parameter is mandatory only if <b>Authentication Type</b> is set to <b>Simple</b> .<br>Enter the username used for logging in to the database.                                                                                                                        |
| Password                | This parameter is mandatory only if <b>Authentication Type</b> is set to <b>Simple</b> .<br>Enter the password used for logging in to the database.                                                                                                                        |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.18 Connecting to an IBM MQ Data Source

### Overview

ROMA Connect can use IBM MQ as a data source for data integration tasks. Before using the IBM MQ data source, you need to connect it to ROMA Connect.

## Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **IBM MQ** and click **Next**.
4. Configure the data source connection information.

**Table 10-19** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                              |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                            |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                     |
| Description             | Enter the descriptive information.                                                                                                                                                                                                       |
| Connection Address      | Enter the IP address and port number for connecting the IBM MQ.                                                                                                                                                                          |
| Queue Mananger          | Enter the name of the IBM MQ manager.                                                                                                                                                                                                    |
| CCSID                   | Enter the CCSID of the IBM MQ manager.                                                                                                                                                                                                   |
| Stream Name             | Enter the stream name for connecting to the IBM MQ manager.                                                                                                                                                                              |
| Username                | Enter the username for connecting to IBM MQ.                                                                                                                                                                                             |
| Password                | Enter the password for connecting to IBM MQ.                                                                                                                                                                                             |
| SSL Authentication      | Determine whether to use SSL authentication for the connection between ROMA Connect and IBM MQ.                                                                                                                                          |
| Cipher Suite            | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Enter the name of the cipher suite used in the queue manager stream specified by <b>Stream Name</b> .                                         |
| Trust Store             | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Enter the trust store used by the client (ROMA Connect) during SSL authentication. It must match the private key used by the server (IBM MQ). |



| Parameter            | Description                                                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Trust Store Password | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Enter the password of the trust store. |

The connection configuration of the IBMMQ data source is similar to that of the ActiveMQ data source. For details, see [Connecting to an ActiveMQ Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.19 Connecting to a Kafka Data Source

### Overview

ROMA Connect can use Kafka as a data source for data integration tasks. Before using the Kafka data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **Kafka** and click **Next**.
4. Configure the data source connection information.

**Table 10-20** Data source connection information

| Parameter | Description                                                                                                   |
|-----------|---------------------------------------------------------------------------------------------------------------|
| Name      | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |

| Parameter                           | Description                                                                                                                                                                                                                                      |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Integration Application             | Select the integration application to which the data source belongs.                                                                                                                                                                             |
| Description                         | Enter the descriptive information.                                                                                                                                                                                                               |
| Connection Address                  | Enter the IP address and port number for connecting Kafka.<br>If Kafka has multiple brokers, click <b>Add Address</b> to enter the connection addresses.                                                                                         |
| Enable SSL                          | Determine whether to use SSL authentication for the connection between ROMA Connect and Kafka.                                                                                                                                                   |
| SSL Username/<br>Application Key    | This parameter is mandatory only if <b>Enable SSL</b> is set to <b>Yes</b> .<br>Enter the username used for SSL authentication. If ROMA Connect MQS is used as the Kafka data source, the username is the key of the integration application.    |
| SSL Password/<br>Application Secret | This parameter is mandatory only if <b>Enable SSL</b> is set to <b>Yes</b> .<br>Enter the password used for SSL authentication. If ROMA Connect MQS is used as the Kafka data source, the username is the secret of the integration application. |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.20 Connecting to a MySQL Data Source

### Overview

ROMA Connect can use the MySQL database as a data source for data integration tasks or data API creation. Before using the MySQL data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **MySQL** and click **Next**.
4. Configure the data source connection information.

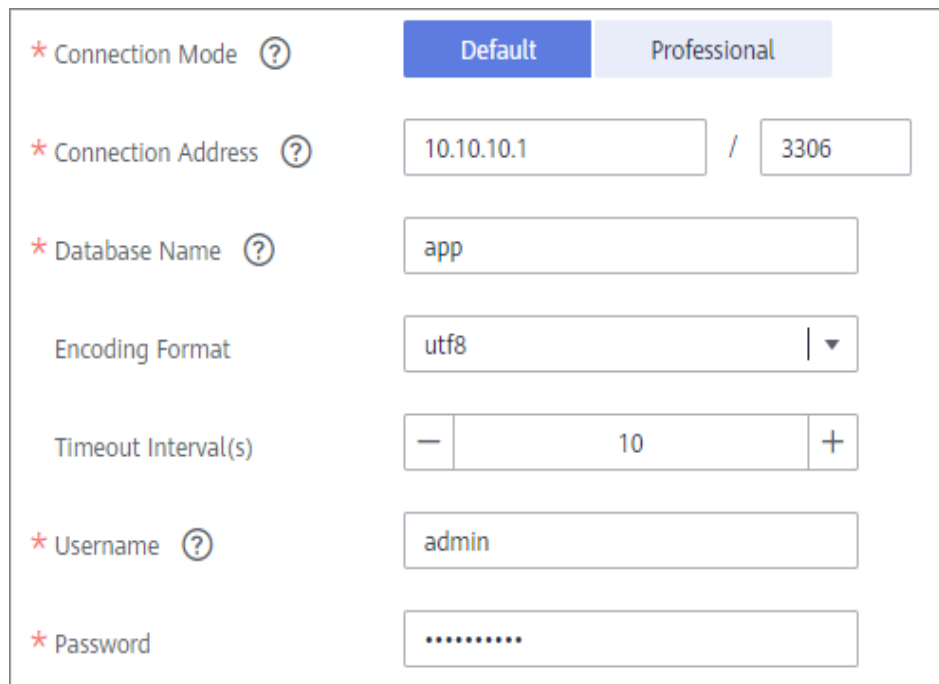
**Table 10-21** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Integration Application | Select the integration application to which the MySQL data source belongs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Connection Mode         | <p>Select a database connection mode.</p> <ul style="list-style-type: none"> <li>• <b>Default:</b> The system automatically concatenates data source connection character strings based on configured data.</li> <li>• <b>Professional:</b> You need to specify the data source connection string manually. <ul style="list-style-type: none"> <li>- If the time zone of data written to a service is different from the default time zone of the MySQL database, you need to use the JDBC mode to connect to the database and specify the time zone in the JDBC connection string. For example, <code>jdbc:mysql://10.10.10.1:3306/testfdi?serverTimezone=GMT%2B8</code>.</li> <li>- If the encoding character set of data written by a service is different from the default encoding character set in the MySQL database, you need to use the JDBC mode to connect to the database and specify the encoding character set in the JDBC connection string. For example, <code>jdbc:mysql://10.10.10.1:3306/testfdi?characterEncoding=utf8</code>.</li> </ul> </li> </ul> |
| Connection Address      | <p>This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b>.</p> <p>Enter the IP address and port number for connecting the database.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Database Name           | <p>This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b>.</p> <p>Enter the name of the database to be connected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

| Parameter         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Encoding Format   | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the encoding format used by the database.                                                                                                                                                                                                                                                                                                                                                                               |
| Timeout Interval  | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the timeout interval for connecting to the database, in seconds.                                                                                                                                                                                                                                                                                                                                                        |
| Connection String | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the MySQL database, for example, <code>jdbc:mysql://{hostname}:{port}/{dbname}</code> . <ul style="list-style-type: none"> <li>• <code>{hostname}</code> indicates the IP address for connecting to the database.</li> <li>• <code>{port}</code> indicates the port of the database.</li> <li>• <code>{dbname}</code> indicates the name of the database to be connected.</li> </ul> |
| Username          | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Password          | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

- The following is an example of the database connection configuration in default mode:

**Figure 10-4** Configuration example in default mode



The screenshot shows a configuration form for a database connection. At the top, there are two tabs: 'Default' (selected) and 'Professional'. Below the tabs, the following fields are visible:

- Connection Mode:** A dropdown menu with 'Default' selected and 'Professional' as an alternative.
- Connection Address:** A text input field containing '10.10.10.1' followed by a slash and another text input field containing '3306'.
- Database Name:** A text input field containing 'app'.
- Encoding Format:** A dropdown menu with 'utf8' selected.
- Timeout Interval(s):** A numeric input field with '10' in the center, flanked by minus and plus signs for adjustment.
- Username:** A text input field containing 'admin'.
- Password:** A text input field with masked characters represented by dots.

- The following is an example of the database connection configuration in professional mode:

**Figure 10-5** Configuration example in professional mode

The screenshot shows a configuration form with the following fields and values:

- Connection Mode:** A dropdown menu with 'Default' and 'Professional' options. 'Professional' is selected.
- Connection String:** A text area containing the JDBC URL: `jdbc:mysql://10.10.10.1:3306/test?characterEncoding=utf8&useSSL=true`
- Username:** A text input field containing the value 'root'.
- Password:** A password input field with masked characters (dots).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.21 Connecting to a MongoDB Data Source

### Overview

ROMA Connect can use the MongoDB database as a data source for data integration tasks or data API creation. Before using the MongoDB data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.

2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **MongoDB** and click **Next**.
4. Configure the data source connection information.

**Table 10-22** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                              |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                            |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                     |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                       |
| Connection Address      | Enter the IP address and port number of MongoDB, in the format of IP:PORT.<br>If MongoDB has multiple replica sets, click <b>Add Address</b> to enter the connection addresses.                                                                                                                                                          |
| Username                | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                      |
| Password                | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                      |
| Authentication Source   | Enter the name of the MongoDB database used for access authentication, that is, the MongoDB database used to authenticate access users. If this parameter is not specified, the database to be connected is used to authenticate users by default.                                                                                       |
| Database Name           | Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                          |
| Cluster                 | Determine whether the MongoDB database to be connected uses the cluster mode. <ul style="list-style-type: none"> <li>• If the MongoDB database uses the cluster mode, select <b>Yes</b> for <b>Cluster</b>.</li> <li>• If the MongoDB database uses the replica set or single-node mode, select <b>No</b> for <b>Cluster</b>.</li> </ul> |
| Replica Set             | This parameter is mandatory only if <b>Cluster</b> is set to <b>No</b> . If the MongoDB database uses the replica set mode, enter the replica set name.                                                                                                                                                                                  |

The following is an example of connecting to a MongoDB data source in single-node mode.

The authentication source is **admin**. That is, the admin database of MongoDB is used to authenticate user root. After the authentication is successful, the user can connect to the doctest database of MongoDB.

**Figure 10-6** MongoDB data source configuration example

The screenshot shows a configuration form for a MongoDB data source. It includes the following fields and options:

- Connection Address:** A text input field containing "10.10.10.1" and a port input field containing "3306". Below the fields is a "+ Add Address" button.
- Username:** A text input field containing "admin".
- Password:** A password input field with masked characters ".....".
- Authentication Source:** A text input field containing "Enter the authentication source".
- Database Name:** A text input field containing "doctest".
- Cluster:** Radio button options for "Yes" and "No", with "No" selected.
- Replica Set:** An empty text input field.

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.22 Connecting to an MRS Hive Data Source

### Overview

ROMA Connect can use MRS Hive as a data source for data integration tasks. Before using the MRS Hive data source, you need to connect it to ROMA Connect.

#### NOTE

If two data integration tasks use MRS data sources of different versions (including MRS Hive, MRS HDFS, and MRS HBase) and Kerberos authentication is enabled for the MRS data sources, the two data integration tasks cannot be executed at the same time. Otherwise, the integration tasks fail.

### Prerequisites

- Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

- The execution permission has been configured for machine-machine interaction users. For details, see "Hive Application Development > Environment Preparation > Preparing a Development User" in the *MapReduce Service Developer Guide*.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **MRS Hive** and click **Next**.
4. Configure the data source connection information.

**Table 10-23** Data source connection information

| Parameter                   | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                        | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                       |
| Integration Application     | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                |
| Description                 | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                  |
| HDFS URL                    | <ul style="list-style-type: none"><li>• If the root directory is used, set this parameter to <b>hdfs://hacluster</b>. This operation requires the administrator rights.</li><li>• If a planned directory is used, set this parameter to the planned directory.</li><li>• If a user database directory is used, for example, <b>/user/hive/testdb</b>, the user must have the permission on the directory.</li></ul> |
| Machine-to-machine Username | Enter the machine-machine username for connecting to MRS Hive.                                                                                                                                                                                                                                                                                                                                                      |
| Configuration File          | Click <b>Upload File</b> to upload the MRS Hive configuration files. For details, see <b>Obtaining the MRS Hive Configuration File</b> .                                                                                                                                                                                                                                                                            |

### Obtaining the MRS Hive Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.  
Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.
- b. Obtain the **hiveclient.properties**, **core-site.xml**, **hdfs-site.xml**, and **hosts** files.



Download the client configuration file from the MRS console by following the procedure described in **Downloading a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:

- Obtain the **hosts** file from `xxx_Services_ClientConfig_ConfigFiles`.
  - Obtain the **hiveclient.properties** file from `xxx_Services_ClientConfig_ConfigFiles > Hive > config`.
  - Obtain the **core-site.xml** and **hdfs-site.xml** files from `xxx_Services_ClientConfig_ConfigFiles > HDFS > config`.
- c. Create a **version** file.  
Create a text file named **Version** without an extension, and add **version=MRS 2.1.0** to the file.
- d. Generate the MRS Hive configuration file.  
Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

 **NOTE**

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
- If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.23 Connecting to an MRS HDFS Data Source

### Overview

ROMA Connect can use MRS HDFS as a data source for data integration tasks. Before using the MRS HDFS data source, you need to connect it to ROMA Connect.

 **NOTE**

If two data integration tasks use MRS data sources of different versions (including MRS Hive, MRS HDFS, and MRS HBase) and Kerberos authentication is enabled for the MRS data sources, the two data integration tasks cannot be executed at the same time. Otherwise, the integration tasks fail.

## Prerequisites

- Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) in advance.
- The execution permission has been configured for machine-machine interaction users. For details, see "HDFS Application Development > Environment Preparation > Preparing a Development User" in the *MapReduce Service Developer Guide*.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **MRS HDFS** and click **Next**.
4. Configure the data source connection information.

**Table 10-24** Data source connection information

| Parameter                   | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                        | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                       |
| Integration Application     | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                |
| Description                 | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                  |
| HDFS URL                    | <ul style="list-style-type: none"><li>• If the root directory is used, set this parameter to <b>hdfs://hacluster</b>. This operation requires the administrator rights.</li><li>• If a planned directory is used, set this parameter to the planned directory.</li><li>• If a user database directory is used, for example, <b>/user/hdfs/testdb</b>, the user must have the permission on the directory.</li></ul> |
| Machine-to-machine Username | Enter the machine-machine username for connecting to MRS HDFS.                                                                                                                                                                                                                                                                                                                                                      |
| Configuration File          | Click <b>Upload File</b> to upload the MRS HDFS configuration file. For details, see <b>Obtaining the MRS HDFS Configuration File</b> .                                                                                                                                                                                                                                                                             |

### Obtaining the MRS HDFS Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.  
Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in

the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.

- b. Obtain the **core-site.xml**, **hdfs-site.xml**, and **hosts** files.

Download the client configuration file from the MRS console by following the procedure described in **Downloading a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:

- Obtain the **hosts** file from **xxx\_Services\_ClientConfig\_ConfigFiles**.
  - Obtain the **core-site.xml** and **hdfs-site.xml** files from **xxx\_Services\_ClientConfig\_ConfigFiles > HDFS > config**.
- c. Generate the MRS HDFS configuration file.

Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

#### NOTE

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
    - If the test result is **Data source connected successfully**, go to the next step.
    - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
  6. Click **Create**.

## 10.24 Connecting to an MRS HBase Data Source

### Overview

ROMA Connect can use MRS HBase as a data source for data integration tasks. Before using the MRS HBase data source, you need to connect it to ROMA Connect.

#### NOTE

If two data integration tasks use MRS data sources of different versions (including MRS Hive, MRS HDFS, and MRS HBase) and Kerberos authentication is enabled for the MRS data sources, the two data integration tasks cannot be executed at the same time. Otherwise, the integration tasks fail.

