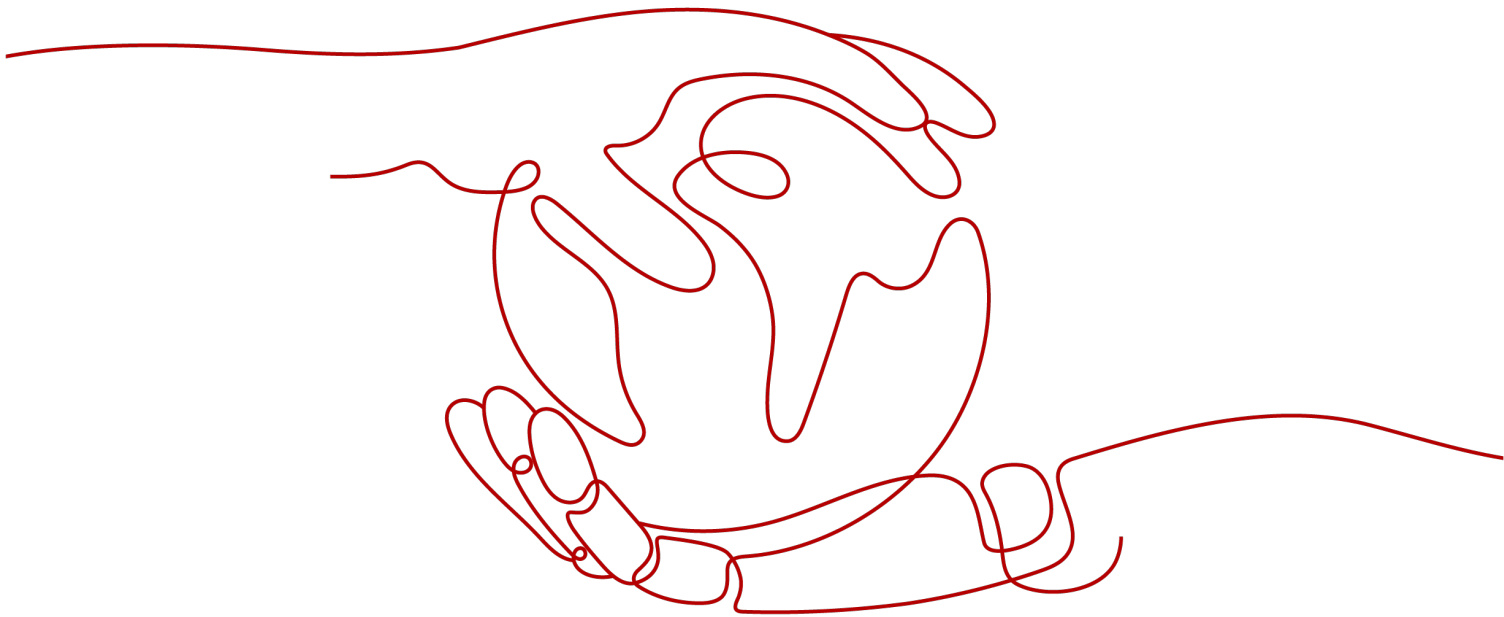


解决方案实践

# 普元元数据资产管理解决方案实践

文档版本 1.1  
发布日期 2024-04-25



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 安全声明

## 漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

# 目录

<b>1 方案概述</b>	<b>1</b>
<b>2 资源和成本规划</b>	<b>13</b>
<b>3 操作流程</b>	<b>14</b>
3.1 数据标签库管理操作流程	14
3.1.1 基础配置-宽表配置	15
3.1.2 基础配置-主体配置	16
3.1.3 标签制作-标签创建	18
3.1.4 业务画像-画像更新	22
3.1.5 标签应用-画像查询与群体分析	24
3.2 数据共享交换管理操作流程	25
3.2.1 数据资源目录管理	26
3.2.2 数据使用	30
3.2.3 数据服务开发	33
3.2.4 服务控制	37
3.2.5 调度管理	39
3.2.6 运行监控	41
3.2.7 统计分析	46
<b>4 实施步骤</b>	<b>50</b>
4.1 数据标签库管理实施步骤	51
4.2 数据共享交换管理实施步骤	56
4.2.1 安装前准备	56
4.2.2 安装须知	58
4.2.3 DSP Governor	59
4.2.4 DSP Server	64
4.2.5 DSP Scheduler	66
4.2.6 DSP DataRelease	68
4.2.7 DSP SSM	70
4.2.8 DSP Agent	72
4.2.9 DSP GatherClient	74
4.2.10 DSP Studio	76
<b>5 附录</b>	<b>78</b>



6 修订记录.....82

# 1 方案概述

## 应用场景

- **场景1：数据标签库应用场景**

通过标签管理分析与预测，企业提升服务质量、效率、回馈。

**客户的痛点：**

- a. 潜在投诉风险预警：热线渠道是电网企业的主要服务渠道，随着业务的不断发展，人工话务强度逐步增高，但精准服务能力低。如何在有限人力资源条件下，准确识别客户诉求、提高人工精准服务和精细化服务水平、降低人工话务工作压力、成为客户服务方面亟需解决的问题。
- b. 电费回收风险：近年来社会经济发展趋缓，电力公司电费回收压力日益增大，电费回收风险防控难度不断增加，同时，由于缺乏对优劣客户的科学评价方法，电力公司无法为客户提供差异化服务，不利于供电企业权益的维护。
- c. 如何识别替代潜力客户：我国环保形势严峻，党和政府高度重视节能减排工作和大气污染治理，大力推进能源消费革命和供给侧结构性改革，在终端能源消费环节使用电能替代散烧煤、燃油的能源消费方式，有利于提高环保水平
- d. 缺乏精细化策略与运维：目前设备状态精细化管控工作，仍没有相关的信息系统支撑，开展电网设备状态评价和运维工作时，主要基于各系统直接量测和录入的数据作为状态评价和运维决策条件，缺少基于设备综合信息的支撑方法

**通过本方案实现的业务效果：**

- a. 客户对于停电通知服务满意度提高，停电报修、咨询话务量大幅下降，停电投诉量大幅减少。  
采用数据建模方式综合识别停电敏感客户，实现对停电敏感度的量化分析，减少了人工主观因素，提升对停电敏感客户的识别精度；  
基于投诉敏感标签，实现对敏感客户的精准服务，提升了客户服务质量，减少人员维系成本。
- b. 企业平均回款时长和逾期缴费率显著下降。  
促使电费风险管控工作从定性主观判断转向定量精准分析，从事后处理走向提前防控，提高了风险防控工作的效率和效益；  
根据客户的不同特征制定差异化防控策略，促使策略与客户特征更加匹配，策略的执行更加有效，节省了催费的人力物力消耗，降低了企业经营成本。

- c. 提高潜力客户的识别准确率，相比于人工普查准确率提高近 20 倍。  
将全面普查转变为针对性排查，从而将有限的企业资源用到高潜力客户上，大幅提高工作效率，高效拓展终端市场；  
依据电能替代潜力标签，结合用户信用、资金实力等标签，针对性地解决用户电能改造中面临的资金投入、生产回报等问题，推动企业落实电能改造工作。
  - d. 支撑企业对设备特征进行标签化提炼，构建了多维度、多层次、多视角的设备标签体系，帮助企业全面深入了解设备特征，使得企业对设备的认知更系统化、明朗化，提高管理效率，降低管理成本。同时将标签融合到采集运维工作的各处理环节中，实现支撑差异化运维策略，使得设备生产运维业务经验得到沉淀、固化、共享。
- **场景2：数据共享交换管理应用场景**  
**客户的痛点：**
    - a. 管理不规范：数据模型管理缺乏规范化体系，大数据平台内数据的技术、业务属性不清晰，没有完善的数据模型管理体系与规范。
    - b. 使用不明确：由于数据业务特性与使用方法不统一，难以对结构化数据、半结构化、非结构化数据的快速集成和共享。
    - c. 监控不到位：在实际数据运行中，会出现数据信息发生中断异常或数据量发生突然变化等数据异常监控。无法通过传统的监控手段及时发现。

**通过本方案实现的业务效果：**

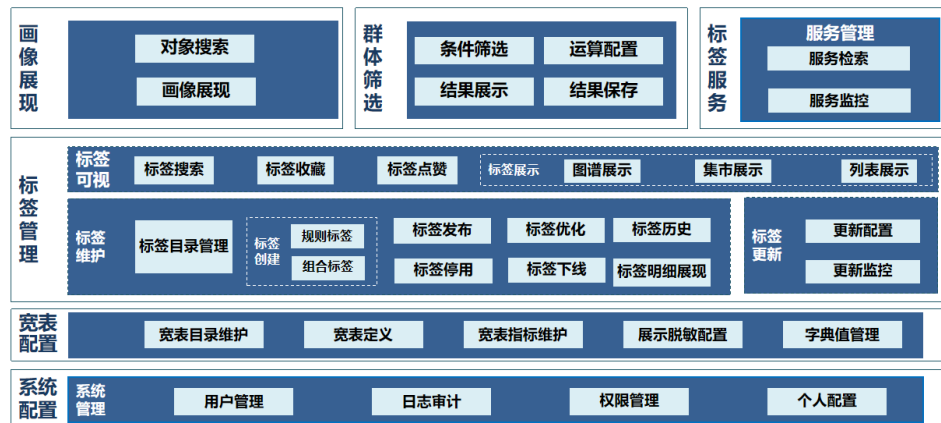
实现数据流整体监控以及数据的资产化，资产的服务化，以及数据服务的共享和监控

- a. 自助数据：可自助组织数据。数据散落到大数据平台各数据表中，通过数据共享发布平台可收集不同表中不同信息字段的数据，提供数据服务化能力；
- b. 数据服务化：实现数据目录中数据的服务化，能将数据目录中的数据以文件服务、Web接口服务、数据库服务、流数据服务等方式对外提供；
- c. 数据协同：跨部门的数据共享协作，通过技术元数据与业务元数据的对接避免数据不一致和数据开发返工；
- d. 数据工作台：工作簿根据收集的数据自动推荐关联数据，协助用户找到适用的数据，提供数据关联分析，如基于地址推荐各数据分类中包含地址数据的所有数据。

## 方案架构

- **数据标签库管理方案架构**
  - a. **功能架构**

图 1-1 功能架构



**主体配置：**支持多个标签主体的创建与维护，可满足对客户、设备、数据、员工、商品等不同主体对象的标签建设需求。

**标签体系管理：**提供多层级的标签分类管理功能，支撑对标签的体系化管理，满足各种标签体系建设需求。

**标签创建：**提供多种标签生成方式，比如规则标签、手工标签、组合标签等，以满足不同特性的标签创建需求。

**标签管理：**支持标签创建、审核、发布、评估、停用、优化下线的生命周期管理流程，实现标签的全生命周期管理，保证标签的运行状态清晰、有序、可控。

**标签调度：**支持自定义标签调度更新时间，实现静态标签与动态标签的创建，满足不同的时效性的数据标签更新需求。

**画像展现：**支持通过精准搜索与模糊搜索进行目标对象画像查询，并支持通过以词云或列式的方式展现标签，直观地将业务对象特征信息进行展现。

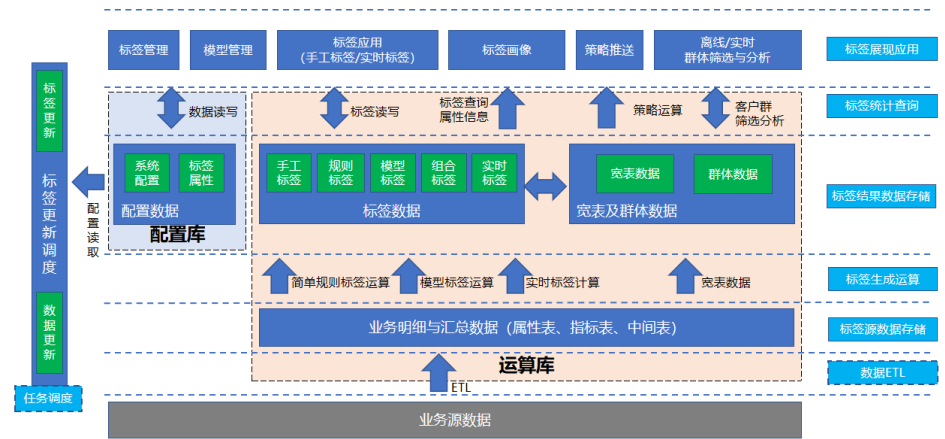
**群体圈选：**支持基于标签和宽表字段进行业务对象筛选功能，灵活快速定位目标群体。

**标签服务：**支持将标签成果以服务的形式对外发布，并支持对标签服务调用情况进行监测

b. **数据逻辑架构**

产品分为配置库与运算库，配置库主要存储系统配置数据需要支持事务性的关系型数据库，运算库按照标签数据量进行选型，默认设置下配置库与运算库会存放于一起

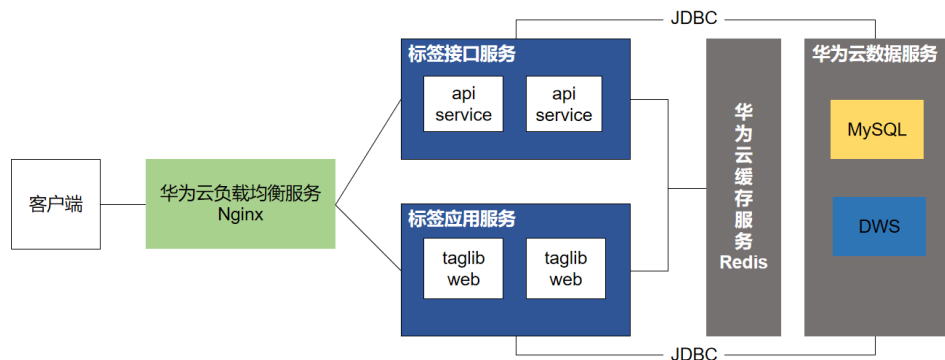
图 1-2 数据逻辑架构



c. 逻辑部署架构

标签画像负载均衡采用华为云平台负载均衡服务，标签画像分接口服务与应用服务进行部署，数据库、缓存等服务直接采用华为云数据服务

图 1-3 逻辑部署架构



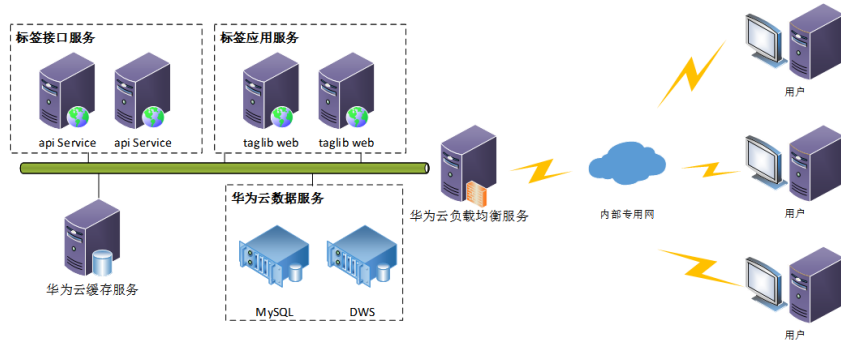
d. 物理部署架构

产品支持线性扩展部署，通过灵活地扩展服务器数量提升系统处理能力，满足不同的标签处理与应用需求

单机方式部署请情况下，标签接口服务、标签画像应用服务共同部署在一台服务器中即可，且不需要Redis缓存服务。

集群情况下，通常由2台服务器部署接口服务组成标签接口服务集群，由2台服务器部署标签画像应用服务组成应用集群，且必须使用Redis缓存服务共享会话及应用热数据

图 1-4 物理部署架构



e. 部署架构与资源建议

产品支持线性扩展部署，通过灵活地扩展服务器数量提升系统处理能力，满足不同的标签处理与应用需求

图 1-5 部署架构与资源建议



• 数据共享交换管理方案架构

a. 应用架构

数据服务共享平台包括资源目录管理、数据使用、数据服务开发、服务控制、调度管理、运行监控、运行维护和数据分析与信息展现，应用架构如下图所示：

图 1-6 应用架构



**资源目录管理：**是可交换数据元数据的结构化展现，需支持数据库、大数据、文件、Web 服务等多类型数据资源技术元数据的采集和业务元数据的维护能力；支持面向消费者业务视图（比如按主题的划分）的创建；提供资源注册、维护和搜索等基本功能；

**数据使用：**主要面向消费方，是消费者如何通过平台申请数据资源及数据管理方审批处理的过程；消费方可通过注册功能自行在平台注册，支持消费方浏览数据资源后提交资源申请（拉）和订阅（推）的方式；消费方可查看所有申请的状态及历史记录，可查看申请的汇总情况；支持单个或批量申请；

**数据服务开发：**主要面向开发人员，提供在线和离线两种开发方式。对于如全量开发同步、数据库到文件同步等的作业，开发人员可在 Web 页面中进行简单配置完成在线作业开发；对于复杂模型，可通过离线方式开发。数据服务开发提供作业目录管理、作业流的可视化配置等；

**服务控制：**平台提供用户对数据服务过程中的权限控制，包含 IP 白名单、服务状态、调用关系管理等；

**调度管理：**作业运行的指挥中心，可通过调度管理配置任务的调度策略，配置任务运行的优先级及触发方式等；

**运行监控：**对整个平台运行过程中的状态进行监控，包括物理资源、服务引擎、传输监控、故障告警、消费方等，同时提供对日志和历史记录的查看功能。

**运维维护：**为方便平台使用的功能，支持资源目录、作业模板、服务接口的导入导出；

**数据分析与展现：**针对平台作业、数据交换总量、文件传输、消费方等进行统计分析。

## b. 数据架构

数据架构反映平台中，各种类型数据的分布及流转情况。在数据服务共享平台中，数据可分为业务数据、元数据（含资源目录）、质量数据、作业模板及运行监控数据、统计分析数据等。

**业务数据：**指承载业务含义（大多来源于企业内部各业务系统及少量外部数据），由各总部及各分子公司业务部门（数据提供方）提供，转移至前置区（数据准备）后，经过相应的处理和扩展，提供给消费方。具备数据提供、数据获取、数据准备、数据处理及服务化开发、数据共享发布和数据消费的完善数据交换过程。业务数据一般存储于 ODS 区，也可直接提供点对点的方式，业务数据在平台中不存储；