### Prerequisites

- Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.
- The execution permission has been configured for machine-machine interaction users. For details, see "HBase Application Development >

Environment Preparation >Preparing a Development User" in the *MapReduce Service Developer Guide*.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **MRS HBase** and click **Next**.
4. Configure the data source connection information.

**Table 10-25** Data source connection information

| Parameter                   | Description                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| Name                        | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                              |
| Integration Application     | Select the integration application to which the data source belongs.                                                                       |
| Description                 | Enter the descriptive information.                                                                                                         |
| Machine-to-machine Username | Enter the machine-machine username for connecting to MRS HBase.                                                                            |
| Configuration File          | Click <b>Upload File</b> to upload the MRS HBase configuration files. For details, see <b>Obtaining the MRS HBase Configuration File</b> . |

### Obtaining the MRS HBase Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.  
Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.
- b. Obtain the **core-site.xml**, **hbase-site.xml**, **hdfs-site.xml**, and **hosts** files.  
Download the client configuration file from the MRS console by following the procedure described in **Downloading a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:
  - Obtain the **hosts** file from **xxx\_Services\_ClientConfig\_ConfigFiles**.
  - Obtain the **hbase-site.xml** file from **xxx\_Services\_ClientConfig\_ConfigFiles > HBase > config**.
  - Obtain the **core-site.xml** and **hdfs-site.xml** files from **xxx\_Services\_ClientConfig\_ConfigFiles > HDFS > config**.

- c. Generate the MRS HBase configuration file.

Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

 **NOTE**

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
    - If the test result is **Data source connected successfully**, go to the next step.
    - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
  6. Click **Create**.

## 10.25 Connecting to an MRS Kafka Data Source

### Overview

ROMA Connect can use MRS Kafka as a data source for data integration tasks. Before using the MRS Kafka data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **MRS Kafka** and click **Next**.
4. Configure the data source connection information.

**Table 10-26** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |

| Parameter                   | Description                                                                                                                               |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Description                 | Enter the descriptive information.                                                                                                        |
| Machine-to-machine Username | Enter the machine-machine username for connecting to MRS Kafka.                                                                           |
| Configuration File          | Click <b>Upload File</b> to upload the MRS Kafka configuration file. For details, see <b>Obtaining the MRS Kafka Configuration File</b> . |

### Obtaining the MRS Kafka Configuration File

- a. Obtain **krb5.conf** and **user.keytab** files.  
 Download the user authentication file from MRS Manager by following the procedure described in "Downloading a User Authentication File" in the *MapReduce Service User Guide*, and decompress the file to obtain the **krb5.conf** and **user.keytab** files.
- b. Obtain the **fikafka-producer.properties**, **fikafka-consumer.properties**, and **hosts** files.  
 Download the client configuration file from the MRS console by following the procedure described in **Downloading a Client Configuration File > Using Hive from Scratch** in the *MapReduce Service User Guide*. After the file is decompressed:  
 Obtain the **fikafka-producer.properties** and **fikafka-consumer.properties** files from **xxx\_Services\_ClientConfig\_ConfigFiles > Kafka > config**.  
 Copy the value of **bootstrap.servers** in the **fikafka-producer.properties** file to the **fikafka-consumer.properties** file.
- c. Generate the MRS Kafka configuration file.  
 Save the obtained files to a new directory and compress them into a **.zip** package. All files are stored in the root directory of the **.zip** package.

#### NOTE

- The file name contains a maximum of 255 characters, including only letters and digits.
  - The file size cannot exceed 2 MB.
5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
    - If the test result is **Data source connected successfully**, go to the next step.
    - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
  6. Click **Create**.

## 10.26 Connecting to an OBS Data Source

### Overview

ROMA Connect can use the OBS database as a data source for data integration tasks. Before using the OBS data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **OBS** and click **Next**.
4. Configure the data source connection information.

**Table 10-27** Data source connection information

| Parameter               | Description                                                                                                             |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.           |
| Integration Application | Select the integration application to which the data source belongs.                                                    |
| Description             | Enter the descriptive information.                                                                                      |
| Address                 | Enter the endpoint address of the OBS bucket to be connected. You can obtain the endpoint address from the OBS console. |
| Port                    | It has a fixed value of <b>443</b> .                                                                                    |
| Access Key ID           | Enter the AK of the OBS bucket owner. .                                                                                 |
| Secret Access Key       | Enter the SK of the OBS bucket owner. .                                                                                 |
| Bucket Name             | Enter the name of the OBS bucket to be connected.                                                                       |
| HTTPS                   | This parameter has a fixed value of <b>Yes</b> , indicating that HTTPS is used.                                         |

The following is an example of connecting to an OBS data source.

**Figure 10-7** OBS data source configuration example

The screenshot shows a configuration form for an OBS data source. It contains the following fields and values:

- Address:** obs.region1.example.com
- Port:** 443
- Access Key ID:** LOSIEP...RFZ9
- Secret Access Key:** Masked with 16 dots
- Bucket Name:** doc-test
- HTTPS:** Yes (checked)

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.27 Connecting to an Oracle Data Source

### Overview

ROMA Connect can use the Oracle database as a data source for data integration tasks or data API creation. Before using the Oracle data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **Oracle** and click **Next**.
4. Configure the data source connection information.



**Table 10-28** Data source connection information

| Parameter                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                      | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Integration Application   | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Description               | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Real-Time Synchronization | Determine whether to enable real-time synchronization for the database. Set this parameter to <b>Enable</b> if Oracle is used as the source data source of a composite integration task.                                                                                                                                                                                                                                                                                                                                                 |
| CDC Mode                  | This parameter is mandatory only if <b>Real-Time Synchronization</b> is set to <b>Enable</b> .<br>Select the CDC mode for the database. <ul style="list-style-type: none"> <li>• <b>XStream</b>: The CDC function of the Oracle database in XStream mode is used.</li> <li>• <b>LogMiner</b>: The CDC function of the Oracle database in LogMiner mode is used.</li> </ul>                                                                                                                                                               |
| Outbound Server           | This parameter is mandatory only if <b>CDC Mode</b> is set to <b>XStream</b> .<br>Enter the name of the outbound server, which must be the same as that set in <a href="#">Configuring Oracle CDC (XStream)</a> .                                                                                                                                                                                                                                                                                                                        |
| Database Mode             | This parameter is mandatory only if <b>CDC Mode</b> is set to <b>LogMiner</b> .<br>Enter the name of the schemas where the data table to be synchronized is located. If this parameter is not specified the database username is used by default.                                                                                                                                                                                                                                                                                        |
| Pluggable Database        | This parameter is mandatory only if <b>Real-Time Synchronization</b> is set to <b>Enable</b> .<br>Enter the name of the PDB where the data table to be synchronized is located.                                                                                                                                                                                                                                                                                                                                                          |
| Connection Mode           | Select a database connection mode. <ul style="list-style-type: none"> <li>• <b>Default</b>: The system automatically concatenates data source connection character strings based on your configured data.</li> <li>• <b>Multi-address</b>: You can enter the IP addresses and port numbers of multiple databases. The system automatically concatenates data source connection character strings based on your configured data.</li> <li>• <b>Professional</b>: You need to enter the data source connection string manually.</li> </ul> |

| Parameter             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Connection Address    | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Database Name         | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Database Address List | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Multi-address</b> .<br>Enter the connection address and port number of the distributed database system. Click <b>Add Address</b> to add multiple addresses and port numbers.                                                                                                                                                                                                                                                                                                                                                                               |
| Service Name          | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Multi-address</b> .<br>Enter the service name of the distributed database system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Connection String     | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter a JDBC connection string of the Oracle database, for example,<br><i>jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST={hostname})(PORT={port})) (CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME={servicename})))</i> . <ul style="list-style-type: none"> <li>• <i>{hostname}</i> indicates the IP address for connecting to the database.</li> <li>• <i>{port}</i> indicates the port of the database.</li> <li>• <i>{servicename}</i> indicates the service name of the Oracle database to be connected.</li> </ul> |
| Username              | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Password              | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

For a common data integration task, the connection configuration of the Oracle data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

For a composite task, the following is an example of connecting to an Oracle data source which is not a CDB database and the CDC mode is **LogMiner**.

**Figure 10-8** Oracle data source configuration example

The screenshot shows the configuration interface for an Oracle data source. It includes the following settings:

- Real-Time Synchronization:** A toggle switch set to **Enable** (with **Disable** also visible).
- CDC Mode:** A dropdown menu set to **LogMiner**.
- Database Mode:** A text input field containing **schenaroma**.
- Pluggable Database:** A text input field with the placeholder text **Enter a pluggable database.**
- Connection Mode:** A radio button selection with **Default** selected, and **Multi-address** and **Professional** as other options.
- Connection Address:** Two text input fields containing **10.10.10.1** and **1443**, separated by a slash.
- Database Name:** A text input field containing **doctest**.
- Username:** A text input field containing **admin**.
- Password:** A text input field with masked characters (dots).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.28 Connecting to a PostgreSQL Data Source

### Overview

ROMA Connect can use the PostgreSQL database as a data source for data integration tasks or data API creation. Before using the PostgreSQL data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, **create an integration application** first.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **PostgreSQL** and click **Next**.
4. Configure the data source connection information.

**Table 10-29** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                            |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Connection Mode         | Select a database connection mode. <ul style="list-style-type: none"> <li>• <b>Default:</b> The system automatically concatenates data source connection character strings based on configured data.</li> <li>• <b>Professional:</b> You need to specify the data source connection string manually.</li> </ul>                                                                                                                                                                                          |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                                                                                                                                                                                                               |
| Database Name           | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                                 |
| Connection String       | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the PostgreSQL database, for example, <i>jdbc:postgresql://{hostname}:{port}/{dbname}</i> . <ul style="list-style-type: none"> <li>• <i>{hostname}</i> indicates the IP address for connecting to the database.</li> <li>• <i>{port}</i> indicates the port of the database.</li> <li>• <i>{dbname}</i> indicates the name of the database to be connected.</li> </ul> |
| Username                | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Password                | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

The connection configuration of the PostgreSQL data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.29 Connecting to a Redis Data Source

### Overview

ROMA Connect can use the Redis database as a data source for data integration tasks or data API creation. Before using the Redis data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **PostgreSQL** and click **Next**.
4. Configure the data source connection information.

**Table 10-30** Data source connection information

| Parameter               | Description                                                                                                   |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search. |
| Integration Application | Select the integration application to which the data source belongs.                                          |
| Description             | Enter the descriptive information.                                                                            |

| Parameter   | Description                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Address     | Enter the IP address and port number for connecting Redis. If there are multiple connection addresses, click <b>Add Address</b> to add them.<br><b>NOTE</b><br>If the Redis data source is a cluster Redis, you need to write the addresses of all nodes. If only one node address is written, ROMA Connect regards the Redis as a single-node Redis. As a result, the cluster Redis fails to be accessed. |
| Password    | Enter the password for connecting to Redis.                                                                                                                                                                                                                                                                                                                                                                |
| Database ID | Enter the ID of the Redis database. If this parameter is not specified, the database numbered <b>0</b> is connected by default.                                                                                                                                                                                                                                                                            |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.30 Connecting to a RabbitMQ Data Source

### Overview

ROMA Connect can use RabbitMQ as a data source for data integration tasks. Before using the RabbitMQ data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **RabbitMQ** and click **Next**.
4. Configure the data source connection information.

**Table 10-31** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                 |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                               |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                        |
| Description             | Enter the descriptive information.                                                                                                                                                                                                          |
| Connection Address      | Enter the IP address and port number of the RabbitMQ.                                                                                                                                                                                       |
| Username                | Enter the username for connecting to RabbitMQ.                                                                                                                                                                                              |
| Password                | Enter the password for connecting to RabbitMQ.                                                                                                                                                                                              |
| Virtual Host            | Enter the name of the RabbitMQ virtual host to be connected.                                                                                                                                                                                |
| SSL Authentication      | Determine whether to use SSL authentication for the connection between ROMA Connect and RabbitMQ.                                                                                                                                           |
| Authentication Protocol | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Select the protocol used for SSL authentication. This parameter has a fixed value of <b>TLS</b> .                                                |
| Trust Store             | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Select the trust store used by the client (ROMA Connect) during SSL authentication. It must match the private key used by the server (RabbitMQ). |
| Trust Store Password    | This parameter is mandatory only if <b>SSL Authentication</b> is set to <b>Enable</b> .<br>Enter the password of the trust store.                                                                                                           |

The connection configuration of the RabbitMQ data source is similar to that of the ActiveMQ data source. For details, see [Connecting to an ActiveMQ Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.31 Connecting to an SAP Data Source

### Overview

ROMA Connect can use SAP as a data source for data integration tasks. Before using the SAP data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

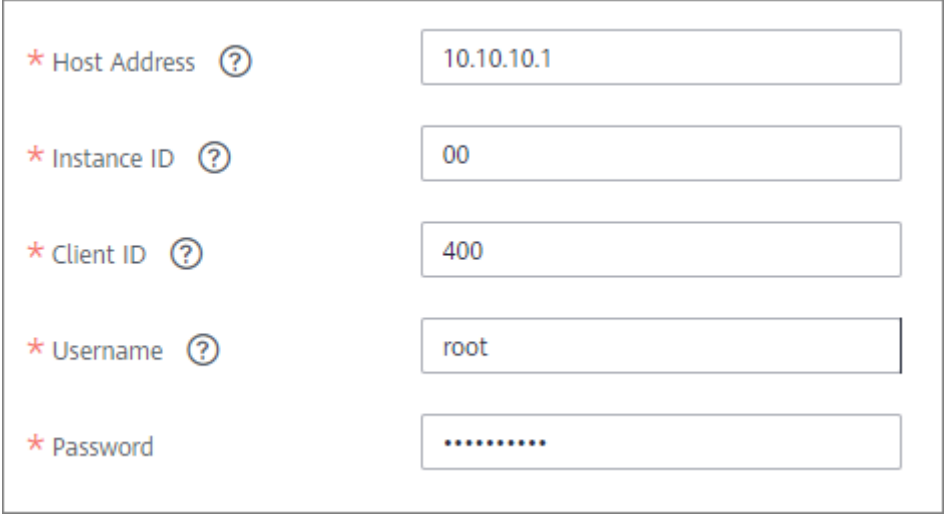
1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **SAP** and click **Next**.
4. Configure the data source connection information.

**Table 10-32** Data source connection information

| Parameter               | Description                                                                                                     |
|-------------------------|-----------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.   |
| Integration Application | Select the integration application to which the data source belongs.                                            |
| Description             | Enter the descriptive information.                                                                              |
| IP Address              | Enter the IP address of the SAP application server.                                                             |
| Instance ID             | Enter the instance number of the SAP application server. The instance number consists of two letters or digits. |
| Customer Code           | Enter the ID of the client to be connected in the SAP system. The client ID consists of three digits.           |
| Username                | Enter the username for connecting to the SAP client.                                                            |
| Password                | Enter the password for connecting to the SAP client.                                                            |

The following is an example of connecting to an SAP data source.



**Figure 10-9** SAP data source configuration example

The screenshot shows a configuration form for an SAP data source. It contains five rows, each with a label, a help icon, and a text input field. The labels are: Host Address, Instance ID, Client ID, Username, and Password. The input fields contain the following values: 10.10.10.1, 00, 400, root, and a masked password (represented by dots).

|                  |            |
|------------------|------------|
| * Host Address ? | 10.10.10.1 |
| * Instance ID ?  | 00         |
| * Client ID ?    | 400        |
| * Username ?     | root       |
| * Password       | .....      |

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.32 Connecting to an SNMP Data Source

### Overview

ROMA Connect can use SNMP as a data source for data integration tasks. Before using the SNMP data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **SNMP** and click **Next**.
4. Configure the data source connection information.

**Table 10-33** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                        |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                 |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                   |
| Connection Address      | Enter the IP address and port number of the SNMP.                                                                                                                                                                                                                                                                                                                                    |
| Network Protocol        | Select the network protocol used by SNMP. The options are <b>UDP</b> and <b>TCP</b> .                                                                                                                                                                                                                                                                                                |
| Version                 | Select the SNMP version. <ul style="list-style-type: none"> <li>● <b>0</b>: SNMPv1</li> <li>● <b>1</b>: SNMPv2</li> <li>● <b>3</b>: SNMPv3</li> </ul>                                                                                                                                                                                                                                |
| Community               | This parameter is mandatory only if <b>Version</b> is set to <b>0</b> or <b>1</b> .<br>Enter the SNMP community name, which is used for identity authentication when you access the SNMP management agent. The community name is equivalent to the access password.                                                                                                                  |
| Security Username       | This parameter is mandatory only if <b>Version</b> is set to <b>3</b> .<br>Enter the security name used for connecting to the SNMP.                                                                                                                                                                                                                                                  |
| Context Name            | This parameter is mandatory only if <b>Version</b> is set to <b>3</b> .<br>Enter the name of the context to be accessed in SNMP.                                                                                                                                                                                                                                                     |
| Security Level          | This parameter is mandatory only if <b>Version</b> is set to <b>3</b> .<br>Select the security level of SNMP. <ul style="list-style-type: none"> <li>● <b>1</b>: Neither authentication nor encryption is required.</li> <li>● <b>2</b>: Authentication is required but encryption is not required.</li> <li>● <b>3</b>: Both authentication and encryption are required.</li> </ul> |
| Authentication Protocol | This parameter is mandatory only if <b>Security Level</b> is set to <b>2</b> or <b>3</b> .<br>Select the authentication protocol for connecting to SNMP. The options are <b>MD5</b> and <b>SHA1</b> .                                                                                                                                                                                |
| Authentication Key      | This parameter is mandatory only if <b>Security Level</b> is set to <b>2</b> or <b>3</b> .<br>Enter the authentication key for connecting to SNMP.                                                                                                                                                                                                                                   |

| Parameter           | Description                                                                                                                                                                             |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Encryption Protocol | This parameter is mandatory only if <b>Security Level</b> is set to <b>3</b> .<br>Select the encryption protocol for connecting to SNMP. The options are <b>DES</b> and <b>TRIDES</b> . |
| Encryption Key      | This parameter is mandatory only if <b>Security Level</b> is set to <b>3</b> .<br>Enter the encryption key for connecting to SNMP.                                                      |

The following is an example for connecting to SNMPv3.

**Figure 10-10** SNMP data source configuration example

The screenshot shows a configuration form for an SNMP data source. The fields are as follows:

- Connection Address**: 10.10.10.1 / 161
- Network Protocol**: UDP
- Version**: 3
- Security Name**: doctest
- Context Name**: testcontext
- Security Level**: 3
- Authentication Protocol**: MD5
- Authentication Key**: .....
- Encryption Protocol**: DES
- Encryption Key**: .....

- Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.

6. Click **Create**.

## 10.33 Connecting to a SQL Server Data Source

### Overview

ROMA Connect can use the SQL Server database as a data source for data integration tasks or data API creation. Before using the SQL Server data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **SQL Server** and click **Next**.
4. Configure the data source connection information.

**Table 10-34** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                     |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                   |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                            |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                              |
| Connection Mode         | Select a database connection mode. <ul style="list-style-type: none"> <li>• <b>Default:</b> The system automatically concatenates data source connection character strings based on configured data.</li> <li>• <b>Professional:</b> You need to specify the data source connection string manually.</li> </ul> |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                      |

| Parameter         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Database Name     | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                                             |
| Connection String | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the SQL Server database, for example, <i>jdbc:sqlserver://{hostname}:{port};DatabaseName={dbname}</i> . <ul style="list-style-type: none"> <li>• <i>{hostname}</i> indicates the IP address for connecting to the database.</li> <li>• <i>{port}</i> indicates the port of the database.</li> <li>• <i>{dbname}</i> indicates the name of the database to be connected.</li> </ul> |
| Username          | Enter the username used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Password          | Enter the password used to connect to the database.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

The connection configuration of the SQL Server data source is similar to that of the MySQL data source. For details, see [Connecting to a MySQL Data Source](#).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.34 Connecting to a TaurusDB Data Source

### Overview

ROMA Connect can use the TaurusDB database as a data source for data integration tasks. Before using the TaurusDB data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

## Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **TaurusDB** and click **Next**.
4. Configure the data source connection information.

**Table 10-35** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Connection Mode         | Select the connection mode for the database. <ul style="list-style-type: none"> <li>● <b>Default:</b> The default mode is used for the database connection.</li> <li>● <b>Professional:</b> The JDBC mode is used for the database connection.</li> </ul>                                                                                                                                                                                                                                                                                                        |
| Connection Address      | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the IP address and port number for connecting the database.                                                                                                                                                                                                                                                                                                                                                                                                       |
| Connection String       | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Professional</b> .<br>Enter the JDBC connection string of the TaurusDB database, for example, <code>jdbc:mysql://{hostname}:{port}/{dbname}?characterEncoding=utf8&amp;useSSL=true</code> . <ul style="list-style-type: none"> <li>● <code>{hostname}</code> indicates the IP address for connecting to the database.</li> <li>● <code>{port}</code> indicates the port of the database.</li> <li>● <code>{dbname}</code> indicates the name of the database to be connected.</li> </ul> |
| Database Name           | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the name of the database to be connected.                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Encoding Format         | This parameter is mandatory only if <b>Connection Mode</b> is set to <b>Default</b> .<br>Enter the encoding format used by the database.                                                                                                                                                                                                                                                                                                                                                                                                                         |

| Parameter | Description                                         |
|-----------|-----------------------------------------------------|
| Username  | Enter the username used to connect to the database. |
| Password  | Enter the password used to connect to the database. |

- The following is an example of the database connection configuration in default mode:

**Figure 10-11** Configuration example in default mode

The screenshot shows a configuration form with the following elements:

- Connection Mode:** Two tabs, 'Default' (selected) and 'Professional'.
- Connection Address:** Two input fields containing '10.10.10.1' and '3306', separated by a slash.
- Database Name:** A text input field containing 'app'.
- Encoding Format:** A dropdown menu currently showing 'utf8'.
- Username:** A text input field containing 'root'.
- Password:** A text input field with masked characters (dots).

- The following is an example of the database connection configuration in professional mode:

**Figure 10-12** Configuration example in professional mode

The screenshot shows a configuration form with the following elements:

- Connection Mode:** Two tabs, 'Default' and 'Professional' (selected).
- Connection String:** A large text area containing the JDBC URL: 'jdbc:mysql://10.10.10.1:3306/test?characterEncoding=utf8&useSSL=true'.
- Username:** A text input field containing 'root'.
- Password:** A text input field with masked characters (dots).

5. Click **Check Connectivity** to check the connectivity between ROMA Connect and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.

- If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
6. Click **Create**.

## 10.35 Connecting to a WebSocket Data Source

### Overview

ROMA Connect can use WebSocket as a data source for data integration tasks. Before using the WebSocket data source, you need to connect it to ROMA Connect.

### Prerequisites

Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.
3. On the **Default** tab page, select **WebSocket** and click **Next**.
4. Configure the data source connection information.

**Table 10-36** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                    | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                                                                                                                                          |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                                                                                                                                                                                                                                   |
| Description             | Enter the descriptive information.                                                                                                                                                                                                                                                                                                                                                                                                     |
| Address                 | Enter the IP address of WebSocket. <ul style="list-style-type: none"> <li>• ws (WebSocket) indicates a plaintext connection. If a plaintext connection is used, the URL must be in the <i>ws://example.com</i> or <i>IP:PORT</i> format.</li> <li>• wss (WebSocket Secure) indicates an encrypted connection. If an encrypted connection is used, the URL must be in the <i>wss://example.com</i> or <i>IP:PORT</i> format.</li> </ul> |



| Parameter           | Description                                                                                                                                                                                                                                                                                                                                                                  |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Authentication Mode | Select the authentication mode used for connecting to WebSocket. <ul style="list-style-type: none"> <li>• <b>None:</b> No authentication is required. Anyone can access WebSocket.</li> <li>• <b>Basic Auth:</b> A user must enter the username and password for authentication. After the authentication is successful, the user is allowed to access WebSocket.</li> </ul> |
| Username            | This parameter is mandatory only if <b>Authentication Mode</b> is set to <b>Basic Auth</b> .<br>Enter the user name for connecting to WebSocket.                                                                                                                                                                                                                             |
| Password            | This parameter is mandatory only if <b>Authentication Mode</b> is set to <b>Basic Auth</b> .<br>Enter the password for connecting to WebSocket.                                                                                                                                                                                                                              |

- Click **Check Connectivity** to check the connectivity between FDI and the data source.
  - If the test result is **Data source connected successfully**, go to the next step.
  - If the test result is **Failed to connect to the data source**, check the data source status and connection parameters, and click **Recheck** until the connection is successful.
- Click **Create**.

## 10.36 Connecting to a Custom Data Source

### Overview

ROMA Connect can use a custom connector as a data source for data integration tasks. Before using a custom data source, you need to connect it to ROMA Connect.

### Prerequisites

- Each connected data source must belong to an integration application. Before connecting a data source, ensure that an integration application is available. Otherwise, [create an integration application](#) first.
- A connector instance is available. Otherwise, [create a connector](#) and [publish a connector](#) first.

### Procedure

- Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
- In the navigation pane on the left, choose **Data Sources**. In the upper right corner of the page, click **Access Data Source**.

3. On the **Custom** tab page, select the connector to be used as the custom data source and click **Next**.
4. Configure the data source connection information.

**Table 10-37** Data source connection information

| Parameter               | Description                                                                                                                                                                                                                        |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Name        | Enter a data source name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                      |
| Integration Application | Select the integration application to which the data source belongs.                                                                                                                                                               |
| Description             | Enter the descriptive information.                                                                                                                                                                                                 |
| Connector Instance      | Select a published connector instance under the connector.                                                                                                                                                                         |
| Other parameters        | Set the subsequent configuration items based on the data source information defined when you create the connector.<br><br>You can find the connector on the <b>Assets</b> page and view the data source metadata of the connector. |

The following is an example of connecting to a custom data source for reading email data.

The **email server**, **protocol**, **user**, and **password** are the data source parameters defined in the connector.

**Figure 10-13** Custom data source configuration example

5. Click **Create**.

# 11 Asset Management

---

[Connectors](#)

[Restoring/Migrating Integration Assets](#)

## 11.1 Connectors

### 11.1.1 Creating a Connector

#### Overview

ROMA Connect supports connections of common data source types, such as relational databases, message queues, APIs, NoSQL databases, and OBS. These data sources can be directly used in ROMA Connect. You can integrate your data from the source to destination only by connecting to data sources and executing data integration tasks.

If the data sources provided by ROMA Connect cannot meet your data integration requirements, you can customize a data source add-on to connect to your data source for data reading and writing. In addition, you can open up the data source add-on to ROMA Connect through a standard RESTful API, so that ROMA Connect can read data from and write data into the data sources. This custom data source add-on is called a connector.

A connector can be directly connected to a data source, which is a data exchange channel between the data source and ROMA Connect. ROMA Connect reads and writes data sources by calling this RESTful API.

After developing and deploying a connector, you need to create the connector on ROMA Connect and define the interconnection parameters of the connector.

#### Procedure

1. Log in to the ROMA Connect console and choose **Assets** in the navigation pane.
2. Click **Create Connector** in the upper right corner of the page.

3. On the **Create Connector** page, enter configuration information about a connector.

**Table 11-1** Parameters for creating a connector

| Parameter   | Description                                                                                                                                                                                                                                                                                                         |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name        | Enter a connector name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                         |
| Type        | Select the permission type of the connector for data operation. <ul style="list-style-type: none"><li>• <b>Reader/Writer:</b> The connector can read and write data sources.</li><li>• <b>Reader:</b> The connector can read data sources.</li><li>• <b>Writer:</b> The connector can write data sources.</li></ul> |
| Description | Enter a brief description of the connector.                                                                                                                                                                                                                                                                         |

| Parameter            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Source Metadata | <p>Define parameters of the data source. When configuring ROMA Connect to access the data source, you need to set the parameters defined here. The message data exchanged between ROMA Connect and the connector is in JSON format, and the parameters are transferred in the Key-Value format.</p> <p>Click <b>Add Parameter</b> to add the parameters for the data source.</p> <ul style="list-style-type: none"> <li>● <b>Parameter Name:</b> specifies the name of a parameter to be displayed on the data source configuration page. It is only used to identify a parameter and does not take effect in an actual data integration task.</li> <li>● <b>Parameter Key:</b> specifies the parameter key transferred in a data source connection request.</li> <li>● <b>Parameter Value Type:</b> specifies the style of the parameter value to be configured on the data source connection page. <ul style="list-style-type: none"> <li>- <b>Text:</b> text box.</li> <li>- <b>Password:</b> character input box. After a character is entered, the character is displayed as a shielded character.</li> <li>- <b>Select:</b> drop-down list box.</li> <li>- <b>Date:</b> date control, which allows you to select a date and time.</li> <li>- <b>Textarea:</b> character text box, which supports line feed.</li> </ul> </li> <li>● <b>Verification Rule:</b> specifies whether to verify the input parameter values. <ul style="list-style-type: none"> <li>- <b>None:</b> The input parameter values are not verified.</li> <li>- <b>No special characters:</b> The system checks whether the parameter value contains special characters. Special characters include digits, letters, hyphens (-), and underscores (_).</li> </ul> </li> <li>● <b>Default Value:</b> specifies a default value to be transferred if a parameter is not specified. If <b>Default Value</b> is not specified, no default value is transferred.</li> <li>● <b>Mandatory:</b> specifies whether the parameter is mandatory.</li> </ul> |

| Parameter       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reader Metadata | <p>Define reader parameters of a data source. When the data source is selected as the source of a data integration task, the reader parameters defined here must be set in the data integration task. The message data exchanged between ROMA Connect and the connector is in JSON format, and the parameters are transferred in the Key-Value format.</p> <p>Click <b>Add Parameter</b> to add the reader parameters of the data source. The parameter configuration description is similar to that of <b>Data Source Metadata</b>.</p>    |
| Writer Metadata | <p>Define writer parameters of a data source. When the data source is selected as the destination of a data integration task, the writer parameters defined here must be set in a data integration task. The message data exchanged between ROMA Connect and the connector is in JSON format, and the parameters are transferred in the Key-Value format.</p> <p>Click <b>Add Parameter</b> to add the writer parameters of the data source. The parameter configuration description is similar to that of <b>Data Source Metadata</b>.</p> |

Figure 11-1 Example of defining connector parameters

The screenshot shows a configuration interface for defining connector parameters. It is divided into two main sections: 'Data Source Metadata' and 'Writer Metadata'. Each section contains a table with columns for Parameter Name, Parameter Key, Parameter Value Type, Verification Rule, Default Value, and Mandatory. Below each table is an 'Add Parameter' button.

| * Data Source Metadata              | * Parameter Name | * Parameter Key | Parameter Value Type | Verification Rule | Default Value | Mandatory                           |
|-------------------------------------|------------------|-----------------|----------------------|-------------------|---------------|-------------------------------------|
|                                     | host             | host            | Text                 | None              |               | <input checked="" type="checkbox"/> |
|                                     | port             | port            | Text                 | None              | 25            | <input checked="" type="checkbox"/> |
|                                     | protocol         | protocol        | Text                 | None              | smtp          | <input checked="" type="checkbox"/> |
|                                     | userName         | userName        | Text                 | None              |               | <input checked="" type="checkbox"/> |
|                                     | password         | password        | Pass...              | None              |               | <input checked="" type="checkbox"/> |
| <input type="radio"/> Add Parameter |                  |                 |                      |                   |               |                                     |
| Writer Metadata                     | * Parameter Name | * Parameter Key | Parameter Value Type | Verification Rule | Default Value | Mandatory                           |
|                                     | sender           | sender          | Text                 | None              |               | <input checked="" type="checkbox"/> |
| <input type="radio"/> Add Parameter |                  |                 |                      |                   |               |                                     |

4. Click **Create**.

## 11.1.2 Publishing a Connector

### Overview

After creating a connector in ROMA Connect, you need to publish the connector so that ROMA Connect can connect to the deployed connector. ROMA Connect uses a published connector instance to connect to the custom data source, read data from, and write data into the data source.