**元数据（资源目录）：**业务数据交换的整个过程，都是以元数据（资源目录）为驱动。元数据可分为技术元数据和业务元数据两种，在数据交换过程中，通过手动和自动的方式进行管理；元数据存储于平台数据库中；

**质量数据：**针对业务数据所开展质量评估后产生的数据，一般包括质量检核规则和检查结果数据；质量数据存储于平台数据库中，也可将结果数据生成文档后发送给业务系统主管部门；

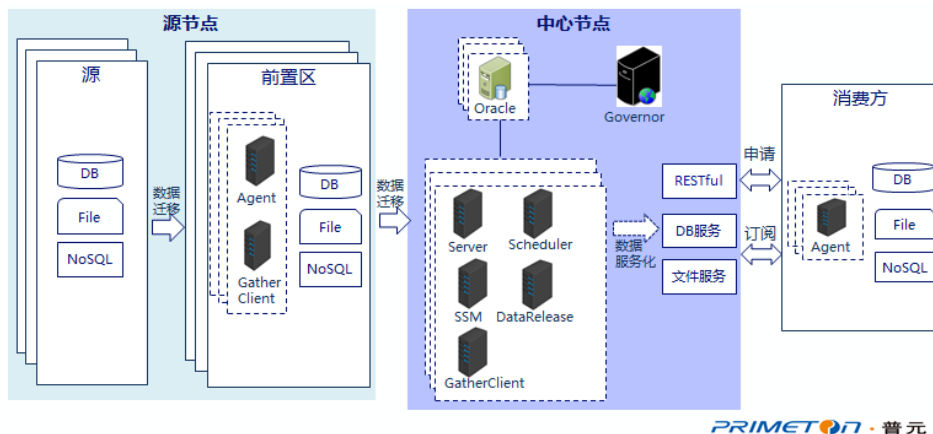
**模型及运行监控数据：**平台运行过程中，各类作业的模式（模板）及配置信息，作业运行过程中的告警、监控信息和日志信息等，作业运行过程中的调度策略及调度信息；模型及运行监控数据存储于平台数据库中；

**统计分析数据：**针对作业、数据交换量、消费方等进行的统计分析；部分数据直接取自模型及运行监控数据，部分数据直接存储于平台数据库。

## c. 部署架构

数据服务共享平台部署机构如下所示：

图 1-7 部署架构



数据服务共享平台适应于单级或多级的部署方式：

**单级部署：**当企业没有上级单位、平级单位、下级单位数据共享需求时可以采用单级部署方式，单级部署时依据生产环境实际情况，部署 DSP 产品所有组件于物理服务器（建议服务器数量最少为二台及以上）中，其中 SSM 与 DataRelease 组件必须关联部署在同一台服务器中，当数据交换的作业量较大时，可以对 Server 组件进行集群部署，当数据服务（RESTful）的访问量较大时，可以对 SSM 与 DataRelease 组件进行集群部署。

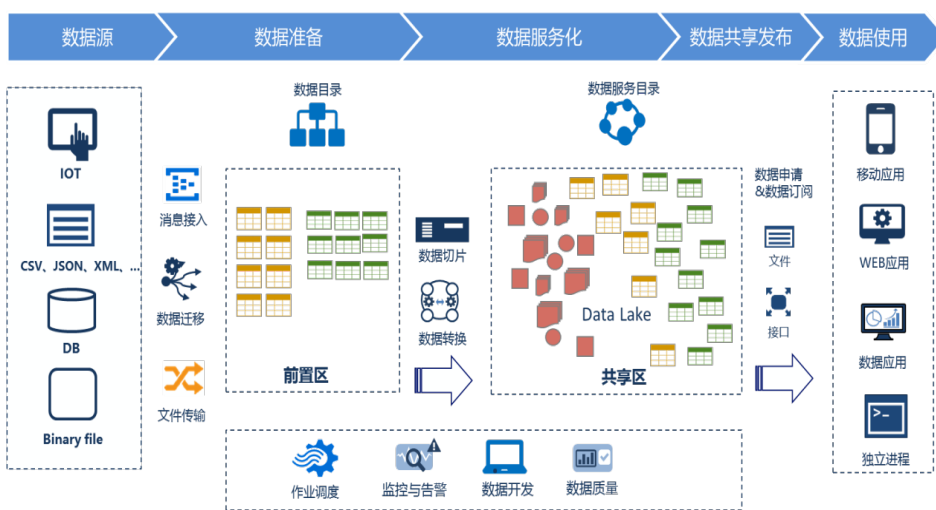
**多级部署：**当企业对上级单位、平级单位、下级单位有数据共享需求时可以采用多级部署方式，多级部署时将 Agent 和 GatherClient 部署于有数据共享需求的上级单位、平级单位、下级单位的物理服务器中，其他产品组件部署于企业内部服务器中，其中 SSM 与 DataRelease 组件必须关联部署在同一台服务器中，当数据交换的作业量较大时，可以对 Server 组件进行集群部署，当数据服务（RESTful）的访问量较大时，可以对 SSM 与 DataRelease 组件进行集群部署。

d. 技术架构

技术视图

数据服务共享平台整体技术框架及各过程所采用技术如下图所示：

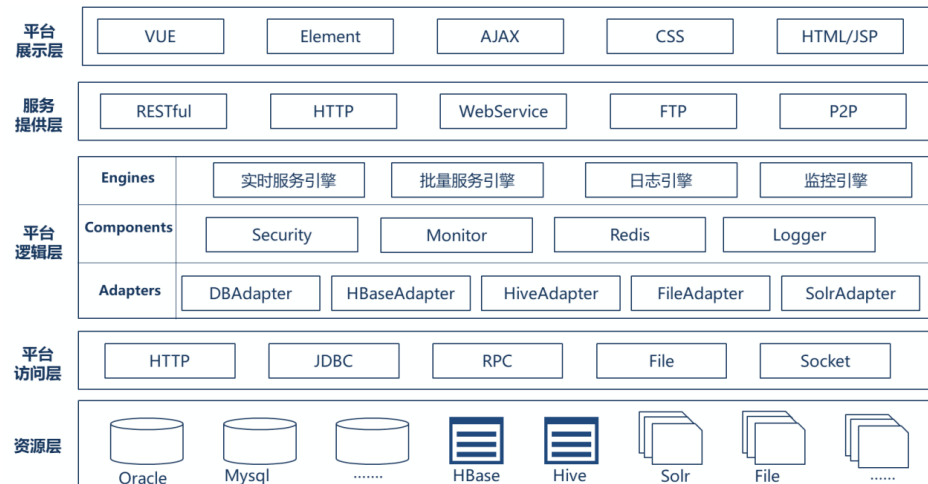
图 1-8 技术架构





数据服务共享平台通过多种数据（结构化数据、非结构化数据、文件数据）、协议（Http、JDBC、Socket）接入方式将已有的业务数据接入 DSP 平台，在平台中通过数据转换、加工计算等逻辑操作，发布为数据服务，通过数据共享区对外提供多种类型的数据服务。

图 1-9 技术实现



**资源层：**支持对主流关系型数据库（Oracle、MySQL、SQLServer、PostgreSQL、MongoDB、Gauss100、DM），大数据（HBase、Hive）、文件等多种类型数据资源技术元数据的采集和业务元数据的维护。

**平台访问层：**基于 JDBC、HTTP、RPC 等通信协议实现与底层数据和文件间的访问交互，基于 Socket 通信实现平台各组件间的数据通信，确保平台中数据间交互与通讯高效、安全、可靠。

**平台逻辑层：**平台逻辑层依托于各类适配器、组件、引擎保证平台的安全稳定运行，适配器有数据库（DBAdapter）、HBase（HBaseAdapter）、Hive（HiveAdapter）、文件（FileAdapter）等类型；组件包含安全、监控、Redis 缓存、日志等组件；引擎包含实时服务引擎、批量服务引擎、监控引擎、日志引擎等。

**服务提供层：**服务提供层采用轻量级的 Web Service 架构（RESTful），其实现和操作比 SOAP 和 XML-RPC 更为简洁，可以完全通过 HTTP 协议实现对外部应用提供统一的 RESTful Web 服务。

**平台展示层：**平台展示层使用 VUE+ElementUI 框架，结合 CSS、AJAX 等业界前沿技术实现丰富的界面展示效果，提升使用者的交互体验。

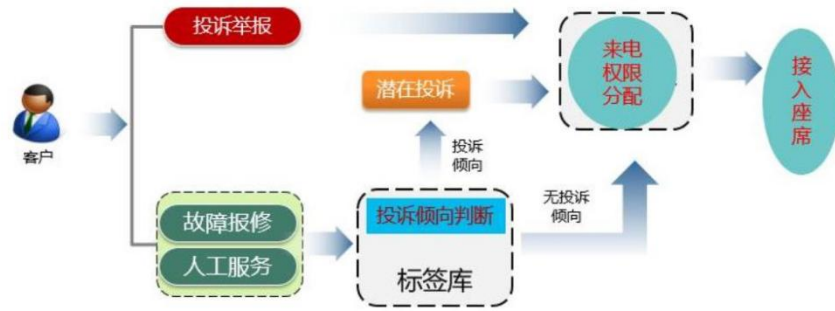
## 方案优势

- **数据标签库管理方案优势**

- a. 潜在投诉风险预警，提升客户服务质量

通过分析 95598 呼叫行为数据、客户档案数据、实时数据等信息，建立潜在投诉倾向模型，输出“潜在投诉高风险”、“同一事件频繁投诉”、“倾向复电不按时投诉”等标签，依据客户标签推荐差异化服务策略，提升客户服务品质。

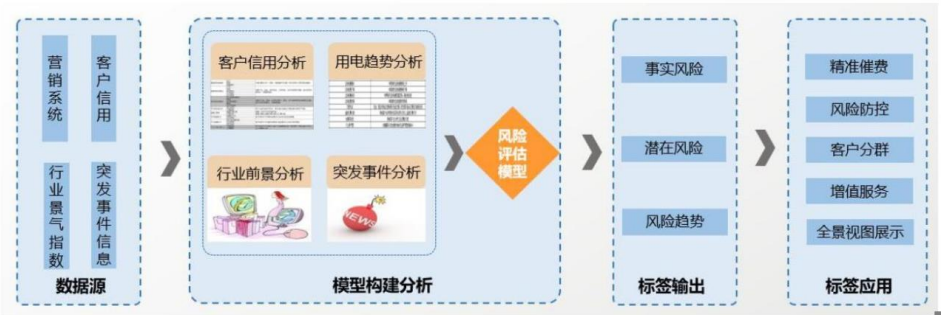
图 1-10 潜在投诉风险预警



b. 精准预测电费回收风险，缩短电费回收周期

通过分析客户信用、用电趋势、行业前景信息和突发事件，构建电费回收风险模型，并将分析成果通过标签的形式固化至普元标签产品，并依据用户风险级别标签等特征采取精准化风险防控策略，从而达到缩短电费回收周期的目的。

图 1-11 精准预测电费回收风险



c. 准确定位电能替代潜力客户，提高市场开拓效率

从已实现电能替代的客户的画像特征出发，通过分析企业行业类别、用电趋势、行业景气度和用电负荷等数据，构建电能替代模型，准确预估企业电能替代潜力，根据不同潜力等级客户及其特征设计相应推广策略，提高市场开拓效率。

图 1-12 准确定位电能替代潜力客户



d. 精细化设备管控策略，提升运维效率

根据设备名称、归属地、年龄、资产价值、运行状态等特征属性以及缺陷敏感度、负载强度、事故风险等业务表现特征，构建多层次、多维度、多视角的设备标签体系，形成全景设备画像。并在标签设计的应用环节，充分考虑了采集异常生成、远程处理、派工、现场处理和运维反馈全流程环节，并结合每个流程业务的特点，将特定的标签应用于相应流程进行计算分析。同时基于产品策略配置能力，通过作业项目、作业周期、作业优先级、标签反馈等类型策略内容配置，支撑一线运维人员通过高效识别和管理设备特征信息，针对性的开展设备运维工作。

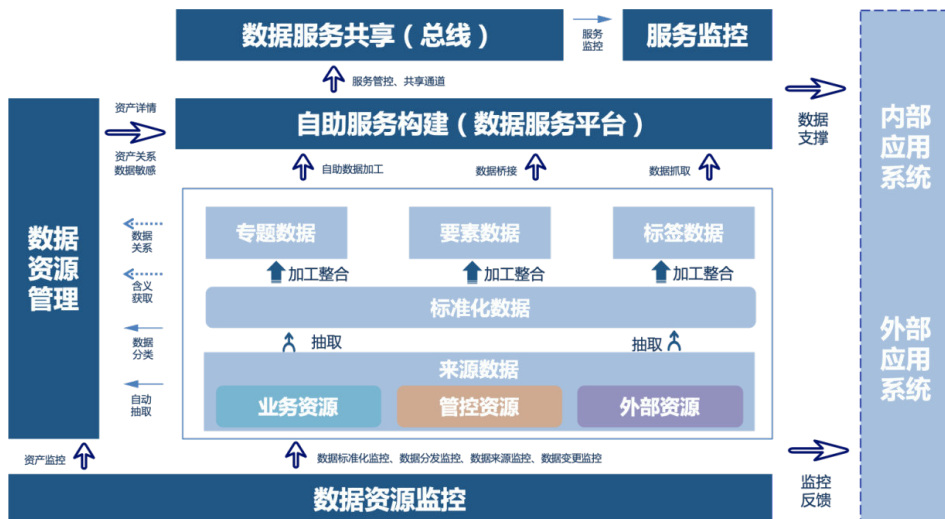
图 1-13 精细化设备管控策略



• 数据共享交换管理方案优势

DSP作为一个统一的数据服务共享平台，主要提供了对内部应用系统和外部应用系统的数据支持，具体方案如下：

图 1-14 数据共享交换管理方案



平台包含数据资源管理、数据资源监控、自助服务构建、数据服务共享、服务监控等几部分内容：

**数据资源管理：**提供对已有的社会资源、管控资源、技术资源等数据资源，实现自动化的元数据采集，摸清大数据资源状况，进行抽取转换，形成标准化的数据资源，对标准化的数据资源进行分类、加工整合，形成标准的可管理的专题、要素、标签等类型数据资源，为数据服务共享打下数据基础。

**数据资源监控：**对平台中数据资源流向、数据抽取转换过程及结果、数据加工过程及结果实现闭环监控，面向客户做到问题数据资源的及时发现与处理，保障平台数据的安全可靠。

**自助服务构建：**区别于传统的数据资源提供方式，支持自主的数据准备，快速整合多元数据，形成服务数据，对外提供web、文件、数据库、流数据四种服务开发能力，并构建服务目录，满足业务及应用需求。

**数据服务共享：**全局角度为跨地域数据和服务调用提供支持，为所有的异构资源提供统一的访问通道，实现系统资源管理、数据传输交换和数据共享服务管理。

**服务监控：**对服务调用，服务异常，数据传输，节点异常，服务耗时，服务热度等，进行服务全生命周期监控。

## 约束与限制

- **数据标签库管理约束与限制**
  - a. 支持的运行环境操作系统
    - Windows server 2008/Windows server 2008 R2/Windows server 2012/Windows server 2012R2/Windows server 2016;
    - Red Hat Enterprise Linux 6 或以上;
    - CentOS 6 或以上;
    - Ubuntu 14 或以上。
  - b. 支持的数据库
    - Oracle 10g/11g/12c;
    - GBase 8a 或以上;
    - Mysql 5.6 或以上;
    - Hadoop 大数据平台-CDH 5.7 或以上;
    - 华为 Gauss DB 100/200;
    - 星环大数据平台-TDH 5.1 或以上;
    - Pivotal Greenplum 5 或以上;
    - SAP HANA 2.0 或以上;
    - PostgreSQL 9 或以上。
  - c. 支持的浏览器
    - IE 10 或以上;
    - Chrome 54 或以上;
    - Firefox 61 或以上;
    - Safari 11 或以上。
- **数据共享交换管理约束与限制**
  - a. 支持的操作系统
    - 运行环境：**
      - 银河麒麟服务器操作系统V4、银河麒麟高级服务器操作系统V10
      - 中标麒麟高级服务器操作系统软件V7.0
      - UOSV20
      - Windows 7/10
      - WindowsServer2008

CentOS 7.2

**开发环境:**

Windows 7/10

WindowsServer2008

b. 支持的JavaEE服务器

AppServerV6.5/V7

Tomcat 8.5.51

c. 支持的数据库

MySQL 5.5.4/5.6.33/5.7.20

Oracle 11g /Oracle12.1.0.2.0/Oracle12.2.0.1

SQL Server 2016

PostgreSql 9.6

DMV7/V8

d. 支持的浏览器

IE 11

Chrome 70

Firefox 65Firefox 78

e. 支持的JDK版本

开发环境: Oracle JDK1.8.0\_172

运行环境: Oracle JDK1.8.0\_172

# 2 资源和成本规划

## 数据标签库管理资源和成本规划

图 2-1 数据标签库管理资源和成本规划

产品名称	系统注册用户数 (根据系统性能参数包但不限于注册用户数/并发数等)	华为云配置详情	可用区	数量	包年 (元/年)	合计(元/年)	
数据标签库管理	1-20个用户	【应用服务器】鲲鹏计算   鲲鹏通用计算增强型   kc1.large.2   2核   4GB 【镜像】 EulerOS   EulerOS 2.8 64bit with ARM 【系统盘】 高IO   40GB	北京四, 可用区二	7	25,130.00	50,130.00	
		【带宽】 IP1个, 按流计费1TB, 时长8640小时	北京四	7			
		【硬盘】 超高IO 100G	北京四, 可用区二	10			7,000.00
		【数据库】 RDS MySQL T V1.0   单机版 8核 32GB 硬盘 超高IO 500GB	北京四, 可用区二	1			18,000.00
		【企业主机安全】 企业版	北京四	7			
	20-100用户	【应用服务器】鲲鹏计算   鲲鹏通用计算增强型   kc1.large.2   4核   8GB 【镜像】 EulerOS   EulerOS 2.8 64bit with ARM 【系统盘】 高IO   40GB	北京四, 可用区二	16	57,440.00	183,520.00	
		【带宽】 IP1个, 按流计费1TB, 时长8640小时	北京四	16			
		【硬盘】 超高IO 100G	北京四, 可用区二	30			30,000.00
		【数据库】 RDS MySQL T V1.0   主备版 18核 32GB 硬盘 超高IO 1000GB	北京四, 可用区二	1			36,000.00
		【数据库】 RDS GaussDB T V1.0   单机版 8核 32GB 硬盘 超高IO 500GB	北京四, 可用区二	1			60,080
	其他	【应用服务器】鲲鹏计算   鲲鹏通用计算增强型   kc1.large.2   4核   8GB 【镜像】 EulerOS   EulerOS 2.8 64bit with ARM 【系统盘】 高IO   40GB	北京四, 可用区二	按需配置		3,590.00	按需配置
		【带宽】 IP1个, 按流计费1TB, 时长8640小时	北京四	按需配置			
		【硬盘】 超高IO 100G	北京四, 可用区二	按需配置	1,000.00		
		【数据库】 RDS MySQL T V1.0   主备版 18核 32GB 硬盘 超高IO 1000GB	北京四, 可用区二	按需配置	36,000.00		
		【数据库】 RDS GaussDB T V1.0   单机版 8核 32GB 硬盘 超高IO 500GB	北京四, 可用区二	按需配置	60,080		