## Prerequisites

- A connector has been developed and deployed offline. For details about how to develop the API for connecting the connector to ROMA Connect, see [RESTful API Specifications of Connectors](#).
- A connector has been created in ROMA Connect. For details, see [Creating a Connector](#).

## Procedure

1. Log in to the ROMA Connect console and choose **Assets** in the navigation pane.
2. On the **Connectors** tab page, click **Publish** of the connector.
3. In the dialog box displayed, set related parameters and click **OK**.

**Table 11-2** Connector configuration

| Parameter            | Description                                                                                                                                                                                                                                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Instance Name        | Enter a connector instance name. It is recommended that you enter a name based on naming rules to facilitate search.                                                                                                                                                                                                    |
| Connection Addresses | Access address of the deployed connector.                                                                                                                                                                                                                                                                               |
| Authentication Mode  | Select the security authentication mode used for accessing the connector instance. For simplicity, select <b>None</b> . <ul style="list-style-type: none"> <li>• <b>None</b>: Access requests are not authenticated.</li> <li>• <b>AppKey</b>: Access requests not authenticated using AppKey and AppSecret.</li> </ul> |
| AppKey               | Enter the AppKey used for connector authentication.                                                                                                                                                                                                                                                                     |
| AppSecret            | Enter the AppSecret used for connector authentication.                                                                                                                                                                                                                                                                  |

### NOTE

Only connectors in the **Editable** state can be edited. Connectors in the **Published** state cannot be edited.

## 11.1.3 RESTful API Specifications of Connectors

### Overview

ROMA Connect interacts with connectors through RESTful APIs exposed by the connectors to read and write data sources. To ensure that ROMA Connect can read and write data properly, the RESTful APIs exposed by the connectors must meet ROMA Connect specifications.

ROMA Connect provides standard specifications for implementation of the data reading and writing APIs exposed by connectors. Connectors must comply with the standard specifications defined by ROMA Connect.

## Data Reading API

### API Specification Definition

- URI

POST /reader

- API request

```
{
  "job_name": "job_name",
  "datasource": {
    "para1": "*****",
    "para2": "*****",
    ...
  },
  "params": {
    "extend": {
      "ex_para1": "*****",
      "ex_para2": "*****",
      ...
    },
    "pagination": {
      "page_no": 1,
      "page_size": "10"
    },
    "migration": {
      "begin": "*****",
      "end": "*****"
    }
  }
}
```

- API response

```
{
  "datas": [
    {
      "para1": "*****",
      "para2": "*****",
      ...
    },
    {
      "para1": "*****",
      "para2": "*****",
      ...
    },
    ...
  ]
}
```

### API Parameters

- Request parameters



**Table 11-3** Request parameters

| Parameter  | Mandatory | Type          | Description                                                                                                                                |
|------------|-----------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| job_name   | Yes       | String        | Task name. The value is a string of 4 to 64 characters, including letters, digits, hyphens (-), and underscores (_).                       |
| datasource | Yes       | Object        | Data source object. The JSON body consists of the parameters on which the connector depends to connect to the database and all parameters. |
| params     | Yes       | <b>Params</b> | Parameter object required by the connector.                                                                                                |

**Table 11-4** Parameter description of params

| Parameter  | Mandatory | Type              | Description                                                                                                   |
|------------|-----------|-------------------|---------------------------------------------------------------------------------------------------------------|
| pagination | No        | <b>Pagination</b> | Pagination object.                                                                                            |
| migration  | No        | <b>Migration</b>  | Incremental migration object.                                                                                 |
| extend     | No        | Object            | Extended parameters to which a connector belongs. The value is a JSON body consisting of extended parameters. |

**Table 11-5** Parameter description of pagination

| Parameter | Mandatory | Type    | Description                     |
|-----------|-----------|---------|---------------------------------|
| page_no   | No        | Integer | Current page number.            |
| page_size | No        | Integer | Number of records on each page. |

**Table 11-6** Parameter description of migration

| Parameter | Mandatory | Type | Description                   |
|-----------|-----------|------|-------------------------------|
| begin     | No        | Date | Start time of data migration. |

| Parameter | Mandato<br>ry | Type | Description                 |
|-----------|---------------|------|-----------------------------|
| end       | No            | Date | End time of data migration. |

- Response parameters

**Table 11-7** Response parameters

| Parameter | Type         | Description                                                                                                                                          |
|-----------|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| datas     | List<Object> | List of data to be read. The value of this parameter must be in JSON array format and is determined by the connector based on the site requirements. |

## Data Writing API

### API Specification Definition

- URI  
POST /writer
- API request

```
{
  "job_name": "job_name",
  "datasource": {
    "para1": "*****",
    "para2": "*****",
    ...
  },
  "params": {
    "extend": {
      "ex_para1": "*****",
      "ex_para2": "*****",
      ...
    }
  },
  "meta-data": [
    {
      "name": "id",
      "type": "String",
      "format": "",
      "path": "datas[i].id"
    },
    {
      "name": "company",
      "type": "String",
      "format": "",
      "path": "datas[i].company"
    },
    ...
  ],
  "datas": [
    {
      "data1": "*****",
      "data2": "*****",
      ...
    },
    {

```

```

        "data1": "*****",
        "data2": "*****",
        ...
    },
    ...
]
}

```

- API response

```

{
  "num_success": "2",
  "num_fail": "0",
  "fail_datas": [
    {}
  ]
}

```

**API Parameters**

- Request parameters

**Table 11-8** Request parameters

| Parameter  | Mandatory | Type                              | Description                                                                                                                                |
|------------|-----------|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| job_name   | Yes       | String                            | Task name. The value is a string of 4 to 64 characters, including letters, digits, hyphens (-), and underscores (_).                       |
| datasource | Yes       | Object                            | Data source object. The JSON body consists of the parameters on which the connector depends to connect to the database and all parameters. |
| params     | Yes       | <a href="#">Params</a>            | Parameter object required by the connector.                                                                                                |
| meta-data  | Yes       | List< <a href="#">Meta-data</a> > | Metadata parameter list.                                                                                                                   |
| datas      | Yes       | List<Object>                      | List of data processed by the connector.                                                                                                   |

**Table 11-9** Parameter description of params

| Parameter | Mandatory | Type   | Description                                                                                                   |
|-----------|-----------|--------|---------------------------------------------------------------------------------------------------------------|
| extend    | No        | Object | Extended parameters to which a connector belongs. The value is a JSON body consisting of extended parameters. |

**Table 11-10** Parameter description of meta-data

| Parameter | Mandatory | Type   | Description                                                                                                                    |
|-----------|-----------|--------|--------------------------------------------------------------------------------------------------------------------------------|
| name      | Yes       | String | Data field name.                                                                                                               |
| type      | Yes       | String | Data field type. The value can be <b>String, Integer, Date, or Long</b> .                                                      |
| format    | No        | String | Format string of data. Format of a character string. This field needs to be specified when <b>type</b> is set to <b>Date</b> . |
| path      | Yes       | String | Path of a field in the source data.                                                                                            |

- Response parameter description

**Table 11-11** Response parameter description

| Parameter   | Type         | Description                                   |
|-------------|--------------|-----------------------------------------------|
| num_success | Integer      | Number of times data is successfully written. |
| num_fail    | Integer      | Number of times data failed to be written.    |
| fail_datas  | List<Object> | List of data that fails to be processed.      |

## 11.2 Restoring/Migrating Integration Assets

### 11.2.1 Overview

#### Scenarios

ROMA Connect allows you to export applications and tasks of an instance as an asset file and import an asset file to restore or migrate integration assets.

- Asset restoration: You can export integration assets from a ROMA Connect instance for data backup. In the event of a faulty instance or data loss, you can import the backup asset file to the instance for restoration.
- Asset migration: You can import an integration asset file that was exported from existing ROMA Connect instances into a new instance to quickly deploy and reuse existing integration assets.

Only integration asset data can be imported, excluding asset-related instance configuration data. After integration asset import, you need to reconfigure asset-related instance data.

**Table 11-12** Instance configurations

| Category                    | Item                                                         | Description                                                                                                                                                                        | Processing Method                                                                                                                                        |
|-----------------------------|--------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Public instance information | Integration application secret                               | The secrets of integration applications are non-asset data.                                                                                                                        | After assets are imported, ROMA Connect automatically generates new secrets. Notify interworking service applications to update the secrets accordingly. |
|                             | Integration application authorization information            | The authorization information of integration applications is non-asset data.                                                                                                       | Reconfigure user authorization after assets are imported.                                                                                                |
| Data integration            | Data source connection information                           | Includes the addresses, accounts, and passwords, which are non-asset data.                                                                                                         | Configure connection information for the corresponding data sources after assets are imported.                                                           |
| Service integration         | APIs that have not been published in the RELEASE environment | Include APIs that have not been published and APIs that have been published in a non-RELEASE environment. These APIs are non-asset data.                                           | Re-create and publish APIs as required after assets are imported.                                                                                        |
|                             | Environment information of APIs                              | Includes the independent domain names, environment variables, cloud server information of load balance channels, and non-default environments. This information is non-asset data. | Reconfigure environment information after assets are imported.                                                                                           |
|                             | Control policy information of APIs                           | Includes request throttling policies, access control policies, client access control policies, and client quota policies. This information is non-asset data.                      | Reconfigure control policy information after assets are imported.                                                                                        |

| Category           | Item                                                  | Description                                                                                                                                              | Processing Method                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | Secret information of APIs                            | Includes signature keys, client's AppCodes, and custom backends' configurations of the password or certificate type. This information is non-asset data. | After asset import: <ul style="list-style-type: none"> <li>ROMA Connect automatically generates new keys and secrets for signature keys. Notify backend services to update the keys and secrets accordingly.</li> <li>Reconfigure the client's AppCode.</li> <li>Reconfigure the configurations of the password or certificate type for the custom backends.</li> </ul> |
|                    | Blacklist and whitelist specific to API authorization | The blacklist and whitelist specific to API authorization is non-asset data.                                                                             | Reconfigure blacklist and whitelist information after assets are imported.                                                                                                                                                                                                                                                                                              |
| Device integration | Device information                                    | The device information is non-asset data.                                                                                                                | Re-create devices after assets are imported.                                                                                                                                                                                                                                                                                                                            |
|                    | Device rule information                               | The device rule information is non-asset data.                                                                                                           | Re-create device rules after assets are imported.                                                                                                                                                                                                                                                                                                                       |

## Restoration/Migration Process

The following table describes the restoration/migration process of integration assets.

**Table 11-13** Service process

| No. | Operation                                | Description                                                                                                                                                                   |
|-----|------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | <b>Preparations</b>                      | Obtain an integration asset package and collect information about interconnection with peripheral systems (such as data sources, backend services, and service applications). |
| 2   | <b>Modifying Instance Configurations</b> | Modify the security group and configuration parameters of the instance you want to restore or a new instance.                                                                 |

| No. | Operation                                                  | Description                                                                                     |
|-----|------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| 3   | <b>Importing Integration Assets</b>                        | Import the integration asset package to the instance you want to restore or a new instance.     |
| 4   | <b>Modifying Integration Application Configurations</b>    | Modify the authorization configurations of integration applications.                            |
| 5   | <b>Modifying Data Source Configurations</b>                | Modify the connection configuration of data sources.                                            |
| 6   | <b>Modifying API Configurations</b>                        | Modify the environment, control policy, secret, and blacklist/whitelist configurations of APIs. |
| 7   | <b>Modifying Device Configurations</b>                     | Create devices and rules in batches.                                                            |
| 8   | <b>Service Interconnection Adaptation and Verification</b> | Interconnect with peripheral systems and verify services.                                       |

## 11.2.2 Preparations

Before you restore or migrate integration assets, make the following preparations:

- Prepare an integration asset package.  
Obtain the integration asset package you want to import.
- Obtain configuration information of the source instance to which an asset package belongs. This information is required only in asset migration scenarios.

The configuration information includes the security group configuration, instance configuration parameters, and data dictionary information of instances.

- Obtain interconnection information of peripheral systems. The information is required only in asset migration scenarios.

| System              | Item                                                           | Purpose                                                     |
|---------------------|----------------------------------------------------------------|-------------------------------------------------------------|
| Data source         | IP address, username, and password for accessing a data source | Used for ROMA Connect to connect to the data source         |
| Service application | Network segment of a service application                       | Used for ROMA Connect to connect to the service application |

## 11.2.3 Modifying Instance Configurations

### Overview

Before you import integration assets, modify the parameter configurations of the instance to which you want to import integration assets, including the security group configuration, instance configuration parameters, and data dictionaries.

In asset migration scenarios, the instance configuration parameters must be the same as those of the source instance to which the asset package belongs. Otherwise, the assets cannot be imported.

### Procedure

1. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
2. Modify the security group configuration. This step is performed only in asset migration scenarios. Otherwise, skip this step.
  - a. On the **Basic Information** tab page of the **Instance Information** page, click the security group name.
  - b. On the security group configuration page, modify the security group rule based on the source instance information obtained in [Preparations](#) and the actual networking requirements of the new instance.  
For details about the security group rule of ROMA Connect instances, see [Table 4-2](#).
3. Modify instance configuration parameters. This step is performed only in asset migration scenarios. Otherwise, skip this step.  
On the **Instance Information** page, click the **Configuration Parameters** tab and modify the configuration parameters based on the source instance information obtained in [Preparations](#).
4. Add data dictionaries.  
On the **Instance Information** page, click the **Data Dictionaries** tab and add data dictionaries based on the source instance information obtained in [Preparations](#).  
For details about how to create a data dictionary, see [Creating a Data Dictionary](#).

## 11.2.4 Importing Integration Assets

### Overview

You can use the asset import function to quickly import an integration asset package to the ROMA Connect instance.

### Prerequisites

- You have obtained an integration asset package.
- In asset migration scenarios, you have modified instance configuration parameters. For details, see [3](#).



## Procedure

1. On the **Instances** page of the ROMA Connect console, click **Import Asset** in the upper right corner.
2. In the **Import Asset** dialog box, select **Append** or **Overwrite** based on the site requirements.
  - In asset restoration scenarios, select **Overwrite**.
  - In asset migration scenarios, select **Append**.
3. In the dialog box displayed, select a local asset file and import it.
4. After the import is complete, click **Close**. You can view the imported assets in the instance.

## 11.2.5 Modifying Integration Application Configurations

### Overview

User authorization information of integration applications is non-asset data, which must be reconfigured after you import integration assets.

- In asset restoration scenarios, you need to configure the imported integration applications.
- In asset migration scenarios, you can configure the integration applications based on the site requirements.

### Prerequisites

You have imported integration assets. For details, see [Importing Integration Assets](#).

### Procedure

1. On the **Integration Applications** page of the ROMA Connect console, click **Authorize User** on the right of the target integration application.
2. In the **Authorize User** dialog box, configure authorization information for the application.  
For details about how to configure authorization information, see [Configuring Integration Application Authorization](#).
3. Click **OK**.

## 11.2.6 Modifying Data Source Configurations

### Overview

Data source connection information includes the connection address, account, and password, which is non-asset data. After you complete importing integration assets, you must reconfigure the data source connection information.

### Prerequisites

You have imported integration assets. For details, see [Importing Integration Assets](#).

## Procedure

1. On the **Data Sources** page of the ROMA Connect console, click **Edit** on the right of the target data source.
2. On the **Edit Data Source** page, modify connection information about the data source.
  - In asset restoration scenarios, change the password.
  - In asset migration scenarios, change the IP address, username, and password of the data source.

For details about the data source parameters, see [Connecting to Data Sources](#).

3. After the modification is complete, click **Check Connectivity** to verify the connectivity between ROMA Connect and the data source.
4. Click **Save**.

## 11.2.7 Modifying API Configurations

### Overview

The environment information, control policy information, secret information, blacklist and whitelist for API authorization, and APIs that are not published in the RELEASE environment are not asset data. You need to reconfigure this information after importing integration assets.

- Environment information includes the independent domain names, environment variables, cloud server information of load balance channels, and non-default environments.
- Control policy information includes request throttling policies, access control policies, client access control policies, and client quota policies.
- Secret information includes clients' AppCodes and custom backends' configurations of the password or certificate type.

### Prerequisites

You have imported integration assets. For details, see [Importing Integration Assets](#).

### Configuring Environment Information

1. Bind an independent domain name.
  - a. Log in to the ROMA Connect console, choose **API Connect > API Management**, and click the **API Groups** tab.
  - b. Choose **More > Manage Domain Name** on the right of the target API group.
  - c. On the **Domain Names** tab page, click **Bind Independent Domain Name** to bind an independent domain name to the API group.

For details about the parameters, see [Binding Domain Names](#).
  - d. Click **OK**.
2. Create an environment.

- a. On the **API Management** page, click the **Environments** tab.
  - b. Click **Create** and create an environment for publishing APIs.
  - c. Click **OK**.
3. Add environment variables.
    - a. On the **API Management** page, click the **API Groups** tab.
    - b. Choose **More > Manage Variable** on the right of the target API group.
    - c. Select the environment to which you want to add an environment variable and click **Create Variable**.  
For details about the parameters, see [Creating an Environment Variable](#).
    - d. Click **OK**.
  4. Modify a load balance channel member.
    - a. On the **API Management** page, click the **Load Balance Channels** tab.
    - b. Click **Edit** on the right of the target load balance channel.
    - c. On the **Edit Load Balance Channel** page, click **Next**.
      - For a load balance channel whose **Member Type** is **Instance**, click **Select ECS** to add load balance channel members.
      - For a load balance channel whose **Member Type** is **IP address**, change the backend service address based on the backend server in use.For details about the parameters, see [\(Optional\) Creating a Load Balance Channel](#).
    - d. After the configuration is complete, click **Finish**.

## Configuring Control Policy Information

1. Configure a request throttling policy.
  - a. On the **API Management** page, click the **Request Throttling Policies** tab.
  - b. Click **Create** and add a request throttling policy.  
For details about the parameters, see [Creating a Request Throttling Policy](#).
  - c. Click **OK**.
  - d. Bind the request throttling policy to the API, application, or tenant by following the procedure described in [Configuring a Request Throttling Policy](#).
2. Configure an access control policy.
  - a. On the **API Management** page, click the **Access Control Policies** tab.
  - b. Click **Create** and add an access control policy.  
For details about the parameters, see [Creating an Access Control Policy](#).
  - c. Click **OK**.
  - d. Bind an access control policy to the API by following the procedure described in [Configuring an Access Control Policy](#).

3. Configure an access control policy for a client.
  - a. On the **API Calling** page, click the **Clients** tab.
  - b. Click **Set Access Control** on the right of the target client and set the access control policy for the client.

For details about the parameters, see [Configuring an Access Control Policy](#).
  - c. Click **OK**.
4. Configure a quota policy for a client.
  - a. On the **API Calling** page, click the **Client Quota Policies** tab.
  - b. Click **Create** and add a client quota policy.

For details about the parameters, see [Creating a Client Quota Policy](#).
  - c. Click **OK**.
  - d. Bind the quota policy to a client by following the procedure described in [Configuring a Client Quota Policy](#).

## Configuring Secret Information

1. Add an AppCode for a client.
  - a. On the **API Calling** page, click the **Clients** tab.
  - b. Click the client name to access the client details page.
  - c. On the **AppCode** tab page, click **Add AppCode** to add an AppCode for the client.
  - d. Click **OK**.
2. Add the configurations of the password or certificate type for custom backends.
  - a. On the **Custom Backend** page, click the **Configurations** tab.
  - b. Click **Edit** on the right of the target configuration.
    - For configuration of the password type, change the password.
    - For configuration of the certificate type, modify the certificate content and password.
  - c. Click **OK**.

## Modifying Blacklist and Whitelist Configurations of API Authorization

Only APIs that are accessed using App authentication support authorization configuration.

1. On the **API Management** page, click the **APIs** tab.
2. Click the name of the target API to access the API details page.
3. On the **Authorization** tab page, choose **More > Edit Green Channel** on the right of the authorized integration application and modify the blacklist and whitelist configurations as required.
4. Click **OK**.

## Creating an API and Publishing It in a Non-RELEASE Environment

1. On the **API Management** page, click the **APIs** tab.
2. Click **Create API** and create an API as required.  
For details about the parameters, see [Creating an API](#).
3. Publish the API to a non-RELEASE environment by following the procedure described in [Publishing an API](#).

## 11.2.8 Modifying Device Configurations

### Overview

Device information and device rule information is non-asset data. You need to reconfigure this information after importing integration assets. You can use the import function to restore or create devices and rules in batches.

### Prerequisites

You have imported integration assets. For details, see [Importing Integration Assets](#). Ensure that integration applications and products to which devices and rules belong have been successfully imported.

### Procedure

1. (Optional) Prepare one device file and one rule file.  
Obtain the exported device file and rule file from the source instance to which the asset package belongs.
2. Create a device.
  - a. On the ROMA Connect console, choose **LINK > Device Management**.
    - If you have obtained a device file, click **Import** to create a device by importing the file.
    - If you have not obtained a device file, click **Create Device** to manually create a device.  
For details about the parameters, see [Creating a Device](#).
  - b. After the import is successful, you can view the imported device in the device list.
3. Create a rule.
  - a. On the ROMA Connect console, choose **LINK > Rule Engine**.
    - If you have obtained a rule file, click **Import** to create a rule by importing the file.
    - If you have not obtained a rule file, click **Create Rule** to manually create a rule.  
For details about the parameters, see [Creating a Rule](#).
  - b. After the import is successful, you can view the imported rule in the rule list.

## 11.2.9 Service Interconnection Adaptation and Verification

### Overview

After you have restored or migrated integration assets in the ROMA Connect instance, notify the service applications interworking with ROMA Connect to update their interconnection configurations so that they can process services properly.

### Service Interconnection Adaptation

The following table describes the interconnection configurations to update on the service applications or devices connected to ROMA Connect.

| Category            | Service Scenario                           | Configuration Data                                                                                                                                                                                                                                                                                 |
|---------------------|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Service integration | Calling an API                             | <ul style="list-style-type: none"> <li>API calling address (required only in asset migration scenarios)</li> <li>API authentication information, including the keys, secrets, AppCodes, and AK/SK of integration applications/clients</li> <li>Key and secret of backend signature keys</li> </ul> |
| Message integration | Sending and receiving messages             | <ul style="list-style-type: none"> <li>Topic connection address (required only in asset migration scenarios)</li> <li>Key and secret used when SASL authentication is enabled.</li> </ul>                                                                                                          |
| Device integration  | Reporting messages and delivering commands | <ul style="list-style-type: none"> <li>Device connection address (required only in asset migration scenarios)</li> <li>Access authentication information of devices, including client IDs, usernames, and passwords</li> </ul>                                                                     |

### Service Verification

After you update the interconnection configurations, check whether service applications can process services properly.

- Check whether a service application can call APIs.
- Check whether a service application can send messages to and receive messages from topics of ROMA Connect.

- Check whether a device can send data to and receive data from ROMA Connect and whether the rule engine can forward data.

# 12 FAQs

- [Common Operations](#)
- [Data Integration Tasks](#)
- [Composite Data Integration Tasks](#)
- [Service Integration](#)
- [Message Integration](#)
- [Device Integration](#)

## 12.1 Common Operations

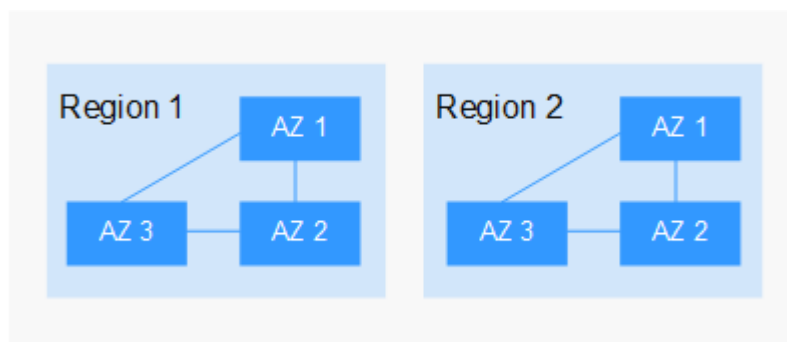
### 12.1.1 Region and AZ

#### Definition

A region and availability zone (AZ) identify the location of a data center. You can create resources in a specific region and AZ.

**Figure 12-1** shows the relationship between regions and AZs.

**Figure 12-1** Regions and AZs





## Selecting a Region

### Selecting an AZ

When deploying resources, consider your applications' requirements on disaster recovery (DR) and network latency.

- For high DR capability, deploy resources in different AZs within the same region.
- For low network latency, deploy resources in the same AZ.

## 12.1.2 Can I Use an Existing VPC to Create an ROMA Connect Instance?

You can use an existing VPC to create an ROMA Connect instance. Before creating an ROMA Connect instance, ensure that your VPC and ROMA Connect instance are in the same AZ and that the subnet and security group have been configured.

## 12.1.3 Does ROMA Connect Have Quota Limits?

[Table 12-1](#) describes the usage restrictions of each component of ROMA Connect.

**Table 12-1** Resource quota limitations

| Component               | Resource                                              | Maximum Quota | Modifiable |
|-------------------------|-------------------------------------------------------|---------------|------------|
| Integration application | Number of integration applications                    | 100           | No         |
| Data source             | Number of data sources                                | 500           | No         |
| APIC                    | Number of APIs                                        | 5000          | Yes        |
|                         | Number of API groups                                  | 1500          | Yes        |
|                         | Number of APIs in a single API group                  |               | Yes        |
|                         | Number of environment variables in a single API group | 50            | Yes        |
|                         | Number of request throttling requests                 |               | Yes        |
|                         | Number of access control policies                     |               | Yes        |
|                         | Number of environments                                | 10            | Yes        |

| Component | Resource                                 | Maximum Quota | Modifiable |
|-----------|------------------------------------------|---------------|------------|
|           | Number of signature keys                 | 200           | Yes        |
|           | Number of load balance channels          | 200           | Yes        |
|           | Number of ECSs in a load balance channel | 10            | Yes        |
|           | Number of custom authorizers             | 50            | Yes        |
|           | Number of backend APIs                   | 500           | Yes        |
|           | Number of client quota policies          | 20000         | Yes        |
| LINK      | Number of product templates              | 100           | Yes        |
|           | Number of products                       | 500           | Yes        |
|           | Number of devices                        | 1000          | Yes        |
|           | Number of rules                          | 2000          | No         |

## 12.2 Data Integration Tasks

### 12.2.1 Which Types of Data Are Supported by FDI Databases?

FDI databases support the following types of data:

- Kafka: string
- MySQL: integer, bigint, number, varchar, char, float, date, timestamp, and datetime
- Oracle: integer, bigint, number, varchar, char, float, date, and timestamp
- SQL Server: integer, bigint, varchar, char, float, date, datetime, and timestamp (Timestamp data is stored in binary format in the SQL Server database.)
- PostgreSQL: integer, number, varchar, char, float, date, and timestamp
- MongoDB: integer, double, string, boolean, date, long, object, and array
- DIS: integer, JSON, arraystring, boolean, long, double, bigdecimal, and date
- MRS Hive: integer, bigint, string, float, timestamp, varchar, char, and date

- MRS HBase: string

 NOTE

You are not advised to use varchar, char, and date for the MRS Hive database. If the data source length exceeds the limit, FDI cuts off the excessive the length, which may cause incomplete information.

## 12.2.2 What Is a Trace Number? Can Data Be Traced by Using a Trace Number?

A trace number identifies a task scheduling. It can be used by a backend to trace task status, but not tracing data.

## 12.2.3 Does an FDI Task Support Clearance of a Destination Table?

In FDI, a destination table can be cleared only if the DWS data source is selected as the destination during task creation. Other data sources do not support table clearance.

## 12.2.4 Can a Task Collect Data from One Table to Another Table?

A common task can collect data only from a single table at the source to a single table at the destination. A composite task can collect data from multiple tables at the source to multiple tables at the destination, at a time.

## 12.2.5 Is the FDI Task Created by One User Visible to Another User Under the Same Account?

FDI tasks are isolated by user. If permissions are not granted to a user, the user cannot view the FDI tasks created by another user under the same account. If a user wants to view the FDI tasks of other users, the user must be granted permissions through integration application authorization.

## 12.2.6 How Do I Configure FDI to Connect to MRS Hive of Other Tenants Through a Public Network?

Bind all MRS IP addresses to EIPs, and replace all private IP addresses in the MRS configuration file with EIPs. Then, add the `dfs.client.use.datanode.hostname` configuration item to the `hdfs-site.xml` file and set its value to `true`.

## 12.2.7 Will Collected Data Be Updated After a File Is Parsed from OBS to the RDS Database?

If the same file is collected and the destination database has a primary key, FDI updates data based on the primary key each time the file is updated on OBS.

If multiple files are collected and the destination database has same primary keys, FDI updates data with the value in the earliest file each time. By default, FDI obtains files from OBS based on the timestamp in descending order. The file

uploaded at the latest time is parsed first, and the file uploaded at the earliest time is parsed last. As a result, after the task execution is complete, the value in the database is still the value in the earliest uploaded file.

## 12.2.8 Why Are the Column Values Incorrect When the CSV File on the Source OBS Is Parsed to the Relational Database?

This occurs during task configuration if the sequence of filling in metadata of the source OBS is different from that of the columns in the CSV file. Therefore, modify the task configuration and fill in the metadata based on the sequence of the columns in the CSV file.

## 12.2.9 Can Data Be Integrated into the Destination When the Data Types of Source and Destination Fields of MRS Hive Do Not Match?

If the field data type of MRS Hive at the destination does not match that of the source, for example, the data type of a source field is **datetime** and the data type of a destination field is **date**, data can be integrated to the destination, but the value of the field whose data type does not match will be **NULL**. Other fields are properly integrated.

## 12.2.10 Is the Mapping Between MRS Hive or MRS HBase and MongoDB Case-sensitive When It Is Manually Entered?

The mapping is case-sensitive. The capitalization rule of source and destination fields must be the same as that of the selected fields. If they are different, the MongoDB database will not be impacted, but MRS Hive and MRS HBase will run abnormally.

## 12.2.11 Does MRS Hive Support Partitioning?

MRS Hive supports partitioning.

When entering mapping information, you can comply with the following format:

```
{
  "partition_col":{
    "origin_col": "Partition field name of the source table",
    "target_col": "Partition field name of the destination"
  },
  "col_seq":{
    "index_0": "Source table field 1",
    "index_1": "Source table field 2",
    "index_2": "Source table field 3"
  }
}
```

Partitioning is optional for MRS Hive.

If there is no partition, comply with the following format when entering mapping information:

```
{
  "col_seq":{
    "index_0": "Source table field 1",
```

```
"index_1": "Source table field 2",
"index_2": "Source table field 3"
}
}
```

## 12.2.12 How Can I Configure the Time for Triggering a Scheduled Task of FDI?

You can configure the time for triggering a scheduled task by using a Quartz Cron expression. A Quartz Cron expression is a string comprised of six fields separated by space characters. The six fields in the expression are described as follows:  
Seconds Minutes Hours Day (day in each month) Month Week

**Table 12-2** describes the meanings of each field in the Quartz Cron expression.

**Table 12-2** Fields in the Quartz Cron expression

| Name                    | Mandatory | Optional Values                                                                                                 | Optional Special Characters |
|-------------------------|-----------|-----------------------------------------------------------------------------------------------------------------|-----------------------------|
| Seconds                 | Yes       | 0-59                                                                                                            | , - * /                     |
| Minutes                 | Yes       | 0-59                                                                                                            | , - * /                     |
| Hours                   | Yes       | 0-23                                                                                                            | , - * /                     |
| Day (day in each month) | Yes       | 1-31. Note that there are no 30 and 31 in February, and there is no 31 in April, June, September, and November. | , - * ? / L W C             |
| Month                   | Yes       | 1-12 or a character string                                                                                      | , - * /                     |
| Week                    | Yes       | 1-7 or a character string                                                                                       | , - * ? / L C #             |

The month in a cron expression can be a number or character string. If you want to set the month to January, either **1** or **JAN** complies with the Quartz Cron expression. For details about the mapping between numbers and character strings, see **Table 12-3**.