## 数据共享交换管理资源和成本规划

图 2-2 数据共享交换管理资源和成本规划

产品名称	系统注册用户数 (根据系统性能参数包但不限于注册用户数/并发数等)	华为云配置详情	可用区	数量	包年 (元/年)	合计(元/年)	
数据共享交换管理	1-20个用户	【应用服务器】鲲鹏计算   鲲鹏通用计算增强型   kc1.large.2   2核   4GB 【镜像】 EulerOS   EulerOS 2.8 64bit with ARM 【系统盘】 高IO   40GB	北京四, 可用区二	7	25,130.00	50,130.00	
		【带宽】 IP1个, 按流计费1TB, 时长8640小时	北京四	7			
		【硬盘】 超高IO 100G	北京四, 可用区二	10			7,000.00
		【数据库】 RDS MySQL T V1.0   单机版 8核 32GB 硬盘 超高IO 500GB	北京四, 可用区二	1			18,000.00
		【企业主机安全】 企业版	北京四	7			
	20-100用户	【应用服务器】鲲鹏计算   鲲鹏通用计算增强型   kc1.large.2   4核   8GB 【镜像】 EulerOS   EulerOS 2.8 64bit with ARM 【系统盘】 高IO   40GB	北京四, 可用区二	16	57,440.00	183,520.00	
		【带宽】 IP1个, 按流计费1TB, 时长8640小时	北京四	16			
		【硬盘】 超高IO 100G	北京四, 可用区二	30			30,000.00
		【数据库】 RDS MySQL T V1.0   主备版 18核 32GB 硬盘 超高IO 1000GB	北京四, 可用区二	1			36,000.00
		【数据库】 RDS GaussDB T V1.0   单机版 8核 32GB 硬盘 超高IO 500GB	北京四, 可用区二	1			60,080
	其他	【应用服务器】鲲鹏计算   鲲鹏通用计算增强型   kc1.large.2   4核   8GB 【镜像】 EulerOS   EulerOS 2.8 64bit with ARM 【系统盘】 高IO   40GB	北京四, 可用区二	按需配置		3,590.00	按需配置
		【带宽】 IP1个, 按流计费1TB, 时长8640小时	北京四	按需配置			
		【硬盘】 超高IO 100G	北京四, 可用区二	按需配置	1,000.00		
		【数据库】 RDS MySQL T V1.0   主备版 18核 32GB 硬盘 超高IO 1000GB	北京四, 可用区二	按需配置	36,000.00		
		【数据库】 RDS GaussDB T V1.0   单机版 8核 32GB 硬盘 超高IO 500GB	北京四, 可用区二	按需配置	60,080		

# 3 操作流程

- 3.1 数据标签库管理操作流程
- 3.2 数据共享交换管理操作流程

## 3.1 数据标签库管理操作流程

图 3-1 数据标签库管理操作流程

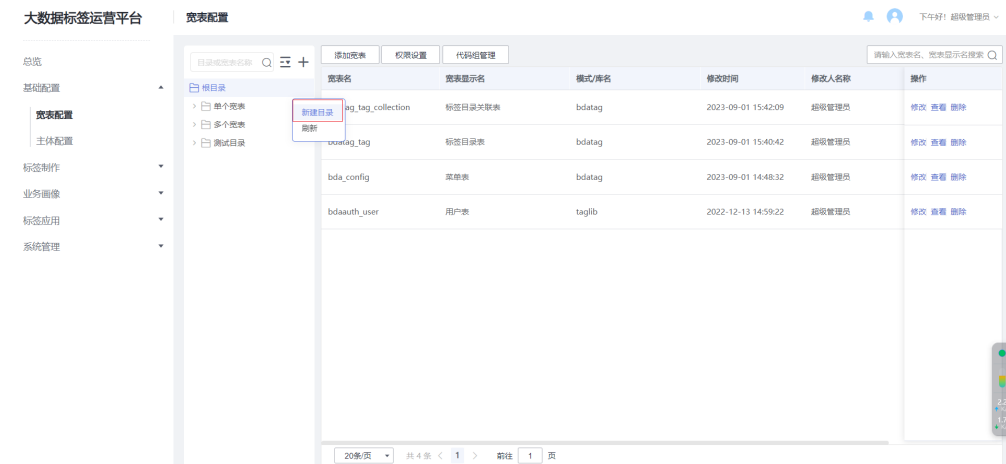




### 3.1.1 基础配置-宽表配置

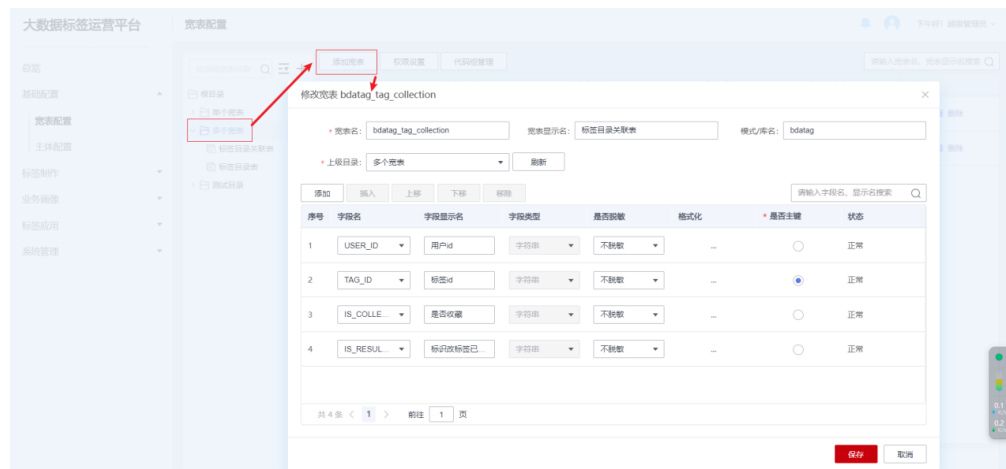
步骤1 基础配置-宽表配置页面，新增目录

图 3-2 宽表配置 1



步骤2 选择左侧目录，单击添加宽表，填写必填项，单击加载，下方显示该宽表的所有字段，可对字段显示名进行编辑

图 3-3 宽表配置 2



说明：

“模式/库名”与“宽表名”是对应的，“模式/库名”就是数据库schema，“宽表名”就是库里面的表；“模式/库名”和“宽表名”可以填写计算库配置的这个账号里面所有有权限的库和表

可以添加多张宽表

是否主键：选择数据库表的主键或者值唯一的字段作为主键，宽表的主键尽量是字符类型

是否脱敏：选择字段是否脱敏

格式化：选择字段格式化（基础配置-宽表配置页面，代码组管理里面新增代码组和代码值。表示该字段的值后续在标签画像页面用设置的代码值表示。



基础配置-宽表配置-权限设置，表示将宽表授权给其他角色，后续其他角色所拥有的用户登录才有权查看该宽表，和使用该宽表

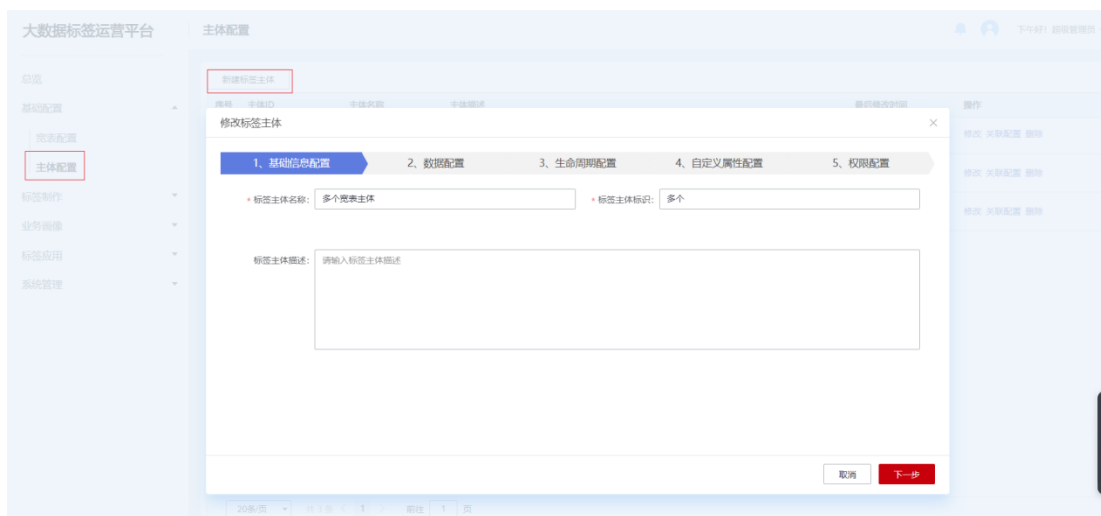
----结束

### 3.1.2 基础配置-主体配置

基础配置-主体配置页面，新增标签主体，在新增标签主体页面

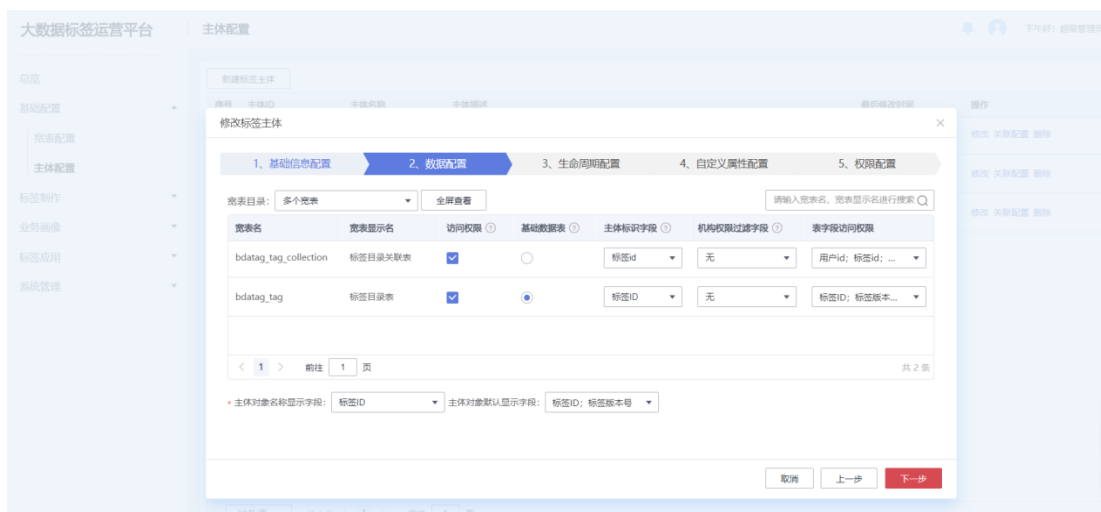
**步骤1** 基础信息sheet页，填写标签主体名称和主体标识，单击下一步

图 3-4 主体配置 1



**步骤2** 数据配置sheet页，选择宽表目录，下方显示该目录下所有宽表，针对宽表进行选择，配置完成，单击下一步

图 3-5 主体配置 2



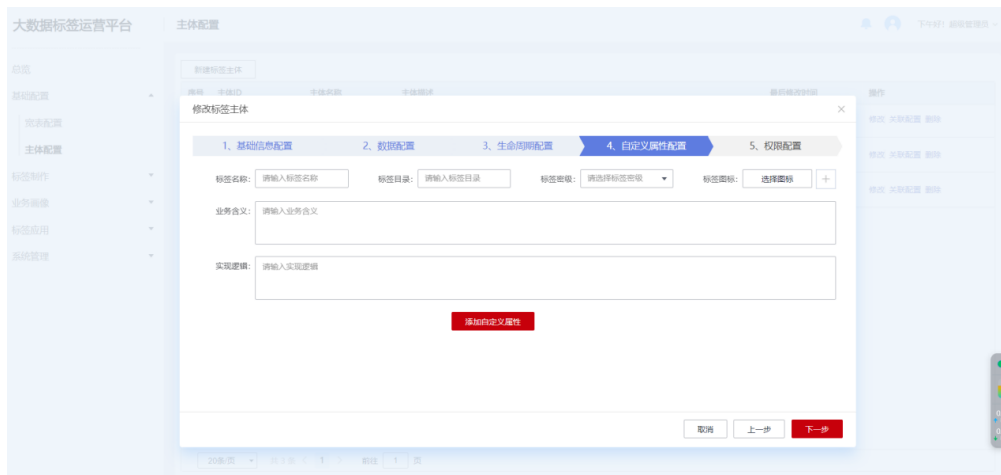
**步骤3** 生命周期配置sheet页，可以针对标签的生命周期设置是否需要审批，单击下一步