**Table 12-3** Mapping between numbers and character strings for the month

| Numeral | String |
|---------|--------|
| 1       | JAN    |
| 2       | FEB    |
| 3       | MAR    |
| 4       | APR    |
| 5       | MAY    |
| 6       | JUN    |

| Numeral | String |
|---------|--------|
| 7       | JUL    |
| 8       | AUG    |
| 9       | SEP    |
| 10      | OCT    |
| 11      | NOV    |
| 12      | DEC    |

The week in a Quartz Cron expression can be a number or character string. If you want to set the day to Monday, either **2** or **MON** complies with the Quartz Cron expression. For details about the mapping between numbers and character strings, see [Table 12-4](#). The Quartz Cron expression defines Saturday as the last day of each week.

**Table 12-4** Mapping between numbers and character strings for the week

| Numeral | String | Description |
|---------|--------|-------------|
| 1       | SUN    | Sunday      |
| 2       | MON    | Monday      |
| 3       | TUE    | Tuesday     |
| 4       | WED    | Wednesday   |
| 5       | THU    | Thursday    |
| 6       | FRI    | Friday      |
| 7       | SAT    | Saturday    |

## Precautions

Besides numbers in each part, the following special characters can also be displayed:

1. \*: It can match any value of a field. If \* is used in the minutes field, an event is triggered every minute.
2. ?: It can only be used in two fields: day (day in each month) and week. It can also match any value of the field, but actually it does not. This is because day (day in each month) and week are mutually affected. For example, if you want to trigger scheduling on the 20th day of each month, you can use the following expression: 13 13 15 20 \* ?, where the last digit can only be ?, but not \*. If you use \*, an event will be triggered regardless of the day of week.
3. -: It specifies a range. For example, **5-20** in the minutes field means that an event is triggered every minute from minutes 5-20.

4. `/`: An event is triggered from the specified time and at a fixed interval. For example, if you set the minutes field to **5/20**, it means that the scheduled task will be triggered at the 5th, 25th, and 45th minutes of each hour.
5. `;`: It specifies a list value. For example, if you set the minutes field to **5,20**, it means that the scheduled task will be triggered every minute between minutes 5 and 20 of each hour.
6. `L`: It indicates the last. It can appear only in the day (day in each month) and week fields. For example, **5L** in the day (day in each month) field means that the event is triggered on the last Thursday.
7. `W`: It indicates a weekday (Monday to Friday) nearest the given day. It can appear only in the day (day of month) field. An event will be triggered on the valid weekday closest to the specified date. For example, **5W** in the day (day of month) field means that an event will be triggered on the nearest weekday to the 5th day of the month. If the 5th day is Saturday, an event will be triggered on the 4th day (Friday). If the 5th day is Sunday, an event is triggered on the 6th day (Monday). If the 5th day is one day from Monday to Friday, an event is triggered on the 5th day. The trigger will not jump over the boundary of a month's days.
8. `LW`: The two characters indicate the last weekday of a month, that is, the last Friday.
9. `#`: It indicates the nth day of a week in each month. It can appear only in the day (day of week) field. For example, **4#2** indicates the second Wednesday of a month.

## Examples

The following examples help you better understand the meanings of the Quartz Cron expression. The simple examples show the primary use of Quartz Cron expressions. The common examples show how to use Quartz Cron expressions in actual scenarios. You only need to simply modify common examples based on site requirements.

### Simple examples:

```
0/5 * * * * ? //Executed every five seconds
0 0/1 * * * * ? //Executed every one minute
0 0 23 * * ? //Executed at 23:00 every day
0 0 1 1 * ? //Executed at 01:00 on the first day of each month
0 0 23 L * ? //Executed at 23:00 on the last day of each month
0 33,55 * * * ? //Executed at the 33th and 55th minute
0 0 18 * * LW //Executed at 06:00 on the last weekday of each month
```

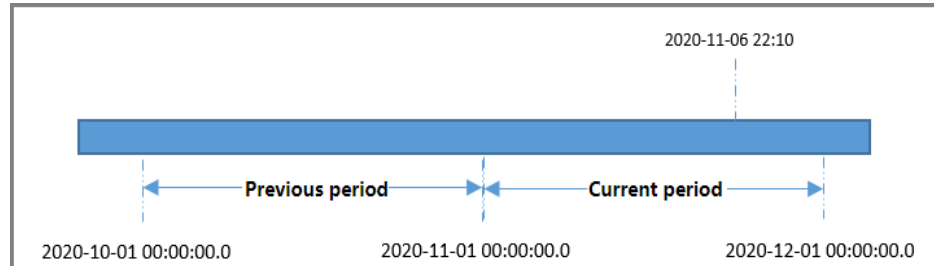
### Common examples:

```
0 0 12 * * ? //Executed at 12:00 every day
0 * 14 * * ? //Executed every minute from 02:00 to 02:59 every day
0 0/30 9-17 * * ? //Executed every half an hour within working hours from 09:00 to 17:00
0 0 12 ? * WED //Executed at 12:00 every Wednesday
0 0/30 9-17 * * MON-FRI //Executed every half an hour from 09:00 to 17:00 every day from Monday to Friday
0 15 10 ? * MON-FRI //Executed at 10:15 every day from Monday to Friday
0 0 2 1 * ? //Executed at 02:00 on the first day of each month
0 15 10 ? * 6#3 //Executed at 10:15 on the third Friday of each month
0 15 10 ? * 6L //Executed at 10:15 on the last Friday of each month
```

## 12.2.13 How Do I Set the Custom Period for the API Data Source at the Source?

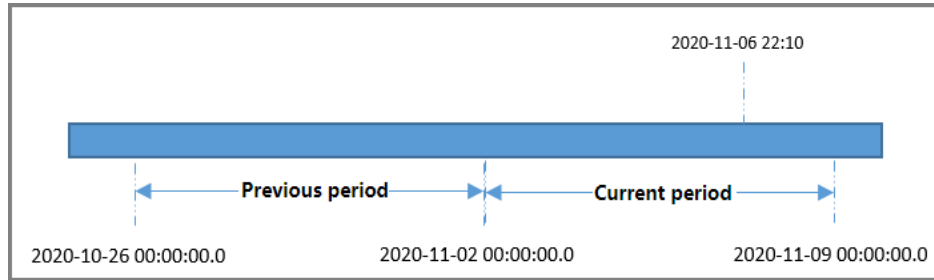
During a common FDI task creation, if you select API as the data source type and enable **Incremental Migration** at the source, you can customize the period. The following examples describe how to configure the period.

- If **Time Interval** is set to **Hour**, the configuration scenarios are as follows:  
2020-11-05 22:10 is used as an example.

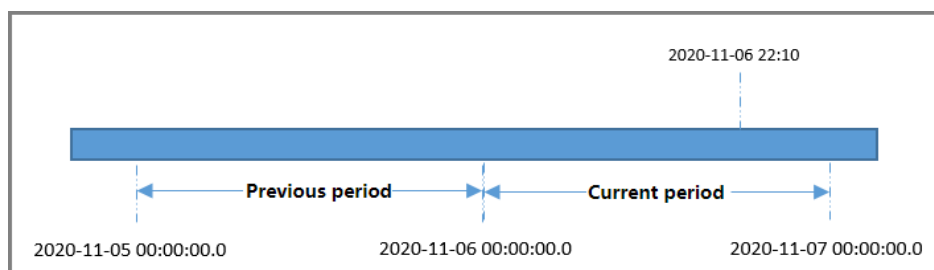


- Period** is set to **Current period** and **Right Periodic Boundary** is set to **Open interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and end time of the current period, respectively.  
The time interval for obtaining data is [2020-11-05 22:00:00.0, 2020-11-05 23:00:00.0].
  - Period** is set to **Current period** and **Right Periodic Boundary** is set to **Closed interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the current period, respectively.  
The time interval for obtaining data is [2020-11-05 22:00:00.0, 2020-11-05 22:59:59.999].
  - Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Open interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and end time of the previous period, respectively.  
The time interval for obtaining data is [2020-11-05 21:00:00.0, 2020-11-05 22:00:00.0].
  - Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Closed interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the previous period, respectively.  
The time interval for obtaining data is [2020-11-05 21:00:00.0, 2020-11-05 21:59:59.999].
- If **Time Interval** is set to **Day**, the following configuration scenarios are available:  
2020-11-06 22:10 is used as an example.





- a. **Period** is set to **Current period** and **Right Periodic Boundary** is set to **Open interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and end time of the current period, respectively.  
The time interval for obtaining data is [2020-11-06 00:00:00.0, 2020-11-07 00:00:00.0].
  - b. **Period** is set to **Current period** and **Right Periodic Boundary** is set to **Closed interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the current period, respectively.  
The time interval for obtaining data is [2020-11-06 00:00:00.0, 2020-11-06 23:59:59.999].
  - c. **Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Open interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and end time of the previous period, respectively.  
The time interval for obtaining data is [2020-11-05 00:00:00.0, 2020-11-06 00:00:00.0].
  - d. **Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Closed interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the previous period, respectively.  
The time interval for obtaining data is [2020-11-05 00:00:00.0, 2020-11-05 23:59:59.999].
- If **Time Interval** is set to **Week**, the following configuration scenarios are available:  
2020-11-06 22:10 is used as an example.



- a. **Period** is set to **Current period** and **Right Periodic Boundary** is set to **Open interval**.  
The values of **Start Time Field** and **End Time Field** are the start time and end time of the current period, respectively.

The time interval for obtaining data is [2020-11-02 00:00:00.0, 2020-11-09 00:00:00.0].

- b. **Period** is set to **Current period** and **Right Periodic Boundary** is set to **Closed interval**.

The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the current period, respectively.

The time interval for obtaining data is [2020-11-02 00:00:00.0, 2020-11-08 23:59:59.999].

- c. **Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Open interval**.

The values of **Start Time Field** and **End Time Field** are the start time and end time of the previous period, respectively.

The time interval for obtaining data is [2020-10-26 00:00:00.0, 2020-11-02 00:00:00.0].

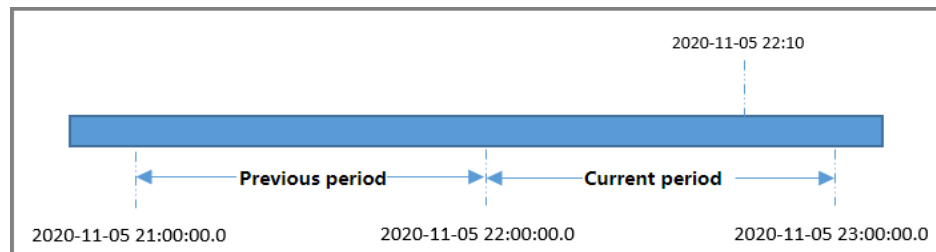
- d. **Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Closed interval**.

The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the previous period, respectively.

The time interval for obtaining data is [2020-10-26 00:00:00.0, 2020-11-01 23:59:59.999].

- If **Time Interval** is set to **Month**, the following configuration scenarios are available:

2020-11-06 22:10 is used as an example.



- a. **Period** is set to **Current period** and **Right Periodic Boundary** is set to **Open interval**.

The values of **Start Time Field** and **End Time Field** are the start time and end time of the current period, respectively.

The time interval for obtaining data is [2020-11-01 00:00:00.0, 2020-12-01 00:00:00.0].

- b. **Period** is set to **Current period** and **Right Periodic Boundary** is set to **Closed interval**.

The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the current period, respectively.

The time interval for obtaining data is [2020-11-01 00:00:00.0, 2020-11-30 23:59:59.999].

- c. **Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Open interval**.

The values of **Start Time Field** and **End Time Field** are the start time and end time of the previous period, respectively.

The time interval for obtaining data is [2020-10-01 00:00:00.0, 2020-11-01 00:00:00.0].

- d. **Period** is set to **Previous period** and **Right Periodic Boundary** is set to **Closed interval**.

The values of **Start Time Field** and **End Time Field** are the start time and the end time subtracted by 1 ms of the previous period, respectively.

The time interval for obtaining data is [2020-10-01 00:00:00.0, 2020-10-31 23:59:59.999].

## 12.3 Composite Data Integration Tasks

### 12.3.1 How Do Source Database Restarts Affect Composite Tasks?

During service usage, the operations such as capacity expansion and restart may be required for databases such as Oracle. The impacts of these operations on composite tasks are as follows:

- If the database takes a short time to restart, the system retries a database connection and attempts to restore a task.
- If the database takes a long time to restart during a composite task, an error will be reported. After the database is restarted, you must restart the composite task.

### 12.3.2 How Do I Handle an Exception of the CDC Function (Using the XStream Mode) Caused by Unexpected Oracle Database Restart?

If the CDC function is abnormal because the Oracle database restarts unexpectedly, perform the following operations on the Oracle database to locate and rectify the fault:

Connect to the Oracle database as user **sys** and run the following command:

```
SELECT CAPTURE_NAME,status FROM ALL_CAPTURE;
```

If the status is **Aborted**, the outbound server of the CDC is abnormal. Run the following command and press **Enter**. Then, enter **/** and press **Enter** again to manually start the outbound server.

```
BEGIN  
DBMS_XSTREAM_ADM.START_OUTBOUND(server_name => 'dbzxout');  
END;
```

## 12.4 Service Integration

## 12.4.1 Which Languages Does APIC Support for SDKs?

APIC supports Java, Go, Python, C#, JavaScript, PHP, C++, C, Android, and curl SDKs. If other programming languages are used, sign and encapsulate HTTP requests by referring to .

## 12.4.2 Does APIC Support Multiple Backend Endpoints?

Yes, APIC supports the configuration of multiple backend endpoints through a load balance channel. You can add multiple Elastic Cloud Servers (ECSs) to a load balance channel.

## 12.4.3 Which Error Codes Will Be Displayed When I Use APIC?

[Table 12-5](#) lists the error codes that may occur when you call APIs.