图 3-6 主体配置 3



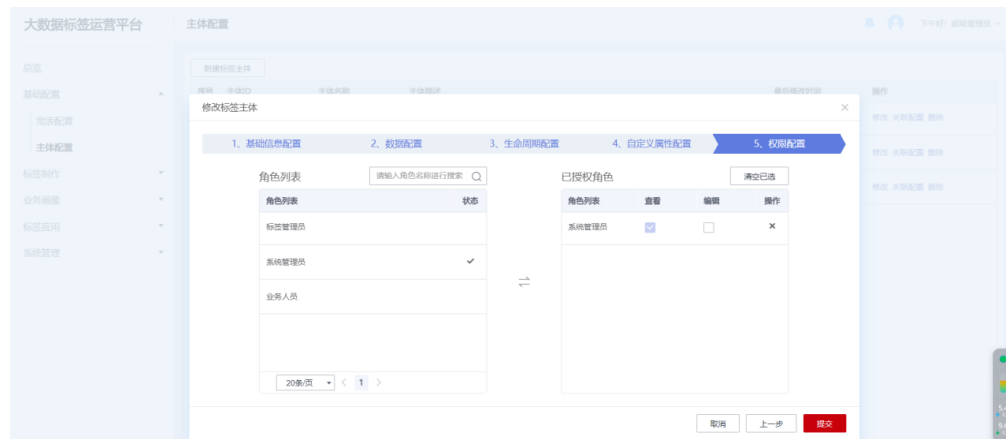
步骤4 属性配置sheet页，填写相关的标签属性，单击下一步

图 3-7 主体配置 4



步骤5 权限配置sheet页，设置该主体的权限<将主体授权给其他角色，后续其他角色所拥有的用户登录才可以使用该主体>

图 3-8 主体配置 5



----结束

### 3.1.3 标签制作-标签创建

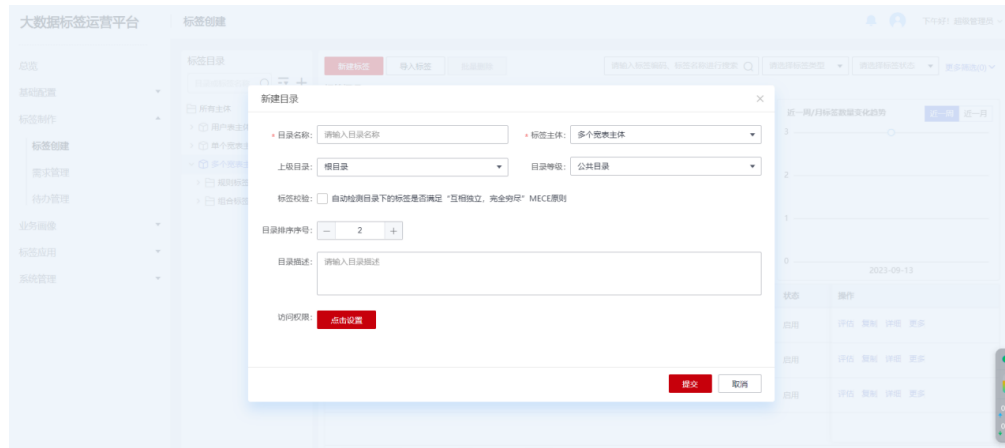
步骤1 标签制作-标签创建页面，对应的主体下面，新建目录

图 3-9 标签创建 1



步骤2 填写目录名称，设置访问权限(即在对应主体有权限访问的基础下，对目录进行权限设置)。

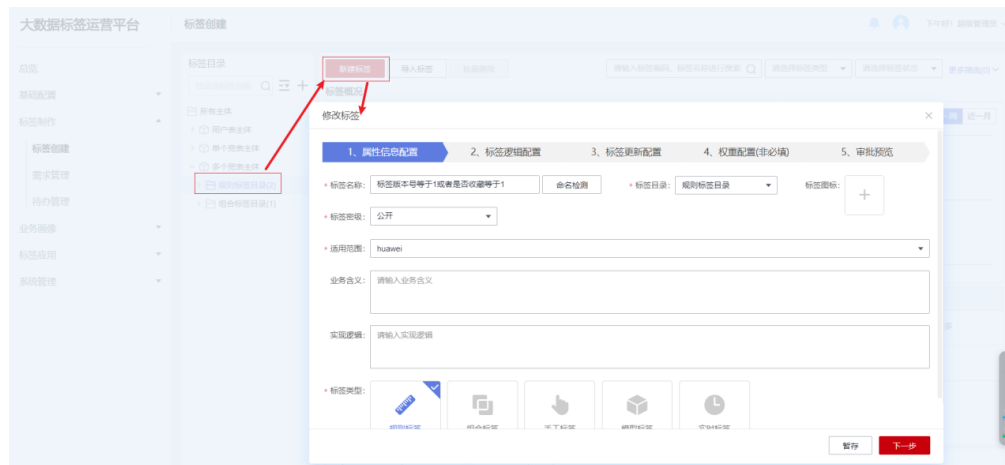
图 3-10 标签创建 2



选择新建的目录，单击标签创建，在新建标签页面

**步骤3** 基础信息sheet页，填写标签名称，选择标签类型，单击下一步

图 3-11 标签创建 3

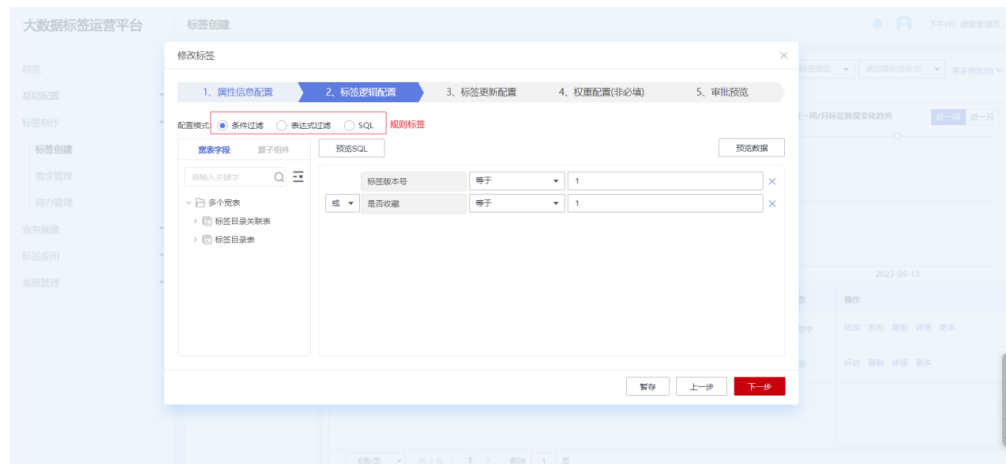


### 说明

标签类型是从标签生成方式的维度对标签进行划分的一种类型，选择后，标签逻辑配置页面不同，可根据选择的标签类型不同，进行配置不同的标签逻辑

**步骤4** 标签逻辑配置sheet页，根据选择的标签类型不同，进行配置不同的标签逻辑，单击下一步

图 3-12 标签创建 4



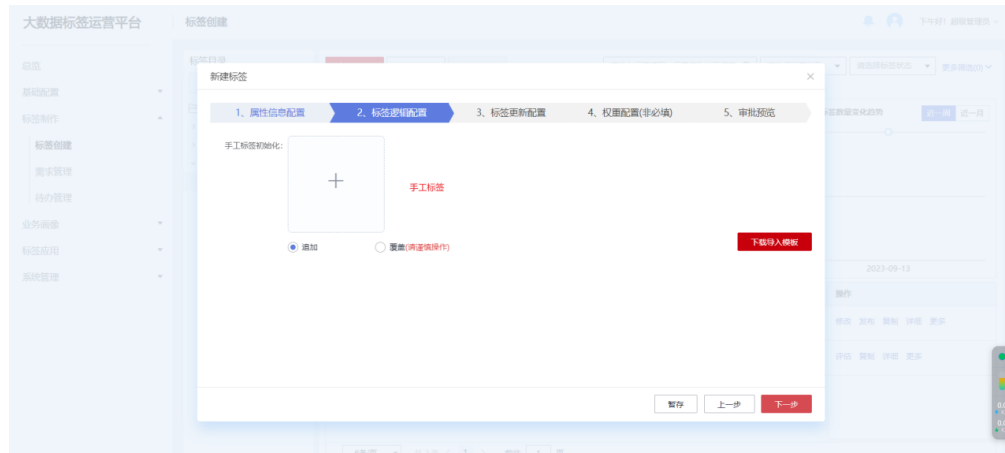
规则标签：是指通过简单规则运算而生成的标签。规则标签分为可视化和 SQL 模式两种规则标签。可视化规则标签，是直接通过拖拽的方式将某些字段拉入逻辑框中，进行基本字段的条件过滤来筛选数据；SQL 模式规则标签，是通过直接写出 SQL 语句来筛选数据。

图 3-13 标签创建 5



组合标签：是在现有的标签基础上，标签跟标签或标签跟字段或先标签跟标签然后再跟字段进行的运算，组合成新的标签

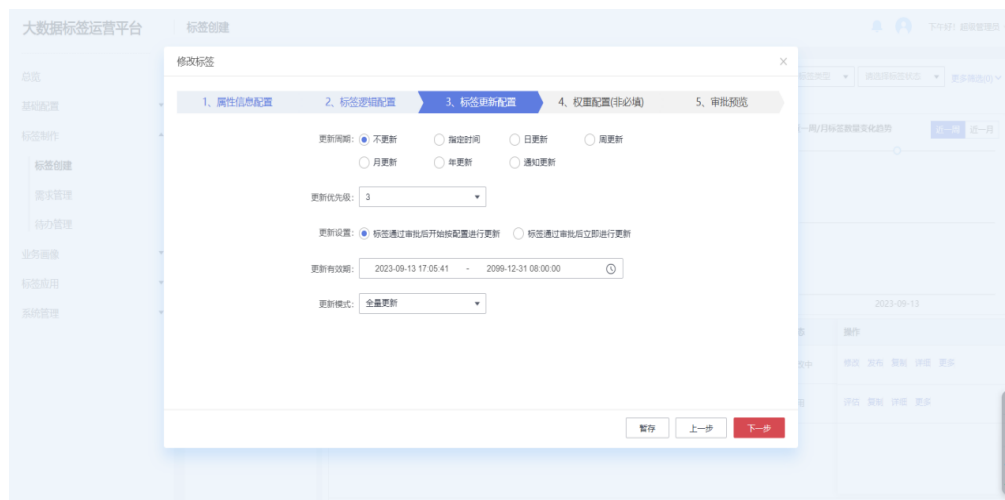
图 3-14 标签创建 6



手工标签：是指业务系统数据无法支撑标签规则运算，但具有业务应用价值，需以人为标记方式生成的标签

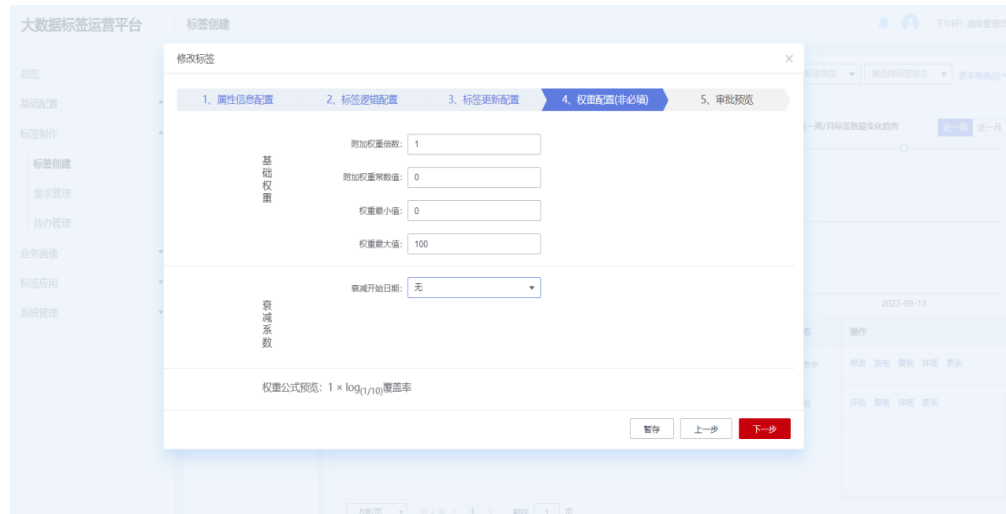
步骤5 更新配置sheet页，选择标签的更新周期

图 3-15 标签创建 7



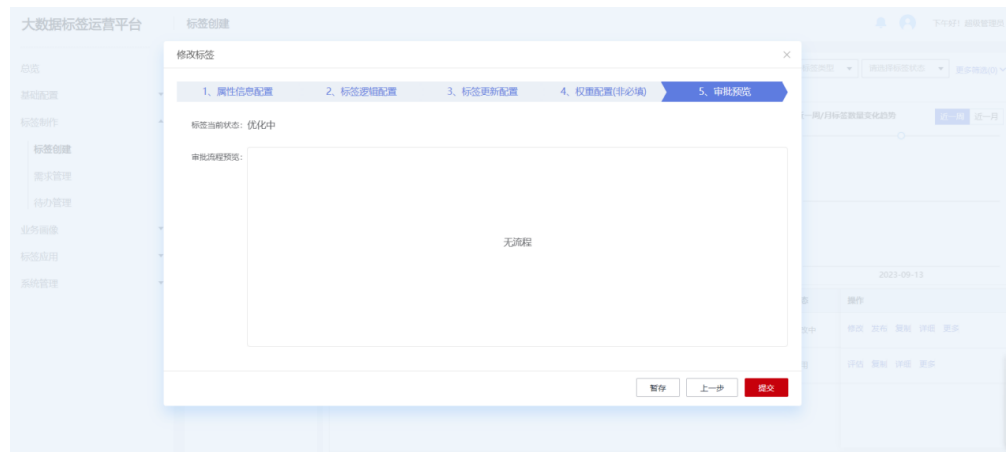
步骤6 权重配置sheet页，可默认，单击下一步

图 3-16 标签创建 8



**步骤7** 提交审批sheet页，当前主体配置的流程配置sheet页发布操作不需要审批时，则此处无流程，直接单击提交。标签创建成功，状态为启动状态

图 3-17 标签创建 9

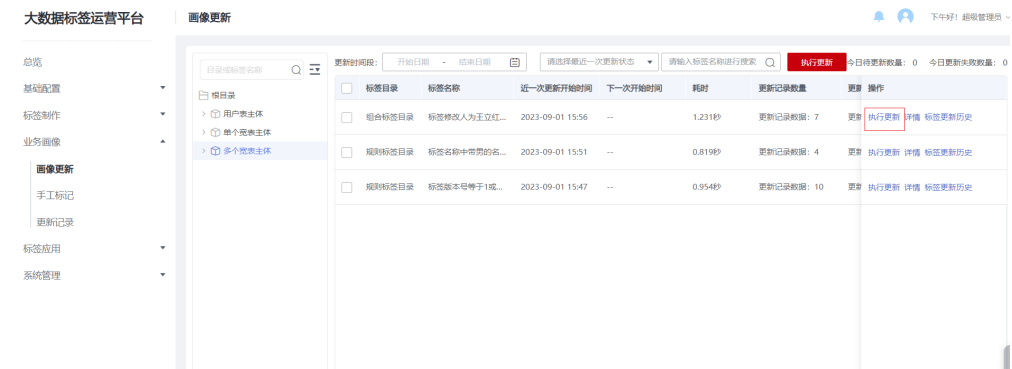


----结束

### 3.1.4 业务画像-画像更新

**步骤1** 业务画像-画像更新页面，左侧选择对应的目录，右侧展示对应的标签，单击执行更新，更新成功后，可查看详情和历史

图 3-18 画像更新 1

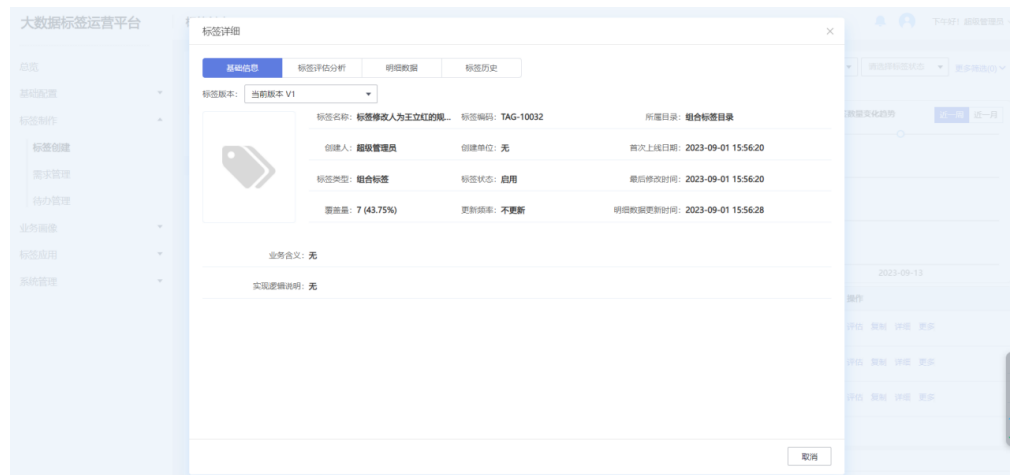


步骤2 更新成功后，标签创建页面，对应标签覆盖数量为中标的数量，单击详情可查看标签的基础信息、标签评估分析和具体明细数据

图 3-19 画像更新 2



图 3-20 画像更新 3



----结束

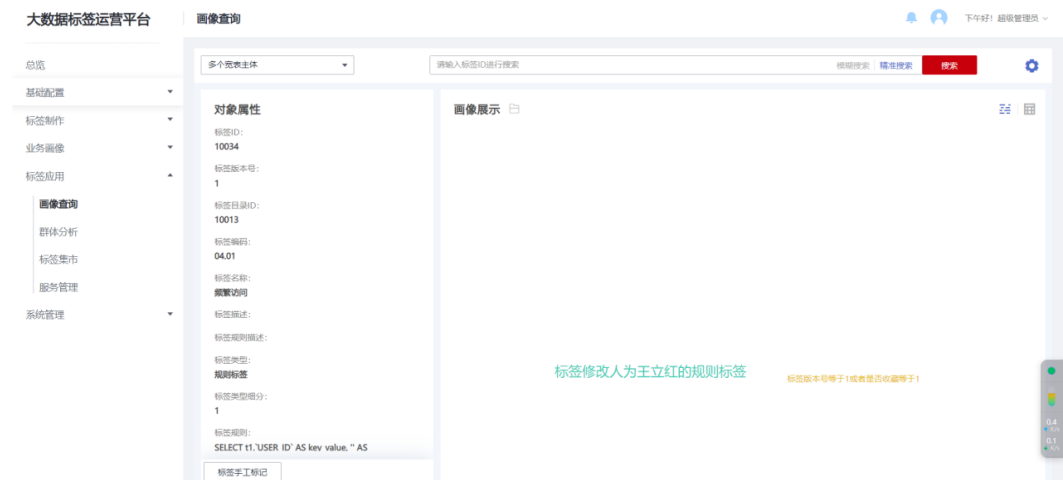


## 3.1.5 标签应用-画像查询与群体分析

### 画像查询

单击标签应用-画像查询，上方左侧选择对应的主体，右侧进行搜索，则下方左侧显示该搜索值的基本信息，下方右侧显示中标画像（当搜索值同时中标该主体下的多个标签时，则画像展示为多个）

图 3-21 画像查询



### 群体分析

单击标签应用-群体分析，在群体分析界面，选择左侧目录，单击筛选目录群体，可筛选标签和列的组合进行筛选数据，进行保存。后续可以直接查看该群体的分析结果和群体画像

图 3-22 群体分析 1

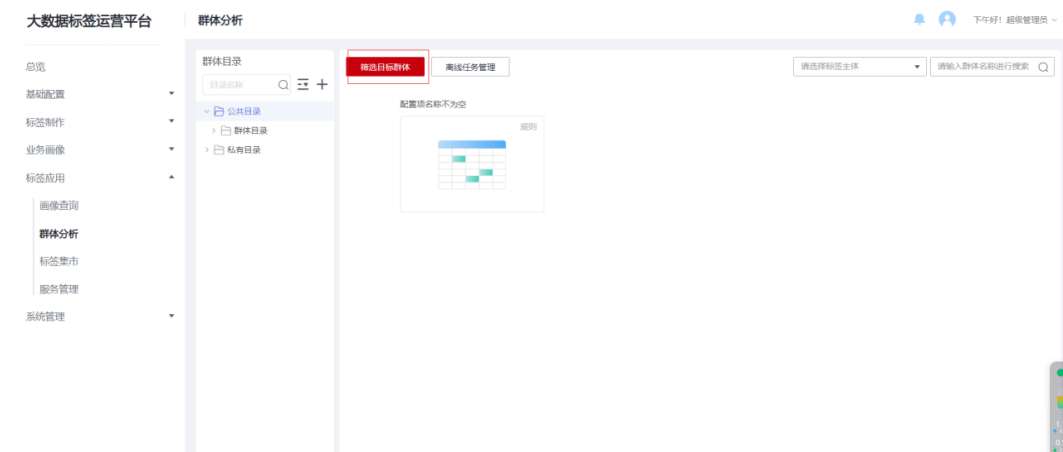


图 3-23 群体分析 2

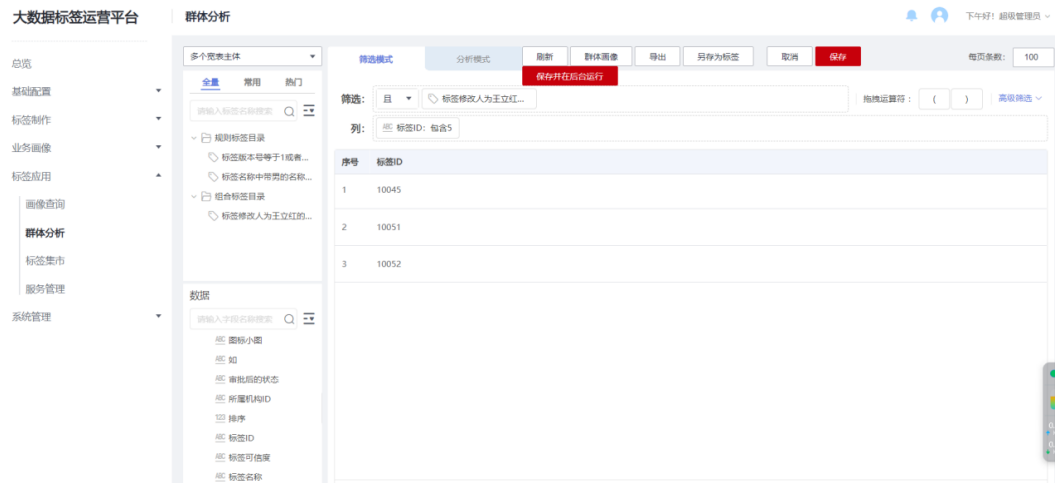
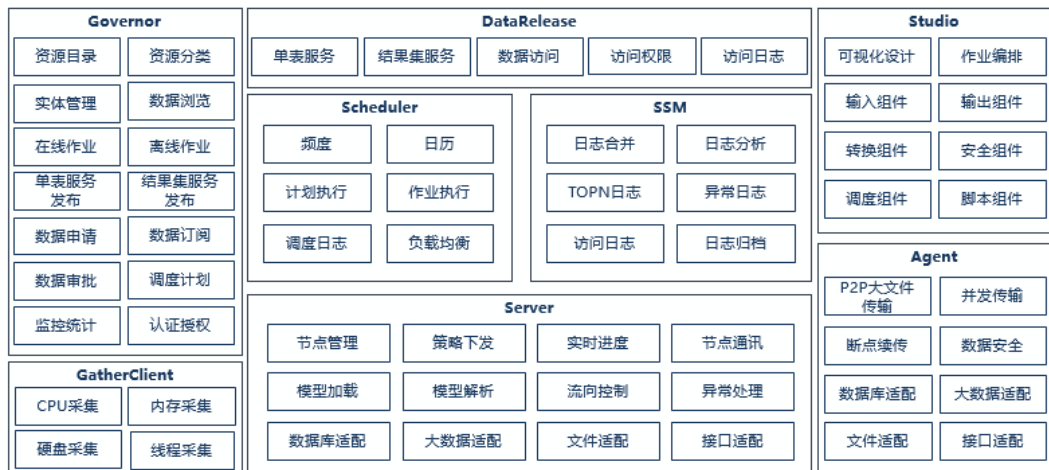


图 3-24 群体分析 3



## 3.2 数据共享交换管理操作流程

图 3-25 数据共享交换管理操作流程



- Governor（管控平台）

数据服务共享统一管控平台，提供资源目录、数据开发、数据使用、服务引擎管理、调度计划、统计监控、服务权限配置等系列功能，对数据资源和服务资源统一注册管理。

- Server（服务引擎）

批量作业服务及文件传输服务引擎，负责批量作业模型解析、批量作业执行、文件传输服务；提供多协议、多数据源适配支持；为服务运行提供高性能、高可靠的运行环境。

- SSM（服务状态监控）

提供日志解析及监控能力，对事前预警、事中告警、事后统计分析等功能提供后台支撑。

- Studio（可视化开发工具）

基于Eclipse技术开发的可视化批量作业设计器，提供可视化批量作业模型定义与调试、可视化的模型性能监控、元数据管理以及数据处理模型部署等功能。

- Agent（传输代理）

数据抽取及文件传输代理，支持数据库、大数据等数据源的数据抽取，支持任意节点及任意大小文件一对一、一对多的传输，并支持压缩、加密等多种文件传输策略。

- Scheduler（调度引擎）

作业调度引擎，支持作业与作业流的串行及并行调度，提供日历、频度、事件等多种规则的调度，为作业和作业流运行提供多样化的调度方式。

- DataRelease（数据发布引擎）

基于SpringBoot架构提供数据服务发布及数据访问能力，支持单表、结果集等形式的实时接口服务发布。

- GatherClient（资源采集客户端）

实时采集各引擎所在物理机的CPU、内存、磁盘、网络等指标，通过Governor进行展示。

### 3.2.1 数据资源目录管理

数据资源目录是以树型结构管理和展现的包括数据库、大数据、Web服务和文件等资源在内的注册信息。数据资源目录管理包括节点管理、数据库资源管理、大数据管理、接口服务管理、业务分类管理等。

数据管理员登录数据服务共享平台，对资源目录进行配置管理。

### 节点管理

节点是各类资源的挂接点，在本平台中，对应于公司总部及各分、子公司（或组织机构）。其主要功能包括：

1. 节点管理需要能够进行节点的新增、修改和删除；
2. 能够实现对节点的检索；

## 自动创建分区（源、前置区、共享区、消费方）

自动创建资源类型（如数据库、大数据、Web服务、文件等），资源类型可通过数据字典可自行定义；

支持资源管理功能，包括资源同步、校验等。

图 3-26 自动创建分区



## 数据库资源管理

数据管理员和开发人员可以通过平台注册、修改、删除源区、前置区、共享区、消费方数据库资源。数据库资源管理支持Oracle、SQLServer、MySQL关系型数据库连接；平台支持在资源目录中数据库资源节点可以展开，分层预览显示数据库表和字段并且可以添加业务描述。其主要功能包括：

1. 支持数据库资源管理功能：包括数据库资源的注册、修改、删除及连接测试等
2. 支持多种数据库类型：Oracle、MySQL、SQLServer等常用关系型数据库；
3. 支持数据库表、字段信息和数据的采集与预览展示；
4. 支持为数据库添加业务信息；
5. 数据库资源与数据服务开发功能的交互；
6. 支持数据库资源的检索。

图 3-27 数据库资源管理



## 文件资源管理

数据管理员和开发人员可以通过数据服务共享平台注册、修改、删除源区、前置区、共享区、消费方的文件资源，文件资源可以配置Agent，用以文件传输。其主要功能包括：

1. 文件资源管理，包括文件资源的注册、修改、删除等；
2. 文件资源注册页面自动关联信息和用户的输入校验；
3. 文件资源与数据服务开发功能的交互；
4. 文件传输Agent，目录授权和目录关联到Agent；
5. 支持文件资源的检索。

图 3-28 文件资源管理



## 接口服务管理

数据服务共享平台通过数据服务发布组件发布的数据服务在此进行管理，其主要功能包括：

1. Web服务资源管理，包括Web服务资源修改，删除功能；
2. Web服务资源注册页面的自动关联信息；
3. Web服务资源与数据服务开发功能的交互，数据服务开发后自动注册在此处；
4. 支持Web服务资源的检索。



## 大数据资源管理

数据管理员可以通过平台注册、修改、删除大数据资源。大数据服务管理支持HBase、Hive等大数据连接。其主要功能包括：

1. 大数据资源管理，包括大数据资源的注册、修改、删除及连接测试等；
2. 支持HBase、Hive的大数据连接；
3. 支持HBase、Hive元数据的采集与展示；
4. 大数据库与数据服务开发功能的交互；
5. 支持数据库资源的检索。

图 3-29 大数据资源管理



## 资源分类管理

数据管理员根据业务主题和业务实体（分类及名称可根据需求定义），实现对数据库、大数据、Web服务、文件资源的分类管理。其主要功能包括：

1. 业务主题管理，包括业务主题的新增，修改，删除功能；
2. 业务实体管理，包括业务实体的新增，修改，删除功能；
3. 业务实体资源分配和资源查看，其中资源分配要求可实现数据库、文件、接口和大数据的关联；
4. 支持分类资源的检索；
5. 业务实体已分配的资源，支持快速的服务发布。

图 3-30 资源分类管理



## 3.2.2 数据使用

### 消费方注册

消费者可通过系统管理员进行账号的创建和权限分配。主要功能包括：

1. 基本信息维护；
2. 密码初始化、修改；
3. 消费者权限分配。

图 3-31 消费方注册



### 数据服务使用

#### 数据资源申请

消费方通过资源申请方式申请主题分类资源树中的数据实体资源，相对平台而言“拉”的方式，消费方申请资源后并且审批通过，向消费方开放Web服务及数据信息，由消费方主动获取资源，主要功能包括：

1. 数据资源分类展现，数据资源搜索；
2. 消费方申请Web服务或DB类型资源；
3. 业务实体汇总；
4. 数据库资源预览；
5. 支持以web服务和表格下载的方式提供数据；
6. 支持单表、多表、结果集数据服务发布（发布RESTful服务）；
7. 资源申请信息维护；
8. 已申请记录列表展示。

图 3-32 数据服务使用 1

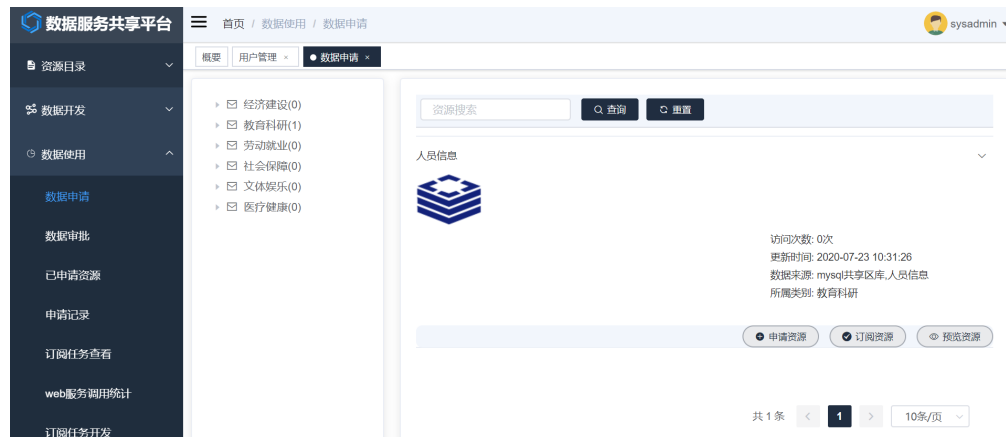


图 3-33 数据服务使用 2



## 数据资源订阅

消费方通过资源订阅在主题分类资源树中的数据实体资源中订阅资源，相对数据服务共享平台而言“推”的方式，消费方订阅资源，向平台提供数据库、文件目录地址，订阅任务审批通过后，由平台向消费方推送资源，主要功能包括：

1. 数据资源分类展现，数据资源搜索；
2. 业务实体汇总；
3. 支持订阅文件和数据库两类资源；
4. 支持数据库资源转文件；
5. 资源订阅申请信息维护；
6. 订阅记录列表展示；
7. 数据资源预览。



图 3-34 数据资源订阅 1

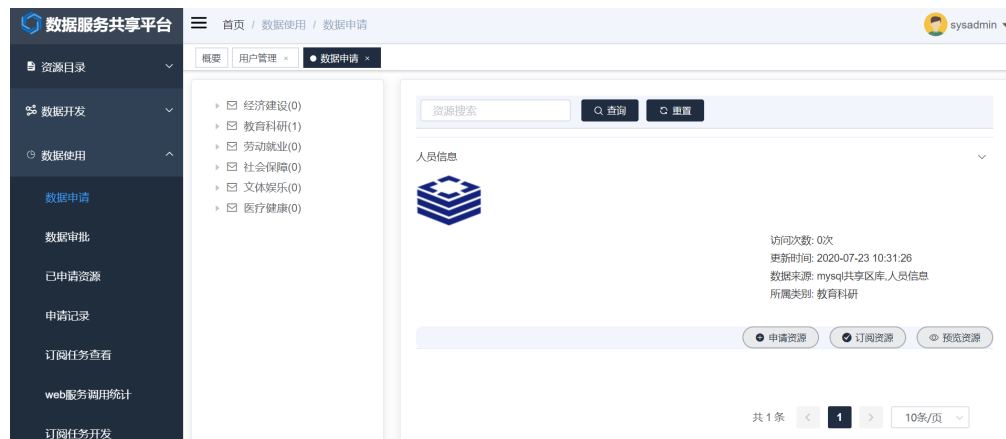


图 3-35 数据资源订阅 2

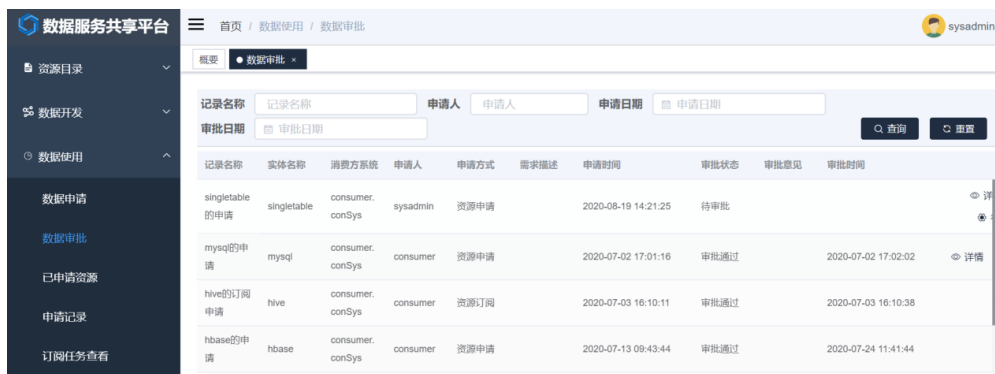


## 数据服务审批

消费者提交数据资源申请（包括申请和订阅两种方式）后，由数据管理员对申请记录进行审批。申请方式（拉）的申请记录审批通过，消费者可以直接查看到相关资源（包括Web服务、数据信息）的地址及获取方式信息；订阅方式的申请记录审批通过后需要数据开发人员根据需要开发相应的数据服务（同时需要消费者注册其对应的数据库地址、文件目录，以向其推送）开发完成后，提供给消费者，开发人员查看申请记录列表。其主要功能包括：

1. 申请记录列表查询；
2. 申请信息查看；
3. 数据服务审批，审批信息维护；
4. 服务审批流程，审批过程中对数据行列级权限进行控制；
5. 开发人员查看审批结果；
6. 订阅方式订阅记录服务开发-在线开发；
7. 订阅方式申请记录服务开发-离线开发。

图 3-36 数据服务审批



### 3.2.3 数据服务开发

数据服务共享平台（DSP）数据服务开发可分为在线开发和离线开发两种。离线开发是只通过Studio进行数据模型开发和Web服务的开发，离线开发的模型只需通过平台部署即可，此处不再赘述。本节内容主要是面向在线开发。

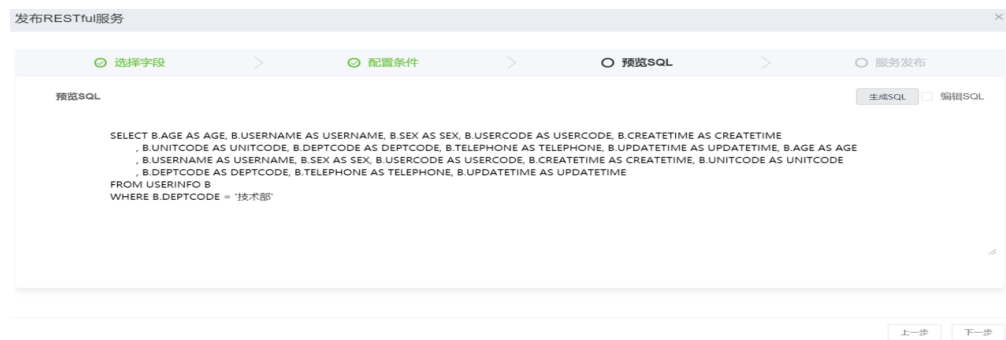
#### 实时数据服务

- 单表服务  
通过平台数据资源目录，可以关联单个表资源发布为数据服务，通过界面化配置选择需要发布的表字段、配置查询条件、自动生成SQL（支持对SQL语句进行编辑）几个步骤完成服务发布，并支持服务的自动注册发布。
- 结果集服务  
支持通过平台数据资源目录，可以关联多张表进行数据组装形成新的结果集，并支持对结果集的自定义，并支持按定义的结果集进行服务的自动注册发布。

图 3-37 实时数据服务 1



图 3-38 实时数据服务 2



## 批量数据服务

- 结构化

- a. 在线作业服务开发

对于在线开发，支持通过页面的方式，根据已经导入到平台中的作业模板配置映射关系，而不用通过studio进行模型开发即可完成批量数据处理工作。其整体过程为：选择模板选择区选择库批量表配置变量配置服务器配，平台在线开发基础模板包含以下几类：

- 全量同步；
- HBase到文件同步
- 全文对比同步、
- 触发器同步
- 时间戳同步
- 数据库到文件同步
- 全量同步
- 文件到数据库同步
- MySQLtoHBASE数据脱敏
- CanalClientToHBase同步
- DBToHbase
- MongoDBToHBaseSingleTable
- MongoDBToHBase
- DBToDBDesensitization。

图 3-39 在线作业服务开发

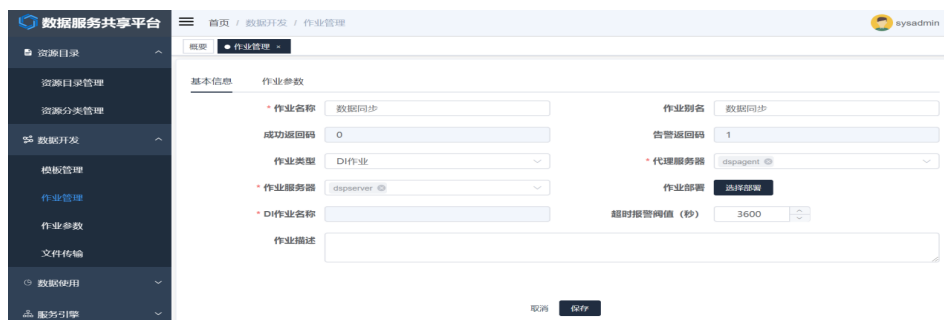


- b. 离线作业服务开发

对于复杂的作业服务，提供独立的开发工具(DSP Studio)，使开发期和运行期分开，支持多种协议，可通过使用图形化界面进行快速开发，满足用户个性化的需求。功能上与平台具备从设计、开发、组装、调试、维护、部署、管理和发布一体的集成开发环境。同时，具有向导、视图和编辑器等工具供开发人员在开发过程中使用。

对于离线作业开发，支持离线数据模型部署在数据服务共享平台。

图 3-40 离线作业服务开发

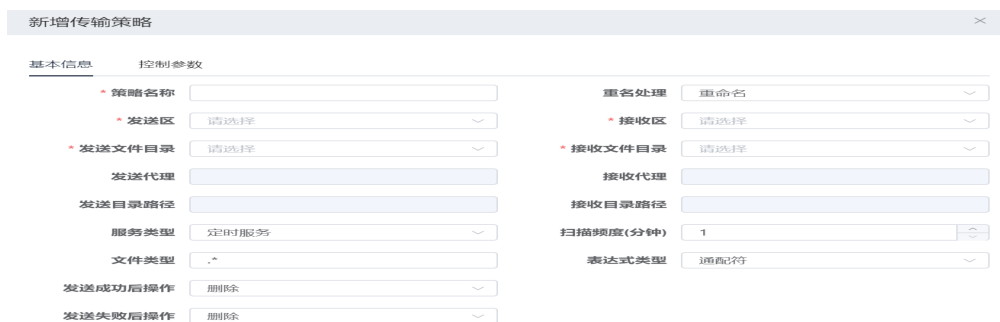


- 非结构化
  - 非结构化数据服务支持文件的传输配置。通过文件传输配置能够将相应的文件资源传输给消费方。其主要功能包括：
    - 新建文件传输配置。
    - 编辑文件传输配置。
    - 删除文件传输配置。
  - 文件传输配置支持一对一、一对多传输；支持文件传输策略的配置。

图 3-41 非结构化 1



图 3-42 非结构化 2



## 订阅任务开发

数据开发人员登录数据服务共享平台，对数据管理员已经审批通过的消费方订阅任务进行开发，订阅任务开发分为离线开发和在线开发，且离线开发和在线开发功能与章节结构化相同。