**Table 12-5** Error codes

| HTTP Status Code | Error Code | Description                                                          | Solution                                                                                                                                                                                                                                                                                                                                 |
|------------------|------------|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 404              | APIC.0101  | The API does not exist or has not been published in the environment. | <ul style="list-style-type: none"> <li>Check whether the domain name, method, and path are consistent with those of the registered API.</li> <li>Check whether the API is published. If the API is published in a non-production environment, check whether the X-Stage header in the request is the name of the environment.</li> </ul> |
| 500              | APIC.0103  | The backend does not exist.                                          | Contact technical support.                                                                                                                                                                                                                                                                                                               |
| 500              | APIC.0104  | The plug-ins do not exist.                                           | Contact technical support.                                                                                                                                                                                                                                                                                                               |
| 500              | APIC.0105  | The backend configurations do not exist.                             | Contact technical support.                                                                                                                                                                                                                                                                                                               |
| 400              | APIC.0106  | Orchestration error.                                                 | Check whether the frontend and backend parameters of the API are correct.                                                                                                                                                                                                                                                                |
| 400              | APIC.0201  | Bad request.                                                         | Set valid request parameters.                                                                                                                                                                                                                                                                                                            |
| 413              | APIC.0201  | Request entity too large.                                            | Reduce the size of the request body to less than 12 MB.                                                                                                                                                                                                                                                                                  |
| 414              | APIC.0201  | Request URI too large.                                               | Reduce the size of the URI to less than 32 KB.                                                                                                                                                                                                                                                                                           |

| HTTP Status Code | Error Code | Description                                       | Solution                                                                                                                                                                                                                          |
|------------------|------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 494              | APIC.0201  | Request headers too large.                        | Reduce the size of request headers to ensure that the length of a single request header is less than 32 KB or the total length of all request headers is less than 128 KB.                                                        |
| 502              | APIC.0202  | Backend unavailable.                              | Check whether the backend address configured for the API is accessible.                                                                                                                                                           |
| 504              | APIC.0203  | Backend timeout.                                  | Increase the timeout duration of the backend service or shorten the processing time.                                                                                                                                              |
| 401              | APIC.0301  | Incorrect IAM authentication information.         | Check whether the token is correct.                                                                                                                                                                                               |
| 403              | APIC.0302  | The IAM user is not authorized to access the API. | Check whether the user is restricted by a blacklist or whitelist.                                                                                                                                                                 |
| 401              | APIC.0303  | Incorrect App authentication information.         | <ul style="list-style-type: none"> <li>• Check whether the request method, path, query parameters, and request body are consistent with those used for signature.</li> <li>• Check whether the client time is correct.</li> </ul> |
| 403              | APIC.0304  | The app is not authorized to access the API.      | Check whether the app has been authorized to access the API.                                                                                                                                                                      |
| 401              | APIC.0305  | Incorrect authentication information.             | Check whether the authentication information is correct.                                                                                                                                                                          |
| 403              | APIC.0306  | API access denied.                                | Check whether you have been authorized to access the API.                                                                                                                                                                         |
| 401              | APIC.0307  | The token must be updated.                        | Update the token.                                                                                                                                                                                                                 |
| 429              | APIC.0308  | The throttling threshold has been reached.        | Refresh the request throttling policy and try again.                                                                                                                                                                              |
| 403              | APIC.0401  | Unknown client IP address.                        | Contact technical support.                                                                                                                                                                                                        |

| HTTP Status Code | Error Code | Description                                         | Solution                                                                                                                  |
|------------------|------------|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| 403              | APIC.0402  | The IP address is not authorized to access the API. | Check whether the IP address is restricted by a blacklist or whitelist.                                                   |
| 503              | APIC.0404  | Access to the backend IP address has been denied.   | Use an available IP address to access the backend service.                                                                |
| 403              | APIC.0405  | The app is not accessed from a trusted IP address.  | Check whether the IP address is restricted by the client access control policy.                                           |
| 500              | APIC.0601  | Internal server error.                              | Contact technical support.                                                                                                |
| 400              | APIC.0602  | Bad request.                                        | Check whether the request is valid.                                                                                       |
| 500              | APIC.0605  | Backend domain name resolution failed.              | Check whether the domain name is correct and has been bound to a correct backend address.                                 |
| 500              | APIC.0606  | Failed to load the API configurations.              | Contact technical support.                                                                                                |
| 400              | APIC.0607  | The following protocol is supported: {xxx}          | Use HTTP or HTTPS to access the API.                                                                                      |
| 500              | APIC.0608  | Failed to obtain the admin token.                   | Contact technical support.                                                                                                |
| 500              | APIC.0609  | The VPC backend does not exist.                     | Contact technical support.                                                                                                |
| 502              | APIC.0610  | No backend available.                               | Check whether all backends are available.                                                                                 |
| 500              | APIC.0611  | The backend port does not exist.                    | Contact technical support.                                                                                                |
| 500              | APIC.0612  | An API cannot call itself.                          | Modify the backend configurations, and ensure that the number of layers the API is recursively called does not exceed 10. |
| 500              | APIC.0705  | Backend signature calculation failed.               | Contact technical support.                                                                                                |

## 12.4.4 How Can I Ensure the Security of Backend Services Invoked by APIC?

You can ensure the security of backend services invoked by APIC by using the following methods:

- Bind signature keys to APIs. After a signature key is bound to an API, APIC adds signature information to each request sent to the backend service. The backend service calculates the signature information in each request and checks whether the signature information is consistent with that on APIC.
- Encrypt requests using HTTPS. Before using this method, ensure that the required SSL certificates are available.

## 12.4.5 Do I Need to Publish an API Again After Modification?

Yes. After you modify the parameters of a published API, you must publish the API again to synchronize the modifications to the environment.

## 12.4.6 What Can I Do If an API Published in a Non-RELEASE Environment Is Inaccessible?

Add the x-stage request header with its value set to the environment name and then access the API again.

## 12.4.7 Can I Invoke Different Backend Services by Publishing an API in Different Environments?

Yes, you can invoke different backend services by publishing an API in different environments while specifying environment variables and backend parameters.

## 12.4.8 What Is the Maximum Size of an API Request Package?

The maximum size of an API request package is 12 MB.

## 12.4.9 Can I Modify an API After It Is Deployed on the Custom Backend?

The deployed API can be modified, but the modification takes effect only after the API is redeployed.

## 12.4.10 How Does APIC Throttle Requests?

You can configure request throttling policies to control the number of APIs that can be accessed in a specified period to protect backend services. .

By default, a single API can be accessed for a maximum of 6000 times per second. If you do not create a request throttling policy, APIC executes the default request throttling policy. On the **Instance Information** page of the ROMA Connect console, you can modify the **ratelimit\_api\_limits** parameter on the **Configuration Parameters** tab page to set the default request throttling policy for APIs.

### 12.4.11 What Are the Causes of an API Calling Failure?

API calling failures are classified into three types: failure of API calling in the same VPC, failure of API calling between different VPCs, and failure of API calling through a public network.

- If the API fails to be called in the VPC, check whether the domain name is the same as that automatically allocated by the API. If the domain name is different from that automatically allocated by the API, the API fails to be called.
- If APIs fail to be called between VPCs, check whether the two VPCs are connected. If the two VPCs are not connected, you can establish VPC peering connections to enable communication between the two VPCs.
- If APIs fail to be called through a public network, the possible causes are as follows:
  - The API is not bound to an EIP. As a result, the API lacks a valid address for public network access, and therefore the API fails to be called through a public network.
  - The inbound rule is incorrectly configured. As a result, the API fails to be called through a public network.
  - The request header "host: Group domain name" is not added during API calling. As a result, the API fails to be called through a public network. The API can be called as long as the message header is added.

### 12.4.12 Can I Call an API If the API Is Not Bound to an EIP?

You can use the domain name automatically allocated by an API to call the API on the intranet. If the VPC in use is the same as the VPC where the API resides, use the domain name automatically allocated by the API to call the API. If the VPC in use is different from the VPC where the API resides,, you can establish VPC peering connections to enable communication between the two VPCs. For details on how to create and use VPC peering connections, see .

### 12.4.13 Which Data Sources Does APIC Support?

[Table 12-6](#) lists the data sources supported by APIC.

**Table 12-6** Data sources supported by data APIs

| Data Source Type | Compatibility                                                                     |
|------------------|-----------------------------------------------------------------------------------|
| DWS              | 1.3.4                                                                             |
| GaussDB 100      | FusionInsight_LibrA_V100R003C20 and FusionInsight_LibrA_V300R001C00               |
| GaussDB 200      | FusionInsight_LibrA_V100R002C70SPC 100,<br>FusionInsight_LibrA_V100R002C80SPC 300 |
| HIVE             | 2.3.2                                                                             |



| Data Source Type | Compatibility                  |
|------------------|--------------------------------|
| MongoDB          | 3.4                            |
| MySQL            | 5.6, 5.7, 8.0                  |
| Oracle           | 11g                            |
| PostgreSQL       | 11.0                           |
| Redis            | 2.8.x, 3.x.x                   |
| SQL Server       | 2008R2, 2012, 2014, 2016, 2017 |

## 12.5 Message Integration

### 12.5.1 Does MQS Support Cross-VPC and Cross-Subnet Access?

MQS supports cross-VPC and cross-subnet access. By establishing a peering connection between two VPCs, ECSs in one VPC can access instances in the other VPC.

### 12.5.2 Do MQS Instances Support Cross-Subnet Access?

Yes.

If the client and the instance are in the same VPC, cross-subnet access is supported.

If the client and instance are located in different VPCs, connect the two VPCs using a VPC peering connection.

You can also access an instance through the public access address bound to the instance.

### 12.5.3 How Do I Select and Configure a Security Group?

To access an MQS instance within a VPC or over public networks, configure the security group rules as follows:

#### Intra-VPC Access

To access a ROMA Connect instance to which MQS belongs, you must deploy your client on an ECS in the same VPC and subnet as the instance. If they are deployed in different VPCs, see [Does MQS Support Cross-VPC and Cross-Subnet Access?](#)

In addition, before you can access the instance through your client, you must configure correct rules for the security groups of both the ECS and ROMA Connect instance to which MQS belongs.

1. You are advised to configure the same security group for the ECS and ROMA Connect instance to which MQS belongs. After a security group is created, the security group rules, by default, include a rule allowing members in the security group to access each other without any restrictions.

- If the ECS and ROMA Connect instance belong to different security groups, add security group rules to ensure that the ECS and ROMA Connect instance can access each other.

 **NOTE**

- Assume that security groups **sg-ecs** and **sg-romaconnect** are configured respectively for your ECS and ROMA Connect instance to which MQS belongs.
- The remote end is a security group or an IP address.

Add the following rule to the security group to which ECS belongs to ensure that the client can access MQS:

**Table 12-7** ECS security group rule

| Direction | Protocol | Port | Destination    |
|-----------|----------|------|----------------|
| Outbound  | All      | All  | sg-romaconnect |

To ensure that your client can access the ROMA Connect instance to which MQS belongs, add the following rule to the security group configured for the instance:

**Table 12-8** MQS security group rule

| Direction | Protocol | Port | Source |
|-----------|----------|------|--------|
| Inbound   | All      | All  | sg-ecs |

## Public Access

Configure security group rules based on [Table 12-9](#) for the client to access the MQS instance.

**Table 12-9** Security group rules for public access

| Direction | Protocol | Port | Source    | Description                                                                  |
|-----------|----------|------|-----------|------------------------------------------------------------------------------|
| Inbound   | TCP      | 9094 | 0.0.0.0/0 | Access MQS through the public network (without SASL authentication).         |
| Inbound   | TCP      | 9095 | 0.0.0.0/0 | Access MQS through the public network (with or without SASL authentication). |
| Inbound   | TCP      | 9096 | 0.0.0.0/0 | Access MQS through the public network (with or without SASL authentication). |

| Direction | Protocol | Port | Source    | Description                                                       |
|-----------|----------|------|-----------|-------------------------------------------------------------------|
| Inbound   | TCP      | 9097 | 0.0.0.0/0 | Access MQS through the public network (with SASL authentication). |

## 12.5.4 Can I Access an MQS Instance Through a Public Network?

MQS instances can be accessed through a public network. Ensure that the public IP address and port are used and public access port 9095 is enabled for the inbound rule of the instance security group.

## 12.5.5 Why Can't I View the Subnet and Security Group Information When Creating an MQS Instance?

The possible cause is that you do not have the Server Administrator and VPC Administrator permissions. For details about how to add the permissions, see .

## 12.6 Device Integration

### 12.6.1 Which Languages Does a LINK SDK Support?

LINK uses the standard MQTT protocol to communicate with devices. You can develop an MQTT SDK based on the site requirements. For details, see the [MQTT official website](#).

### 12.6.2 Which Communication Protocols Are Supported by LINK?

LINK supports the standard MQTT protocol and uses the open-source SDKs to easily connect MQTT-based devices to the cloud.

### 12.6.3 Why Can't My Device Be Connected to LINK?

The possible causes are as follows:

- The related information is not configured.  
Check whether the connection address, connection port, client ID, username, and password are correct.
- The device is disabled.  
Check the device status. A disabled device cannot be connected to LINK. To connect the device to LINK, change the device status from **Disabled** to **Enabled**.
- Public access is not enabled for the instance.  
Enable public access. For details, see .

- No EIP is bound to the instance.  
Check whether an EIP has been bound to the instance. The instance that has not been bound with an EIP cannot be accessed through a public network.
- No port is configured in the inbound rule of the security group.  
Check whether a port is configured in the inbound rule of the security group. Ports that are not configured cannot be accessed.

## 12.6.4 Why Does the Data Destination of a Rule Engine Fail to Receive Messages from My Device After It Is Connected to LINK?

The possible causes are as follows:

- An incorrect inbound rule is added to your security group.  
Click **Instance Information** and locate the target security group. Click the security group name to access its details page. Then, add an inbound rule based on the parameters listed in [Table 12-10](#).

**Table 12-10** Security group parameters

| Parameter                                                   | Port                 |
|-------------------------------------------------------------|----------------------|
| Port for accessing MQS through an intranet                  | 9092                 |
| Port for accessing MQS through an intranet in SSL mode      | 9093                 |
| Port for accessing MQS through a public network             | 9094, 9095, and 9096 |
| Port for accessing MQS through a public network in SSL mode | 9095, 9096, and 9097 |
| Port for accessing MQS through RESTful APIs                 | 9292                 |

- The topic is incorrectly configured.  
You need to use a topic that has permission to publish messages. Check whether your topic has the publish permission. You can find the name of the topic where the fault occurs in the **Topic Name** column, and then check whether the topic has the publish permission in the **Topic Type** column. If a topic does not have the publish permission, the device cannot send messages to the destination of the rule engine.  
To solve this problem, you can configure a topic of the Publish type on the device so that the device can send messages to the destination.
- The first default rule of the security group is deleted.  
When a security group is created, the system automatically generates an inbound rule that allows the destination of the rule engine to receive messages from all devices. If you delete this rule without creating other

inbound rules, the entire security group will lack inbound rules. As a result, the destination of the rule engine cannot receive messages from the device.

You can add inbound rules to solve this problem. If the rule engine is required to receive information from a specific device, you can add an inbound rule and set the port range to the port of the specific device. If the rule engine is required to receive information from all devices, you can add an inbound rule and set the port range to **All**.

- The topic is deleted.

The rule engine needs to forward messages to a specified topic. If the topic is deleted unexpectedly, messages fail to be forwarded to the topic.

To solve this problem, create a topic and configure the topic in the rule engine. When you create the topic, you need to specify the publish or subscribe permission for the topic.

# 13 Troubleshooting

---

[Common Data Integration Tasks](#)

[Composite Data Integration Tasks](#)

[Data Sources](#)

[Service Integration](#)

[Device Integration](#)

## 13.1 Common Data Integration Tasks

### 13.1.1 Garbled Characters Are Displayed When Data Is Written to MRS Hive at the Destination

#### Cause Analysis

When a task is created, the storage type of the destination is inconsistent with that specified during table creation.

For example, the storage type is set to text file during table creation, whereas the storage type of the destination is set to RCFile during task creation.

#### Solution

Change the storage type of the destination to the same as that specified during table creation.

### 13.1.2 All Data Is Written to the First Field of MRS Hive

#### Cause Analysis

Column separators are used during table creation.

## Solution

Run the following command in the database to set the separator:

```
alter table Table name set serdeproperties('field.delim'=' Separator')
```

### 13.1.3 An Error Message Is Displayed at the Destination Indicating Task Execution Times Out

#### Cause Analysis

The task name contains more than 64 characters, and the first 64 characters of multiple scheduled task names are the same.

#### Solution

Reduce the task name length to fewer than 64 characters or change the task names with the same first 64 characters to different values.

### 13.1.4 Error Message "could only be written to 0 of the 1 minReplication nodes. There are 2 datanode(s) running and 2 node(s) are excluded in this operation" Is Reported at the Destination During Data Integration from MySQL to MRS Hive

#### Cause Analysis

All HDFS cluster nodes are bound to EIPs. When an MRS Hive data source is created, EIPs are used as all IP addresses in the uploaded configuration file, and IP addresses and hostnames are configured in Hosts. However, when the file is uploaded, the IP address of the node returned by the HDFS master node is an intranet IP address. As a result, the connection fails and therefore the FDI task fails to be executed.

#### Solution

Add the **dfs.client.use.datanode.hostname** configuration item to the **hdfs-site.xml** file uploaded during MRS Hive data source creation and set its value to **true**. Then, upload the modified configuration file and create the MRS Hive data source again.

## 13.1.5 Error Message "Illegal mix of collations for operation 'UNION'" Is Displayed at the Source Database During MySQL-to-MySQL Data Integration

### Cause Analysis

According to the data source view at the source, union is used in the view definition statement. However, the encoding of the fields on both sides of union is inconsistent.

### Solution

Use CONVERT to unify the encoding for fields, for example, CONVERT (ID USING utf8) AS ID.

## 13.1.6 Data May Be Lost When Incremental Data Collection Is Performed from the Source MySQL on an Hourly Basis

### Cause Analysis

The source database uses **createtime** to record the time when data is inserted and uses **updatetime** to record the time when data is updated. During task creation, **Timestamp** is set to **updatetime**. However, the value of **updatetime** is earlier than that of **createtime** in the source database. When the task is executed, the time specified **updatetime** is not within the incremental collection period. As a result, some data cannot be collected.

### Solution

Change the value of **updatetime** in the source database or use **createtime** as the timestamp field during task creation.

## 13.1.7 Error Message "401 unauthorized" Is Displayed at the Source During API-to-MySQL Data Integration

### Cause Analysis

The possible causes are as follows:

- **Security Authentication** is set to **None** during API creation, whereas **Authentication Mode** is set to **AppKey Auth** during data source creation.
- **Security Authentication** is set to **App** during API creation, whereas **Authentication Mode** is set to **AppKey Auth** and **App Authentication Type** is set to **Secret** during data source creation.

### Solution

You can perform the following operations to solve this issue:

- Select **None** for **Authentication Mode** when creating a data source.



- Change **App Authentication Type** to **Default**.

### 13.1.8 Error Message "cannot find record mapping field" Is Displayed at the Destination During Kafka-to-MySQL Data Integration

#### Cause Analysis

According to the mapping configuration, the length of the field type at the source exceeds the length limit of the field type at the destination or a field configured in mapping information does not exist in the source data.

#### Solution

Check whether the field type matches the field in the mapping configuration, whether the field length exceeds the limit, and whether the field exists. If not, modify the field configuration.

### 13.1.9 Error Message "connect timeout" Is Displayed at the Source During Scheduled API-to-MySQL Data Integration

#### Cause Analysis

There are six nodes on the data plane. Two of them cannot connect to the IP address of the API data source.

#### Solution

This is a network problem. Contact VPC custom service to locate the fault.

### 13.1.10 FDI Fails to Obtain Data During Real-Time Kafka-to-MySQL Data Integration Although Data Exists in MQS Topics

#### Cause Analysis

The address configured for the data source is not the MQS address of the instance. Therefore, the FDI task cannot collect data although messages exist in MQS of the instance.

#### Solution

Check the task configuration and data source configuration and change the data source configuration to the correct MQS address.

## 13.1.11 Value of the Source Field of the tinyint(1) Type Is Changed from 2 to 1 at the Destination During Scheduled MySQL-to-MySQL Data Integration

### Cause Analysis

The MySQL driver automatically identifies tinyint(1) as the bit type. Therefore, the field value is changed to 1 at the destination.

### Solution

Change **Connection Mode** to **Professional** for the MySQL data source and add **tinyInt1isBit=false** to the end of the character string, for example, **jdbc:mysql://ip:port/database?tinyInt1isBit=false**.

## 13.2 Composite Data Integration Tasks

### 13.2.1 Data Fails to Be Written Because the RowId Field Type Is Incorrectly Configured in the Destination Table

#### Cause Analysis

When the RowId mode is used to collect Oracle table data, an error is reported because the table field type is incorrectly configured.

```
2020/09/18 15:11:04 GMT+08:00
The data destination is abnormal. fdi_ds_idc_prod_capital_td2020-09-18 15:07:44error msg = com.huawei.eip.fdi.common.basic.exception.FDIException: Batch entry 0 with w1 as(select substr(('x'||substr(md5('AAAdoaAAGAAA+qjAAA':varchar),1,16))::bit(64)::text,33,64)::bit(32)::int4 as tra_id), upsert as (UPDATE "ods"."capital_ctms_rec_file" SET "filename"='28902_01.pdf':varchar, "upload date"='2019-07-18 16:04:50+08':timestamp::timestamp, "fileurl"='C:\loadFile\28902_01.pdf':varchar, "id"='47':numeric, "copyfileurl"='D:\BPFile\2019-07-18\28902_01.pdf':varchar, "billno"='':varchar, "recid"='28902':numeric where ("rowid"='AAAdoaAAGAAA+qjAAA':numeric) returning * ) insert into "ods"."capital_ctms_rec_file" ("filename", "uploaddate", "fileurl", "id", "copyfileurl", "billno", "recid", "rowid") select '28902_01.pdf':varchar, '2019-07-18 16:04:50+08':timestamp::timestamp, 'C:\loadFile\28902_01.pdf':varchar, '47':numeric, 'D:\BPFile\2019-07-18\28902_01.pdf':varchar, ':varchar', '28902':numeric, 'AAAdoaAAGAAA+qjAAA':numeric from w1
```

#### Solution

Modify the table field type. RowId is a string of 18 characters containing letters and digits. The field type must be character when the field is collected to the destination.

### 13.2.2 Error Message "binlog probably contains events generated with statement or mixed based replication forma" Is Displayed When the Binlog of the MySQL Database Is Read

#### Symptom

A composite task fails to be executed, and the error message "binlog probably contains events generated with statement or mixed based replication forma" is displayed.



## Solution

This problem is caused by a bug in the current Debezium version. You must upgrade the MySQL database.

For a composite task, delete the configuration of the table first. After the task is started, the data is skipped. After the task is successfully executed, configure the table again and collect data again.

### 13.2.3 Data Still Fails to Be Written After an FDI Task Failure Is Rectified

#### Cause Analysis

When the writer writes data to a database, it consumes data from a topic in Kafka. After the data is successfully saved to the database, the writer submits the consumption offset to Kafka. If an exception occurs when data is written, the consumption offset is not submitted. After the fault is rectified, restart the task to process the data that fails to be saved to the database again. Therefore, if the data does not meet the requirements of the destination after the task configuration is modified, the task still fails to be executed.

## Solution

You can use any of the following methods to rectify the fault:

- As data in a Kafka topic will age six hours later, wait until the dirty data is cleared and then start the task.
- Delete the original task and create a new task. In this way, the topics in Kafka will be deleted and the dirty data will be cleared.
- For a composite task, delete the table and start the task. Wait until the Writer consumes the data in this table and then add the table to collect data again.

### 13.2.4 Camel Fails to Access the Database Because Table Names Contain Garbled Characters

#### Cause Analysis

The parameters used for accessing the database are incorrect. As a result, the composite task fails to be started and finally the task is terminated.

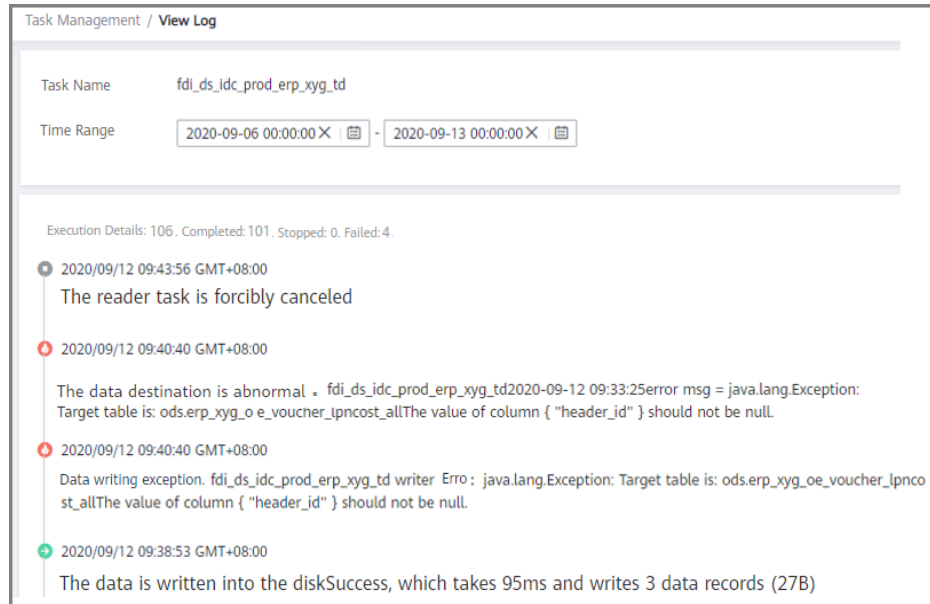
## Solution

Check whether garbled characters exist in table names in the database. If yes, contact the database administrator, delete the tables whose names contain garbled characters, and restart the task.

## 13.2.5 Inserted Data Violates the Non-null Constraint

### Cause Analysis

The task fails to be executed. The following information is displayed in the log:



### Solution

Perform the following operations to resolve the issue:

- This field in the source table is set to null, but is set to a non-null value in the destination table. To resolve this issue, modify the field attributes in the source and destination tables to ensure consistency.
- A redundant table mapping is configured during task creation. Therefore, data in the source table with a similar name is collected to the destination table. As a result, empty data is written to the field. To resolve this issue, correctly configure the table mapping.
- In the redo logs generated by the Oracle database, the **WHERE** condition is missing in the field of the UPDATE statement. As a result, the value of this field is missing. To resolve this issue, add the **WHERE** condition to the field of the UPDATE statement.

## 13.2.6 FDI Task Fails to Be Executed Because DWS Changes to the read-only State

### Cause Analysis

When the DWS disk space reaches the threshold, for example, 80%, or data skew occurs in some tables with a large amount of data, the DWS cluster becomes read-only. On the FDI side, all tasks for accessing DWS stopped abnormally at a certain time point.

## Solution

Handle the read-only issue of DWS. After the database service is restored, directly start the FDI task if the task is stopped for less than 6 hours. If the task is stopped for more than 6 hours, the channel data has expired. Reset the synchronization progress as required and start the task again. If there are tables whose data is not fully collected before the task is stopped, all data of the tables will be collected again.

In addition, analyze the causes of the read-only state for database development optimization, or expand the disk space to prevent the problem from recurring.

## 13.2.7 Speed of Writing Data to DWS Becomes Slower

### Cause Analysis

If the time required for writing data is at the second level, it is normal. If it takes more than 5 seconds to write thousands of data records, the performance deteriorates.

### Solution

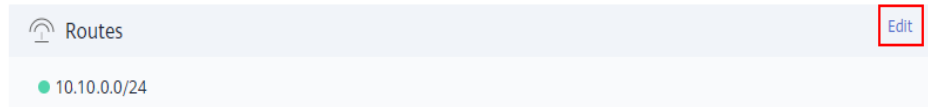
If a DWS table is frequently accessed, the write performance deteriorates. In this case, contact DWS engineers for optimization. For example, periodically use the **analyze** plan table and **VACUUM FULL**.

## 13.3 Data Sources

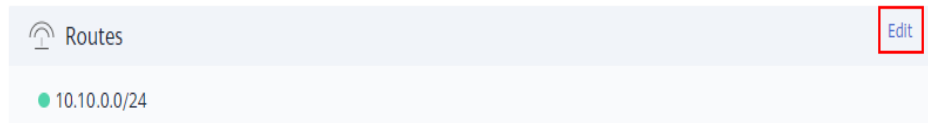
### 13.3.1 Troubleshooting a Data Source Connection Test Failure

1. Check whether the data source configuration is correct. For example, check whether the database name and password are correct.
2. Check whether the data source IP address and FDI are in the same VPC. If no, create a VPC peering connection.  
For details about how to create a VPC peering connection, see .
3. Check whether the connection port is enabled on the firewall of the data source server.
4. Check whether the IP address of the FDI data plane is added to the IP address whitelist of the data source server.
5. Check whether the data source IP address is the IP address of the customer's data center or private network. If yes, establish a VPN connection first.  
For details about how to create a VPN connection, see .
6. Check whether the route from FDI to the data source is correct.  
If the IP address is a private network address and is not in the 10.0.0.0/8, 172.16.0.0/12, or 192.168.0.0/16 network segment, add a route on the **Instances** page.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.

- b. In the **Routes** area, click **Edit**, click **Add Address**, add a route, and click **Save**.



7. Check whether the security group is bypassed.
  - a. Log in to the ROMA Connect console. On the **Instances** page, click **View Console** next to a specific instance.
  - b. In the **Security Group** area, click the security group name, for example, **default**, to access the security group console.



- c. On the **Outbound Rules** tab page, check whether the IP address configured for the data source is in the destination list.  
If the destination IP address is 0.0.0.0/0, all IP addresses are reachable. If no rule is available, click **Add Rule** and add the configured IP address to the outbound rule.

## 13.3.2 MRS Hive Data Source Connection Failed

### Cause Analysis

The possible causes are as follows:

- The private IP address specified in the MRS Hive configuration file cannot communicate with the IP address of the FDI data plane. As a result, the connection fails.
- **HDFS URL** is set to the root directory **hdfs://hacluster**, but the user specified by **Machine-machine Username** does not have the administrator permission. As a result, the connection fails.

### Solution

- Bind an EIP to the MRS Hive server and change the IP address in the configuration file to the EIP. Then, add the **dfs.client.use.datanode.hostname** configuration item to the **hdfs-site.xml** file, and set the value to **true**.
- Change **Machine-machine Username** to the name of a user with the administrator permission or change **HDFS URL** to a directory on which the user has sufficient permissions.

If the connection still fails after the preceding operations are performed, rectify the fault by referring to [Troubleshooting a Data Source Connection Test Failure](#).

### 13.3.3 FTP Data Source Connection Failed

#### Cause Analysis

The connection mode used by the server is different from that specified for the FTP data source. For example, the active connection mode is used by the server, whereas the connection mode of the FTP data source is set to **Passive**.

#### Solution

Modify the connection mode to ensure that the connection mode of the server is the same as that of the data source.

If the connection still fails after the preceding operation is performed, rectify the fault by referring to [Troubleshooting a Data Source Connection Test Failure](#).

### 13.3.4 OBS Data Source Connection Failed

#### Cause Analysis

The possible causes are as follows:

- The data source address is not set to the endpoint address.
- The AK/SK and bucket name are incorrect.

#### Solution

- Change the data source address to the endpoint address.
- Enter the correct AK/SK and bucket name.

If the connection still fails after the preceding operations are performed, rectify the fault by referring to [Troubleshooting a Data Source Connection Test Failure](#).

### 13.3.5 Kafka Data Source Connection Failed

#### Cause Analysis

**Enable SSL** is set to **Yes** when the connection address is an MQS intranet address and both **MQS SASL\_SSL** and **Intra-VPC Plaintext Access** are enabled.

#### Solution

Change **Enable SSL** to **No**.

The screenshot shows a configuration interface with two rows. The first row is labeled 'Connection Address' with a red asterisk and a help icon, followed by a text input field containing the placeholder text 'Enter the connection address.'. The second row is labeled 'Enable SSL' with a red asterisk and a help icon, followed by two radio buttons: 'Yes' (unselected) and 'No' (selected).

If the connection still fails after the preceding operation is performed, rectify the fault by referring to [Troubleshooting a Data Source Connection Test Failure](#).

## 13.4 Service Integration

### 13.4.1 Backend Service Fails to Be Invoked

#### Cause Analysis

The possible causes are as follows:

- The backend service address is incorrect.
- The timeout duration is incorrect.

If a backend service fails to return a response within the configured timeout duration, APIC displays a message indicating that the backend service fails to be invoked.

- The security group of the Elastic Cloud Server (ECS) on which the backend service is deployed cannot be accessed.

#### Solution

- Change the backend service address in the API definition.
- Increase the backend timeout duration in the API definition.
- Ensure that the inbound and outbound ports and protocols configured for services are correct.

### 13.4.2 Error Message "No backend available" Is Displayed When an API Is Called

#### Cause Analysis

The possible causes are as follows:

- The backend service cannot be accessed.
- The required port is not enabled in the ECS security group.

#### Solution

- Modify the backend service.
- Enable the required port in the ECS security group.

### 13.4.3 Error Message "The API does not exist or has not been published in an environment" Is Displayed When an API Is Called Using JavaScript

#### Cause Analysis

The possible causes are as follows:

- The API is not published in an environment.



- The URL to be accessed is different from that in the API details.
- The API uses the OPTIONS method for cross-domain requests. However, CORS is not enabled and **OPTIONS** is not selected for **Method** during API creation.

## Solution

- Publish the API in an environment.
- Modify the URL to ensure that the URL is the same as that in the API details. If a slash (/) is missing after the URL, the API cannot be matched.  
Example: **http://example.com/test/** and **http://example.com/test** match different APIs.
- Enable CORS and select **OPTIONS** for **Method** when creating the API.

## 13.5 Device Integration

### 13.5.1 Error Message "java.lang.IllegalArgumentException: {IP address}\_{timestamp}" Is Displayed During Demo Running

#### Cause Analysis

No EIP has been bound to the ROMA Connect instance.

#### Solution

Purchase a new EIP or bind an existing EIP to the ROMA Connect instance. After the binding is successful, run the demo.