图 3-43 订阅任务开发

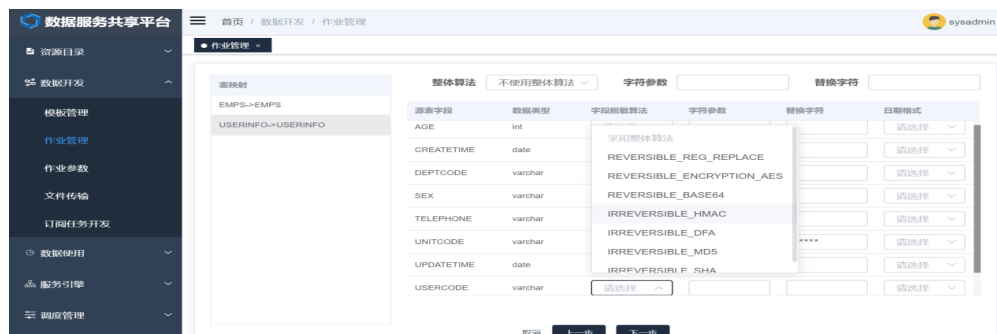


## 脱敏算法配置

数据服务共享平台数据脱敏模板内置多种数据脱敏算法，可实现替换、加密等处理。其主要功能包括：

- 目标数据源选择；
- 算法配置
- -DFA算法
- -AES算法
- -BASE64算法
- -MD5算法
- 替换处理。

图 3-44 脱敏算法配置



## 作业模板开发

作业模板实际上是参数化的数据模型。在线作业开发中用到的作业类型都是通过对模板的开发产生的，模板的开发是通过studio开发dbr或dsr模板文件，开发的过程与ETL模型开发相同，不同的是模板开发对数据源和表都有新的标记定义，如：数据源填写时用%{source}进行标记，表填写时用[sourcetablename]进行标记。在开发场景中使用到该模板时可对数据源和表的变量进行映射。

在作业中需要进行数据转换、数据质量检验和数据脱敏的时候需要使用到业务调度dsr来进行模型组合完成，dsr能够在模型之间传递变量和结果集。这个是作业流目前所做不到的，需要共享平台提供底层工程化的支撑来完成有前后条件的这种模型的执行。

当前基础模板包含以下几类：

- 全量同步；
- HBase到文件同步
- 全文对比同步、
- 触发器同步
- 时间戳同步
- 数据库到文件同步
- 全量同步
- 文件到数据库同步
- MySQLtoHBASE数据脱敏
- CanalClientToHBase同步
- DBToHbase
- MongoDBToHBaseSingleTable
- MongoDBToHBase
- DBToDBDesensitization。

除上述模板之外，可依据项目实际需求进行开发。采用模板化的在线开发，能简化使用难度，不需要专业开发人员进行模型的开发和设计。

图 3-45 作业模板开发



## 3.2.4 服务控制

### 系统调用关系管

消费者访问数据服务共享平台中web服务共享资源，数据管理员给消费者分配web服务的访问权限。其主要功能包括：

- 支持web服务调用关系管理；
- 支持Server端调用关系控制处理；
- 支持Server端调用关系控制告警信息处理；
- 支持消费者与Web服务调用关系控制关系配置。

图 3-46 系统调用关系管



## 服务状态管理

消费者访问数据服务共享平台中Web服务共享资源，数据管理员对服务状态进行管理，控制服务能否被消费方访问调用。

支持对已发布的web服务进行启动、暂停、恢复、注销；

支持按照操作名称、操作编码、接口名称、接口编码查询。

图 3-47 服务状态管理



## 系统 IP 地址管理

消费者访问数据服务共享平台中web服务资源，数据管理员给消费者分配web服务的IP白名单访问权限，对于不在IP此名单中的用户，不能访问数据服务资源。其主要功能包括：

- web服务IP白名单控制管理；
- Server端IP白名单控制处理；
- Server端IP白名单控制告警信息处理；
- 消费者与Web服务IP白名单控制关系配置。

图 3-48 系统 IP 地址管理



## 3.2.5 调度管理

### 作业策略管理

支持自定义作业执行的频度、日历信息，可以为不同作业定义不同调度频率和日历。

图 3-49 作业策略管理 1



图 3-50 作业策略管理 2



### 作业计划场景

数据服务共享平台为开发人员提供按照计划自动执行作业的能力，支持在计划中配置自定义的频度、日历、作业、作业执行优先级等内容。



图 3-51 作业计划场景 1

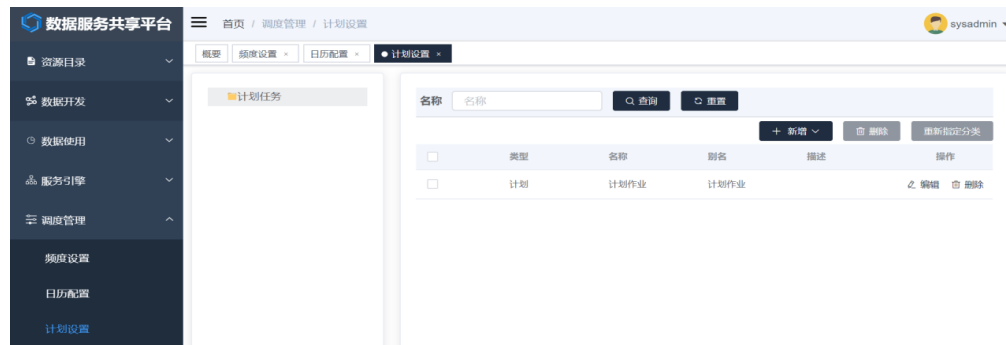
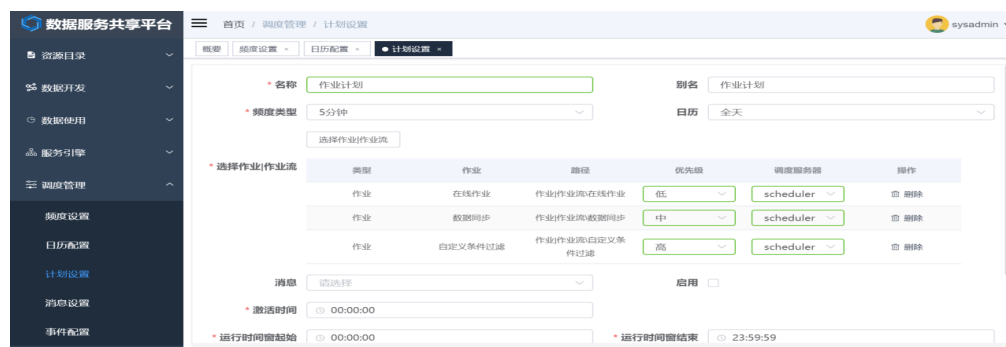


图 3-52 作业计划场景 2



## 作业图形编排

数据服务共享平台为开发人员提供图形编排调度作业的能力，包括作业的设计优先级，依赖关系等。

图 3-53 作业图形编排

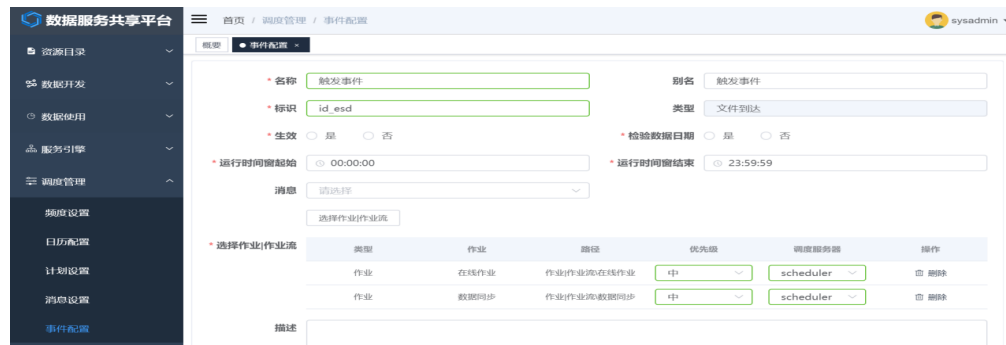


## 事件触发

数据服务共享平台为开发人员提供基本的调度管理触发功能，支持通过事件触发作业的执行，通过文件到达的方式触发作业、作业流的运行，其主要功能包括：

- 支持按照名称进行查询；
- 时间基本信息维护、修改、查看；
- 与已创建的作业、作业流关联。

图 3-54 事件触发



### 3.2.6 运行监控

#### 交换监控

数据服务共享平台上的作业和文件传输运行状态的监控，其中作业状态展示最近一次作业的运行状态数据，文件传输则显示当前的实时传输进度。

- 前置交换监控查询浏览  
从前置区角度查看最近一次作业运行状态  
从前置区角度查看文件传输实时进度
- 共享交换监控查询浏览  
从共享区角度查看最近一次作业运行状态  
从共享区角度查看文件传输实时进度

图 3-55 交换监控



#### 消费者监控

监控已经注册到系统当中的消费者提出的数据申请记录以及实际进行共享区数据的使用记录。使用记录包括了：

- 通过作业将共享区资源传输到消费者指定数据源中的数据量；
- 通过文件传输推送到消费者指定文件夹的文件记录。

主要功能包括：

根据消费方进行文件、作业数据使用量统计查询

图 3-56 消费者监控



## 计划、作业流监控

对作业计划的执行情况进行监控，对作业流执行情况监控，主要包括：

- 作业计划信息查看
- 作业计划执行历史信息查看
- 作业计划执行日志信息查看
- 作业流详情信息查看
- 作业流执行历史信息查看
- 作业流执行详细日志查看

图 3-57 计划、作业流监控 1

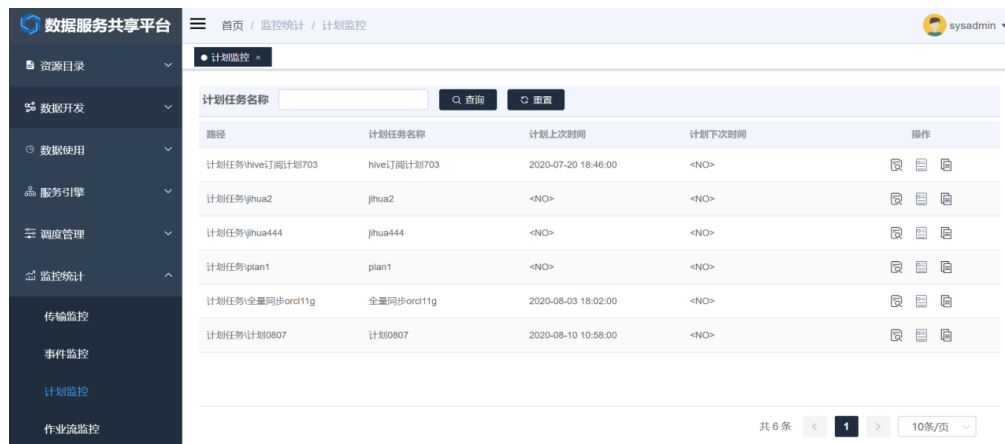


图 3-58 计划、作业流监控 2

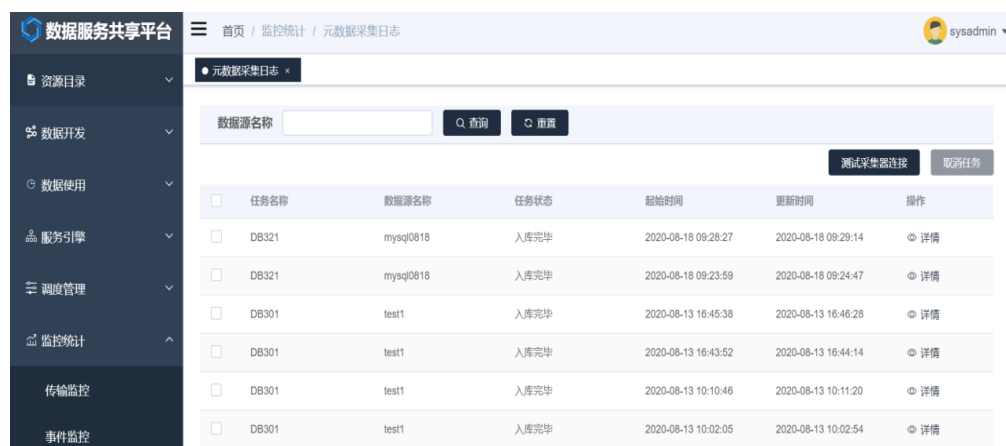


## 故障告警监控

数据服务共享平台中作业、文件传输任务和Web服务在中执行时，可能存在着运行时的错误和异常，这些异常信息可以在故障告警监控中进行浏览，浏览告警信息可以查阅到故障的服务和具体的作业、传输任务，通过这些信息为排除故障做出调整提供参考依据。其主要功能包括：

- 收集各个服务引擎的故障信息
  - a. 存储来自服务引擎接口的故障信息
  - b. 故障信息的定时清理
- 查询故障信息  
查询故障信息，其中包含足以定位故障位置的信息
- 异常信息预警  
对出现故障的信息，通过消息、短信或邮件的方式向管理人员进行提示
- 元数据采集监控
  - a. 提供采集日志界面，对元数据采集结果进行监控，主要包括：
  - b. 采集器连接测试
  - c. 采集任务日志查询
  - d. 采集任务详情查看

图 3-59 故障告警监控



## 历史记录查询

可以查询共享区或者交换区中作业、文件传输任务的执行历史，历史记录中记录每次作业或者任务执行的最终状态。其主要功能包括：

- 文件传输历史记录查询
- 作业历史记录查询

图 3-60 历史记录查询

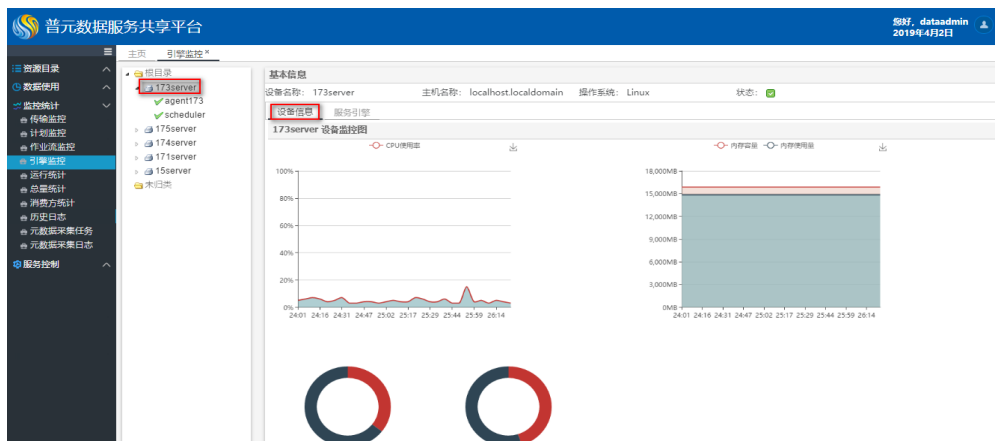


## 物理资源监控

交换平台所有物理设备之间的通信关系，服务引擎所在设备的物理资源使用情况，监控的物理资源包括CPU、内存、硬盘。其主要功能包括：

- 收集各个物理设备资源的使用状况
  - 收集CPU负载
  - 收集内存负载
  - 收集存储设备信息
  - 物理资源使用状况的刷新
- 展示物理资源使用数据
  - 图表方式展示各个物理设备的运行状况；
- 物理设备信息维护
  - 物理设备信息的注册、注销、更新
  - 物理设备的浏览

图 3-61 物理资源监控

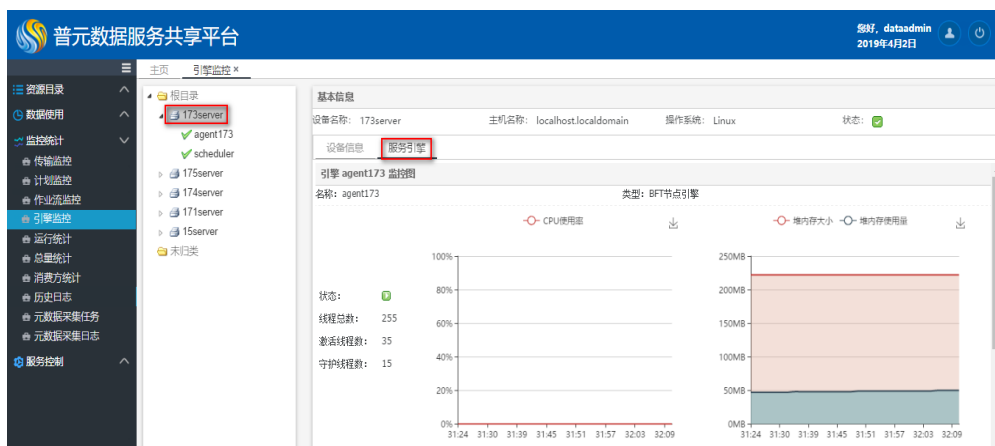


## 服务引擎监控

部署在物理设备上的服务引擎运行状态监控，以及各个引擎产生的系统日志。运行状态和日志查看为排查进程异常和服务错误提供依据。引擎状态数据包括了服务进程是否正常、JVM运行情况等。其主要功能包括：

- 收集服务引擎运行状态数据
  - 收集服务引擎JVM内存区域状态
  - 收集服务引擎的进程状态
  - 收集服务引擎JVM的CPU负载状态
- 展示物理设备上的服务引擎运行状态
  - 服务引擎域部署的物理设备之间关联
  - 提供服务引擎状态接口，定时获取状态信息
  - 图表方式浏览指定设备上的所有服务引擎信息
- 服务引擎系统日志查看
  - web浏览器中查看指定服务引擎的系统日志
  - 服务引擎系统日志集中收集

图 3-62 服务引擎监控



## 3.2.7 统计分析

### 交换总量

对源和前置区或者前置区和共享区、共享区和消费者之间数据交换的数据量进行监控。对作业执行的数据总量（条数）等进行监控统计。通过从提供者的纬度来对数据指标进行显示。其主要功能包括：

- 节点到节点之间数据交换总量统计
- 库到库之间数据交换总量统计
- 表和表之间数据交换总量统计
- 缓存统计（定期计算统计维度指标）

图 3-63 交换总量



### 作业统计

数据管理员根据配置好的源和前置区、前置区和共享区之间的作业模型调度进行数据交换，对作业执行情况进行统计。其主要功能包括：

- 基础数据指标计算和统计
- 页面图表展示
- 页面查询列表展示

图 3-64 作业统计



## 文件统计

数据管理员根据配置好的源和前置区，前置区和共享区之间进行文件传输，对文件传输的数据进行监控。其主要功能包括：

- 基础数据指标计算和统计
- 页面图表展示
- 页面查询列表展示

图 3-65 文件统计



## 消费方数据统计

消费方数据统计主要是消费者和共享区之间进行数据交换，对传输的数据量进行监控和统计，主要针对DI作业执行情况、文件传输、Web服务调用情况进行监控和统计。其主要功能包括：

- DI作业运行产生数据流转的统计和分析
  - 基础数据指标计算和统计
  - 页面图表展示
  - 页面查询列表展示



- 文件传输产生的数据进行统计和分析  
历史数据统计分析  
页面图表展示  
页面查询列表展示
- Web服务接口产生数据进行统计和分析  
web服务调用过程中基础数据采集  
基础历史数据指标计算和统计  
页面图表展示  
页面查询列表展示

图 3-66 消费方数据统计

消费方	业务实体名称	作业名称	新增量	更新量	删除量	错误数	成功数	失败数	操作
sysadmin		hive2bx在线	0	0	0	0	0	0	详情
sysadmin		hbase2bx0708	0	0	0	0	0	1	详情
sysadmin		mysql02hbase	0	0	0	0	0	1	详情

## Web 服务统计

对服务Web服务接口调用情况提供统计查询界面，方便用户对服务的运行情况进行统计监控。其主要功能包括：

- 服务消费方维度Web服务统计
- 服务提供者维度Web服务统计
- Web服务调用情况统计
- 查询缓存

图 3-67 Web 服务统计



# 4 实施步骤

- 4.1 数据标签库管理实施步骤
- 4.2 数据共享交换管理实施步骤

## 4.1 数据标签库管理实施步骤

图 4-1 安装流程图



### 安装前准备

表 4-1 准备硬件与操作系统

服务器	配置	数量	用途
Linux服务器	8线程cpu/32G 300G	2	部署标签画像应用集群
Linux服务器	16线程cpu/32G 300G	2	部署标签画像接口集群

表 4-2 准备软环境

环境	软件/工具	版本	备注
基础环境	JDK	华为jdk-8u202	Java程序运行时环境
	Tomcat	8.5.63	程序中间件，内嵌在应用服务中
数据平台	MySQL	5.7.22+	配置库
	DWS	6.5.1+	运算库

表 4-3 准备工具和软件

软件工具	作用
SSH远程登录工具 (例如PuTTY)	跨平台远程访问工具。用于在软件安装过程中在Windows系统上访问各linux服务器进行安装操作
WinSCP	跨平台文件传输工具。用于在Windows系统和Linux系统间传输文件，例如上传软件包、上传配置文件等操作
网络浏览器	用于登录标签画像平台，版本要求： Google Chrome 建议使用Google Chrome 78以上版本 Internet Explorer 建议使用Internet Explorer 11以上版本 Mozilla Firefox 建议使用□Mozilla Firefox 80以上版本
7-zip	7-zip 用于解压“*.zip”和“*.rar”文件 建议使用7-Zip 16.04版本 建议使用WinRAR 5.1版本

## 安装过程

- **MySQL数据库初始化**

- 📖 **说明**

- MySQL数据必须禁用大小写敏感

- 登录MySQL数据库按以下顺序执行初始化脚本文件：

- 1\_mysql\_create.sql
    - 2\_mysql\_init\_data.sql

- **DWS数据库初始化**

- a. 高斯数据库根据使用的模式不一样，初始化操作也不一样，分为三种情况：  
使用以用户名来命名的模式：登录高斯数据库，连接以用户名命名的模式，按顺序执行以下初始化脚本：3\_gauss\_create.sql

使用public模式：登录高斯数据库，连接public模式，按顺序执行以下初始化脚本：3\_gauss\_create.sql

使用用户名跟public以外的模式：登录高斯数据库，连接模式，按顺序执行以下初始化脚本：3\_gauss\_create.sql

- b. 修改dws连接用户默认使用的模式，请执行以下sql:

```
ALTER USER <user_name> SET search_path to <schemaName>  
<user_name>为数据库账号名，<schemaName>为模式名
```

- **标签应用安装**

检查localhost的网络

执行命令查看hostname

hostname

```
[root@localhost home]# hostname  
localhost.localdomain  
[root@localhost home]#
```

ping localhost.localdomain

```
[root@localhost home]# ping localhost.localdomain  
PING localhost (127.0.0.1) 56(84) bytes of data.  
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.048 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.101 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.125 ms  
^C  
--- localhost ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
```

能ping通则继续往下操作，无法ping通则执行命令

hostname 127.0.0.1

- **检查服务器时间**

服务器的时间、市区会影响系统的准确性，因此要检查时间是否准确，时区是否为东八区，执行命令：

date -R

```
7dev/sda2 247G 18G 217G 8% /home  
[root@localhost Asia]# date -R  
Fri, 27 Sep 2019 17:40:27 +0800  
[root@localhost Asia]#
```

最后字段为“+0800”则表示时区为东八区，如果不是，或者时间不准确，则需要联系服务器管理员修改。

上传程序包以及配置文件

上传应用部署包taglib.tar.gz文件到服务器目标路径，包含了程序包，执行解压命令：tar -zxvf taglib.tar.gz

注：本文档以安装到/home位置进行说明。

## 安装后检查

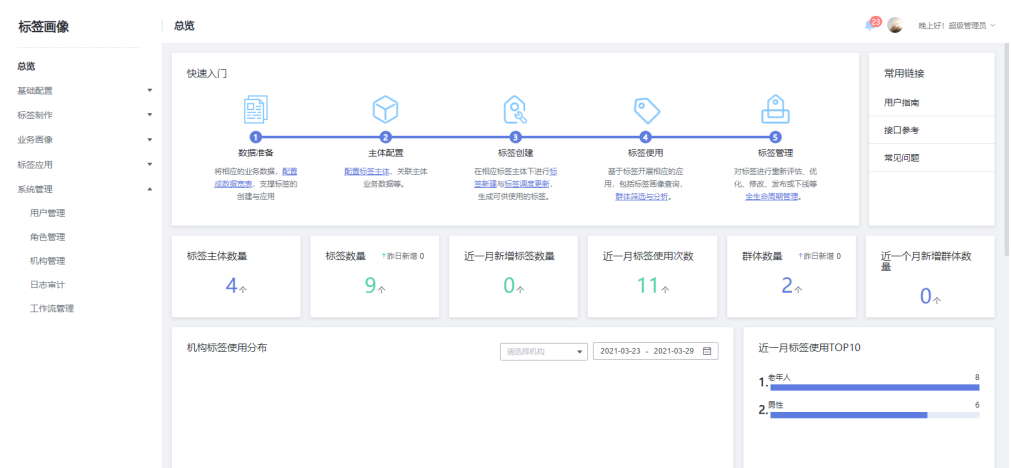
- 启动应用  
参考章节：软件运维>启动应用
- 功能调试  
使用浏览器访问http://{ip}: 18089/taglib

图 4-2 登录界面



输入正确的账号和密码可以登录成功。

图 4-3 系统总览



成功登录

## 软件运维

- 启动应用

打开taglibr/bin目录，执行启动脚本命令：

```
./startup.sh
```

启动过程大约需要3分钟左右。

- 停止应用

执行命令查看taglibr进程号pid，然后使用kill命令停止进程，命令如下：

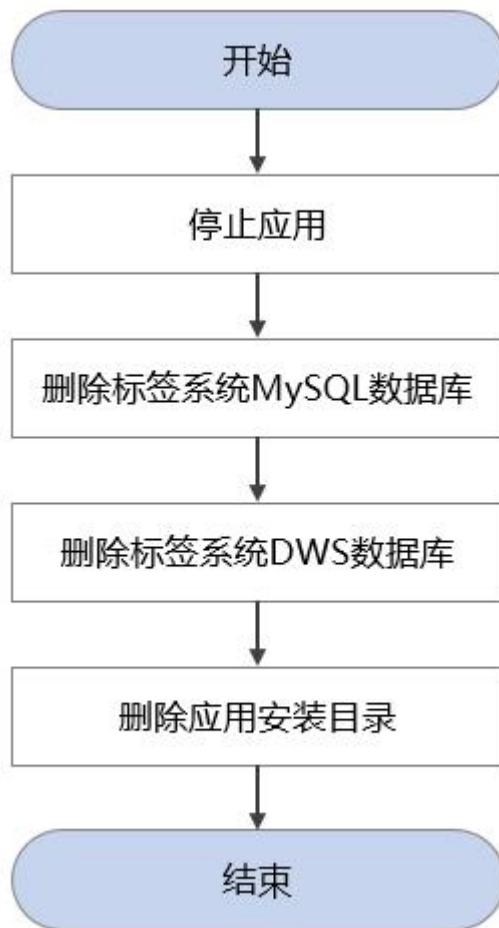
```
ps -ef|grep taglibr
```

```
Welcome to Huawei Cloud Service
PID
[root@ecs-puyuan-007 ~]# ps -ef|grep tomcat
root    11175    1 38 16:51 pts/0    00:24:23 /home/
va.util.logging.manager=org.apache.juli.ClassLoaderLogg
urces -Dorg.apache.catalina.security.SecurityListener.L
ddress=47008 -Dignore.endorsed.dirs= -classpath /home/
s_bdatag -Dcatalina.home=/home/rms_gauss_bdatag -Djava
root    11956 10257  0 17:54 pts/5    00:00:00 grep -
[root@ecs-puyuan-007 ~]# █
```

kill -9 PID

## 软件卸载

图 4-4 卸载流程



**步骤1** 停止应用，参考章节：软件运维>停止应用

**步骤2** 登录MySQL数据库，执行删除库命令：DROP DATABASE 标签画像MySQL数据库名称；

**步骤3** 登录DWS数据库，执行删除命令：DROP DATABASE 标签画像DWS数据库名称



**步骤4** 登录应用服务器：rm -rf 标签画像安装的文件夹  
----结束

## 4.2 数据共享交换管理实施步骤

### 4.2.1 安装前准备

已下载需要安装组件的安装包，\*.tar 安装包适用于 Linux 操作系统，\*.zip 安装包适用于 Windows 操作系统。

表 4-4 安装包清单

软件包名称	软件包描述
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_Governor_Linux_64.tar</li><li>• Primeton_DSP_7.0_Governor_Windows_64.zip</li></ul>	管控平台安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_Server_Linux_64.tar</li><li>• Primeton_DSP_7.0_Server_Windows_64.zip</li></ul>	服务引擎安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_SSM_Linux_64.tar</li><li>• Primeton_DSP_7.0_SSM_Windows_64.zip</li></ul>	服务状态监控安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_Studio_Windows_64.zip</li></ul>	可视化开发环境安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_Agent_Linux_64.tar</li><li>• Primeton_DSP_7.0_Agent_Windows_64.zip</li></ul>	传输代理安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_Scheduler_Linux_64.tar</li><li>• Primeton_DSP_7.0_Scheduler_Windows_64.zip</li></ul>	调度引擎安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_DataRelease_Linux_64.tar</li><li>• Primeton_DSP_7.0_DataRelease_Windows_64.zip</li></ul>	数据发布引擎安装包
<ul style="list-style-type: none"><li>• Primeton_DSP_7.0_GatherClient_Linux_64.tar</li><li>• Primeton_DSP_7.0_GatherClient_Windows_64.zip</li></ul>	资源采集客户端安装包
primeton-installer-1.0.0.zip	哑安装软件包

#### 许可证申请与安装

#### 申请

开发永久许可版：软件包自带，自带的License满足如下控制项条件

表 4-5 开发永久许可版

模块	控制项	key	默认值
Governor	并发访问数	concurrency	10
Server	作业并行执行数	jobconcurrency	2
	文件传输Agent登录数	agentconcurrentcy	3
DataRelease	并发访问数	serviceconcurrentcy	3
Agent	作业并行执行数	jobconcurrency	2

正式版：如果自带的License不满足实际需求，需要申请正式许可证，请联系普元销售人员，申请正式版许可证文件

## 安装

**步骤1** 将申请到的许可证“primetonlicense.xml”文件复制到对应产品组件的license存放位置，分别如下

1. Governor: {DSP Governor 安装目录}/governor/apache-tomcat-8.5.51/webapps/dsp/WEB-INF/classes/META-INF/\_srv

### 📖 说明

后续如果替换许可请放置：(/governor/apps\_config/DSP/license路径下)

2. Server: {DSP Server安装目录}/dspserver
3. DataRelease: {DSP DataRelease安装目录}/apache-tomcat-8.5.51/webapps/datarelease/WEB-INF/classes
4. Agent: {DSP Agent安装目录}/dspagent

**步骤2** 启动应用程序即可。

----结束

## 环境要求

表 4-6 环境要求

项目	说明
CPU	主频2.0GHz以上
内存	8GB以上
硬盘	<ul style="list-style-type: none"><li>• 临时目录空间：20GB以上</li><li>• 安装目录空间：20GB以上</li></ul>

项目	说明
操作系统	<ul style="list-style-type: none"><li>• Windows 10</li><li>• CentOS 7.2</li><li>• 银河麒麟 4/10</li></ul>
数据库	<ul style="list-style-type: none"><li>• MySQL 5.5.4/5.6.33/5.7.20</li><li>• Oracle 11g/12.1.0.2.0/12.2.0.1 • SQL Server 2016</li><li>• PostgreSQL 9.6</li><li>• DM7/8</li></ul>
应用服务器	<ul style="list-style-type: none"><li>• Tomcat 8.5.51</li><li>• AppServer 7</li></ul>
浏览器	<ul style="list-style-type: none"><li>• Chrome 84</li><li>• Firefox 78</li><li>• IE 11</li></ul>

## 4.2.2 安装须知

标准产品 DSP V7.0 配套的组件及相关资源版本如下所示

表 4-7 相关资源

资源名称	版本号	说明
JDK	1.8.0_172	OracleJDK
EOS Platform	8.1	Governor 是用 EOS8 开发的
Tomcat	8.5.51	Governor、Scheduler、DataRelease内置
Mysql 数据库驱动	5.7.20	mysql-connector-java-5.1.32.jar
Oracle 数据库驱动	11g	ojdbc6.jar
Oracle 数据库驱动	12c	ojdbc7.jar
SQL Server 数据库驱动	2016	mssqlserver-4.0.jar
PostgreSQL 数据库驱动	9.6	postgresql-42.2.8.jar
DM 数据库驱动	7	Dm7JdbcDriver17.jar DmDialect-for-hibernate5.0.jar (方言)
	8	DmJdbcDriver18.jar DmDialect-for-hibernate5.3.jar (方言)

#### 免责声明：

本产品安装指南仅为向客户提供关于安装及卸载DSP

V7.0产品的说明而准备。我们不对在本产品安装指南中所援引的任何第三方产品的内容、服务功能、可靠性、可用性、兼容性或其他任何性能做作出任何承诺，本产品的范围亦不包括该等第三方产品。我们在本产品安装指南中所对任何第三方提供的产品的援引，仅为说明示例，不代表我们曾经、目前或将来使用、测试、下载该等第三方提供的产品或向用户提供该等第三方产品的下载。如果用户需要使用该等第三方产品，应当通过合法途径自行购买或下载。如果因网络状况、通讯线路、第三方提供的产品质量或政府管理部门的要求、技术故障等任何原因而导致任何使用人不能正常使用DSP V7.0产品的，我们不因此承担任何法律责任。

如果我们在本产品安装指南中所援引的任何第三方提供的产品存在侵犯他人知识产权、名誉权或其他合法权益的内容，或存在损害国家利益、违背现行法律法规及相关政策、违背公序良俗的情形，我们将对在知悉上述情况后对相应内容采取屏蔽、删除等处理措施。尽管有上述承诺，我们对上述内容的具体处理方式以法律要求的义务为限，我们不因未履行超出法律规定之义务而承担责任。

如果个人、法人及非法人组织（以下简称“投诉人”）认为我们在本产品安装指南中援引的任何第三方提供的产品存在侵犯自身合法权益的内容，应依照法律规定向我们发出通知并提供具有法律效力的证明材料。我们将依照投诉人通知内容以及现行法律规定对涉嫌侵权内容进行处理。我们依法处理相关内容，即视为履行完毕相关法律义务，除此之外不再对涉嫌侵权内容承担任何法律责任。

## 4.2.3 DSP Governor

安装到Tomcat

解压安装

### 安装与配置

**步骤1** 创建“DSPGovernor”目录。

**步骤2** 解压安装包到“DSPGovernor”目录中。

- Linux  
`#tar -xvf Primeton_DSP_7.0_Governor_Linux_64.tar -C DSPGovernor`
- Windows  
用解压工具将“Primeton\_DSP\_7.0\_Governor\_Windows\_64.zip”解压至“DSPGovernor”目录

**步骤3** 以Mysql数据库为例，创建数据库。在“DSPGovernor\scripts\server\mysql目录下，按照数字顺序执行数据库脚本。

## 📖 说明

- Mysql数据库执行初始化脚本请用Navicat Premium工具，且mysql执行存储过程脚本，需要手动在首行添加“delimiter /”。

如果安装的是Mysql5.7版本，需要修改数据库 my.cnf 文件的 sql\_mode 参数，修改如下所示

```
# sql_mode=ONLY_FULL_GROUP_BY, ....#  
ONLY_FULL_GROUP_BY  
sql_mode=STRICT_TRANS_TABLES, ERROR_FOR_DIVISION_BY_ZERO, NO_AUTO_CREATE_USER,  
NO_ENGINE_SUBSTITUTION
```

提示：如果用户开启了mysql的binlog功能，初始化数据库之前，需要设置数据库属性：  
log\_bin\_trust\_function\_creators=TRUE。

- SQL Server数据库执行初始化脚本请用Navicat Premium工具。
- Oracle数据库执行初始化脚本请用PLSQL工具。
- 达梦数据库执行初始化脚本请使用数据库自带客户端工具。如果执行“dm\3\_esb\2\_struct.sql”脚本创建序列报错，需把序列脚本单行执行。
- PostgreSQL数据库执行初始化脚本请使用Navicat Premium工具，执行存储过程脚本，需要手动删除“/”。

**步骤4** 将数据库驱动包复制到“DSPGovernor\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\lib”目录下。

**步骤5** 修改“DSPGovernor\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\src\config\user-config.xml”配置文件。

```
(.....)  
<group name="default">  
//  
<configValue key="Database-Type">MySql</configValue>  
//MySql  
<configValue key="Jdbc-Type"/>  
<configValue  
key="C3p0-DriverClass">com.mysql.jdbc.Driver</configValue> //  
<configValue  
key="C3p0-Url">jdbc: mysql: //127.0.0.1: 3306/dsp</configValue> //URL  
<configValue key="C3p0-UserName">root</configValue> //  
<configValue key="C3p0-Password">123456</configValue> //  
<configValue key="C3p0-PoolSize">50</configValue>  
<configValue key="C3p0-MaxPoolSize">50</configValue>  
<configValue key="C3p0-MinPoolSize">50</configValue>  
(.....)  
<module name="DataRelease">  
<group name="Service">  
<configValue key="Publish-Ip">localhost</configValue> //datareleaseipdatarelease  
<configValue key="Publish-Port">9091</configValue>  
//datareleasetomcat  
</group>  
</module>  
<module name="MetaCube">  
<! -- DSP / -->  
<group name="extractor">  
<configValue key="host">localhost</configValue>  
//IPDSP GovernorIP  
<configValue key="port">9999</configValue>  
//  
</group>  
</module>  
<module name="GatherClient">  
<group name="Service">  
<configValue key="Port">8000</configValue>  
//  
</group>  
</module>  
<module name="HBaseMapping">
```

```
<group name="Service">  
<configValue key="Ip">localhost</configValue>  
//HBaseMappingIPDSP GovernorIP  
<configValue key="Port">8383</configValue>  
//HBaseMapping8383  
<!--<configValue  
key="DefaultNameSpace">http://www.primeton.com/</configValue-->  
</group>  
</module>
```

----结束

## 验证

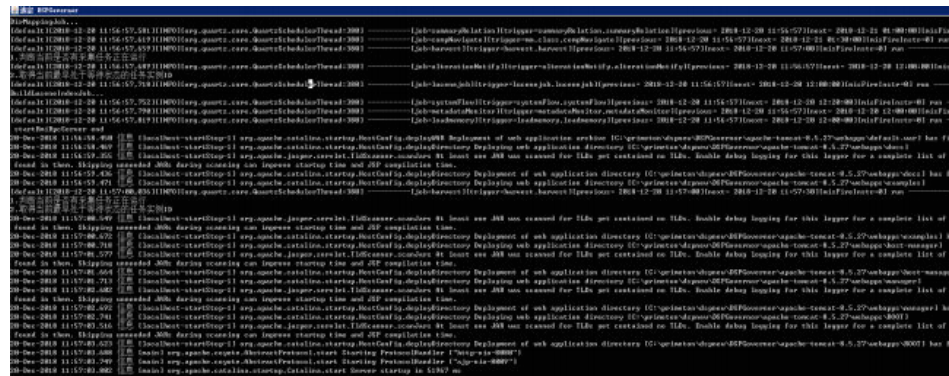
步骤1 在“DSPGovernor”目录下，执行如下命令，启动DSP

Governor，启动时后台会开启三个终端进程，分别是：Governor、HbaseMapping、Extractor。

- Linux  
# chmod -R +x  
# nohup ./startup.sh &
- Windows  
>startup.cmd

启动完成后，Governor如下图所示。

图 4-5 Governor



HbaseMapping如下图所示。

图 4-6 HbaseMapping

```
HbaseMapping
2019-01-03 09:30:56.841 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'configurationPropertiesRebinder' has been autodetected for JMX exposure
2019-01-03 09:30:56.841 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'refreshEndpoint' has been autodetected for JMX exposure
2019-01-03 09:30:56.841 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'environmentManager' has been autodetected for JMX exposure
2019-01-03 09:30:56.841 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Bean with name 'refreshScope' has been autodetected for JMX exposure
2019-01-03 09:30:56.841 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'environmentManager': registering with JMX server as MBean [org.springframework.cloud.context.environment:name=environmentManager,type=EnvironmentManager]
2019-01-03 09:30:56.997 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'refreshScope': registering with JMX server as MBean [org.springframework.cloud.context.scope.refresh:name=refreshScope,type=RefreshScope]
2019-01-03 09:30:57.013 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'configurationPropertiesRebinder': registering with JMX server as MBean [org.springframework.cloud.context.properties:name=configurationPropertiesRebinder,context=79179359,type=ConfigurationPropertiesRebinder]
2019-01-03 09:30:57.013 INFO 17160 --- [main] o.s.j.e.a.AnnotationMBeanExporter : Located managed bean 'refreshEndpoint': registering with JMX server as MBean [org.springframework.cloud.endpoint:name=refreshEndpoint,type=RefreshEndpoint]
2019-01-03 09:30:57.169 INFO 17160 --- [main] o.s.c.s.DefaultLifecycleProcessor : Starting beans in phase 0
2019-01-03 09:30:57.216 INFO 17160 --- [main] o.s.c.s.DefaultLifecycleProcessor : Starting beans in phase 2147483647
2019-01-03 09:30:57.309 INFO 17160 --- [main] org.springframework.scheduling.concurrent.ThreadPoolTaskScheduler : Starting scheduler
2019-01-03 09:30:57.966 INFO 17160 --- [main] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8383 (http)
2019-01-03 09:30:57.981 INFO 17160 --- [main] c.p.e.Application : Started Application in 53.099 seconds (JVM running for 121.332)
```

dgs-extractor采集器如下图所示。

图 4-7 dgs-extractor 采集器

```
C:\WINDOWS\system32\cmd.exe
2019-01-03 09:29:56.036 INFO ClassPathXmlApplicationContext:411 - Refreshing org.springframework.context.support.ClassPathXmlApplicationContext@215be6bb: display name [org.springframework.context.support.ClassPathXmlApplicationContext@215be6bb], startup date [Thu Jan 03 09:29:55 CST 2019], root of context hierarchy
2019-01-03 09:29:56.943 INFO XmlBeanDefinitionReader:323 - Loading XML bean definitions from class path resource [extractor/spring/context-adapter-rmi-server.xml]
2019-01-03 09:30:03.866 INFO ClassPathXmlApplicationContext:426 - Bean factory for application context [org.springframework.context.support.ClassPathXmlApplicationContext@215be6bb]: org.springframework.beans.factory.support.DefaultListableBeanFactory@4e7dc304
2019-01-03 09:30:04.085 INFO DefaultListableBeanFactory:414 - Pre-instantiating singletons in org.springframework.beans.factory.support.DefaultListableBeanFactory@4e7dc304: defining beans [extractorService,registry,rmiDataEngineService]; root of factory hierarchy
2019-01-03 09:30:04.554 INFO RmiRegistryFactoryBean:251 - Looking for RMI registry at port '9999'
2019-01-03 09:30:05.569 INFO RmiRegistryFactoryBean:261 - Could not detect RMI registry - creating new one
2019-01-03 09:30:07.928 INFO RmiServiceExporter:271 - Binding service 'extractorService' to RMI registry: RegistryImpl[unicastServerRef [liveRef: [endpoint:[192.168.253.1:9999](local),objID:[0:0:0, 0]]]]
采集模块服务运行中 ...
| 停止采集模块, 请键入stop |
```

**步骤2** 启动完成后，在浏览器地址栏中输入http://部署DSP Governor主机名或IP：服务器端口号/dsp，（例如：http://192.168.30.173：8080/dsp），按回车键后进入数据服务共享平台登录界面，如下图所示。

图 4-8 登录界面



步骤3 输入用户名和密码后，单击“登录”，进入数据服务共享平台主页。

图 4-9 主页



----结束

## 卸载

步骤1 在“DSPGovernor”目录下，执行如下命令，停止DSP

Governor。停止/关闭时会同时关闭三个终端进程，分别是：Governor、HbaseMapping、Extractor。如果失败请手动逐个停止/关闭。

- Linux  
#./shutdown.sh
- Windows  
>shutdown.cmd

步骤2 停止服务后，直接删除“DSPGovernor”目录来卸载。

----结束



## 4.2.4 DSP Server

解压安装

### 安装与配置

**步骤1** 创建“DSPServer”目录。

**步骤2** 解压安装包到“DSPServer”目录中。

- Linux

```
#tar -xvf Primeton_DSP_7.0_Server_Linux_64.tar -C DSPServer
```

- Windows

用解压工具将“Primeton\_DSP\_7.0\_Server\_Windows\_64.zip”解压至“DSPServer”目录

**步骤3** 将数据库驱动包复制到“DSPServer\dspserver\lib”目录下。

**步骤4** 在“DSPServer\dspserver\config\di-startup-switch.xml”文件，确认DI Server和BFT Server开关。

```
<?xml version="1.0" encoding="UTF-8" ?>
<bigData>
<DI-Server enable="true"/>
<BFT-Server enable="true"/> //true
<BFT-Agent enable="true"/> //true “bft-agent-config.xml” DSP Agent
</bigData>
```

**步骤5** 在“DSPServer\dspserver\config\di-jmx-config.xml”文件。

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration xmlns="http://www.primeton.com/">
<jmxconfig>
```

<pre>&lt;host&gt;127.0.0.1&lt;/host&gt; &lt;rmiPort&gt;6300&lt;/rmiPort&gt; &lt;/jmxconfig&gt; &lt;/configuration&gt;</pre>	<pre>//DSP // JMXDS P</pre>	<pre>ServerIP ServerDSP Agent</pre>
---	-----------------------------	-------------------------------------

**步骤6** 在“DSPServer\dspserver\config\bft-server-config.xml”文件。

```
(.....)
<dbconfig>
<dbtype>Mysql</dbtype>
//
<driver>com.mysql.jdbc.Driver</driver>
//
<connectionurl>jdbc: mysql: //127.0.0.1: 3306/dsp?useSSL=false</connectionurl> //URL
<user>root</user>
//
<password>123456</password>
//
(.....)
<nettyconfig>
<ip>127.0.0.1</ip>
//DSP ServerIP
<port>8091</port>
//BFTnetty0-65535
<reConnectTime>5</reConnectTime>
</nettyconfig>
<jmxconfig>
<host>127.0.0.1</host>
```

```
//DSP ServerIP  
<port>6210</port>  
//BFTjmx0-65535  
</jmxconfig>
```

步骤7 在“DSPServer\dspserver\config\bft-server-jetty.xml”文件。

```
(.....)  
<Set name="port">  
<SystemProperty name="jetty.http.port" default="8170" />  
//BFTjetty, 0-65535  
</Set>  
(.....)
```

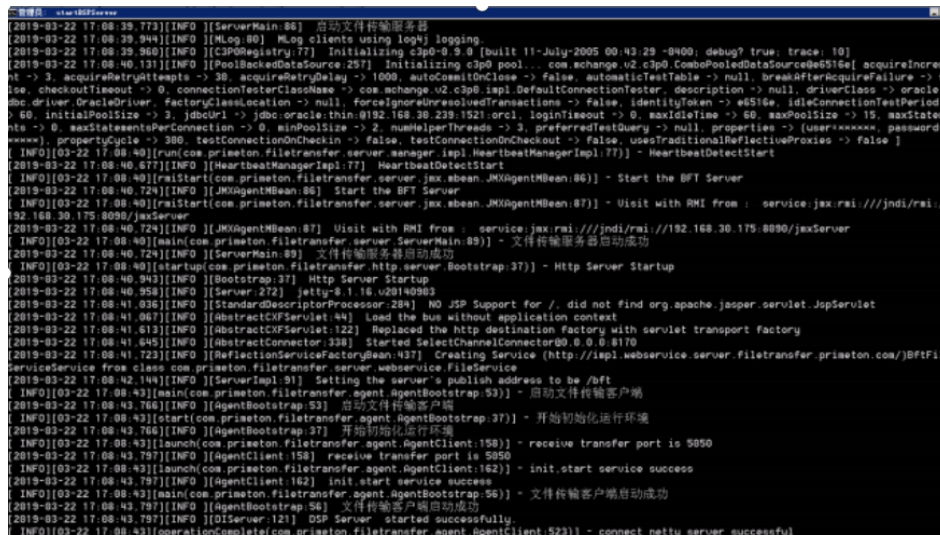
----结束

## 验证

步骤1 在“DSPServer”目录下，执行如下命令，启动DSP Server。

- Linux  
# chmod -R +x  
# nohup ./startDSPServer.sh &
- Windows  
>startDSPServer.cmd  
启动完成后，如下图所示。

图 4-10 启动 DSP Server



步骤2 在“DSPServer\jobagent”目录下，执行如下命令，启动代理服务jobagent。

- Linux  
#./start.sh
- Windows  
>start.bat  
启动完成后，如下图所示。

图 4-11 启动代理服务 jobagent



----结束

## 卸载

**步骤1** 在“DSPServer”目录下，执行如下命令，停止DSP Server。

- Linux  
#./stopDSPServer.sh
- Windows  
>stopDSPServer.cmd

**步骤2** 在“DSPServer\jobagent”目录下，执行如下命令，停止jobagent。

- Linux  
#./stop.sh
- Windows  
>stop.bat

**步骤3** 停止服务后，直接删除“DSPServer”目录来卸载。

----结束

## 4.2.5 DSP Scheduler

安装到Tomcat

### 安装与配置

**步骤1** 创建“DSPScheduler”目录。

**步骤2** 解压安装包到“DSPScheduler”目录中。

- Linux  
#tar -xvf Primeton\_DSP\_7.0\_Scheduler\_Linux\_64.tar -C DSPScheduler
- Windows  
用解压工具将“Primeton\_DSP\_7.0\_Scheduler\_Windows\_64.zip”解压至“DSPScheduler”目录

**步骤3** 将数据库驱动包复制到“/scheduler/apache-tomcat-8.5.51/webapps/scheduler/WEB-INF/lib”目录下。

**步骤4** 在“DSPScheduler/scheduler/dams-config.xml”文件中，修改数据库相关配置。

```
(.....)
<license-file>./license-file>
<appPort>8080</appPort> //DSP SchedulerTomcat, DSP
SchedulerDSP DataReleaseDSP Governor
(.....)
<database-type>MYSQL</database-type> //
```

```
<database-url>jdbc: mysql: //127.0.0.1: 3306/dsp</database-url>  
<database-driver>com.mysql.jdbc.Driver</database-driver>  
<server-name>dsp</server-name>  
<username>root</username>  
<encrypt-type>0</encrypt-type>  
<pass>123456</pass>  
(.....)
```

**步骤5** 如果governor和scheduler在同一台服务器上，则需要修改“DSPScheduler/apache-tomcat-8.5.51/conf/server.xml”文件中的端口号。

<pre>&lt;Server port="8005" shutdown="SHUTDOWN"&gt; (.....) &lt;Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" /&gt;</pre>	<pre>//0~65535 //80804tomcat</pre>
<pre>(.....) &lt;Connector port="8009" protocol="AJP/1.3" redirectPort="8443" /&gt;</pre>	<pre>//0~65535</pre>

**步骤6** 在“DSPScheduler\listener\config\config.properties”文件中，修改事件触发端口号。

<pre>#Created by JInto - www.guh-software.de #Fri May 07 14: 25: 21 CST 2010 baseDir=./filelistener chartSet=GBK fileSuffix=ctl period=2 schedulerServer=http: //localhost: 8080/scheduler/</pre>	<pre>//4tomcat</pre>
---	----------------------

----结束

## 验证

在“DSPScheduler”目录下，执行如下命令，启动DSP Scheduler。

- Linux  
#chmod -R +x  
#nohup ./startup.sh &
- Windows  
>startup.cmd

启动完成后，如下图所示。

图 4-12 启动 1

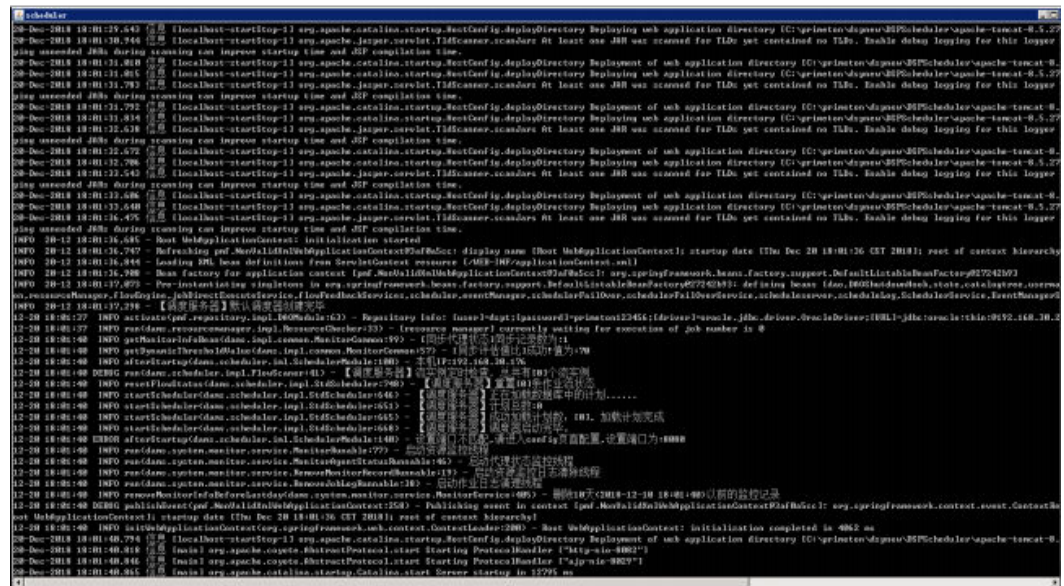
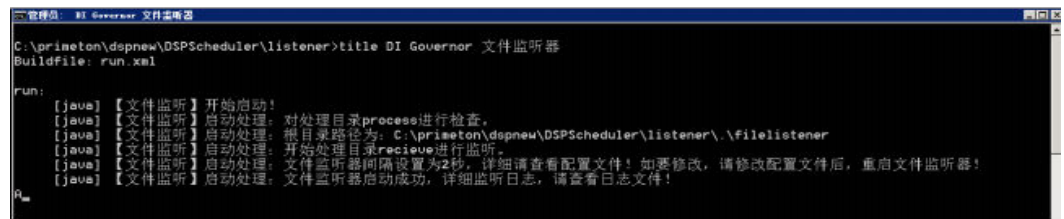


图 4-13 启动 2



## 卸载

**步骤1** 在“DSPScheduler”目录下，执行如下命令，停止DSP Scheduler。

1. Linux  
#./shutdown.sh
2. Windows  
>shutdown.cmd

**步骤2** 停止服务后，直接删除“DSPScheduler”目录来卸载。

----结束

## 4.2.6 DSP DataRelease

安装到Tomcat

### 安装与配置

**步骤1** 创建“DSPDataRelease”目录。

**步骤2** 解压安装包到“DSPDataRelease”目录中。

- Linux  
#tar -xvf Primeton\_DSP\_7.0\_DataRelease\_Linux\_64.tar -C DSPDataRelease

- Windows  
用解压工具将 “Primeton\_DSP\_7.0\_DataRelease\_Windows\_64.zip” 解压至 “DSPDataRelease” 目录

**步骤3** 将数据库驱动包复制到 “DSPDataRelease\apache-tomcat-8.5.51\webapps\datarelease/WEB-INF/lib” 目录下。

**步骤4** 在 “DSPDataRelease/apache-tomcat-8.5.51\webapps\datarelease/WEB-INF/classes/application.yml” 文件，修改数据库相关配置。

```
(.....)
datasource:
url: jdbc:mysql://127.0.0.1:3306/dsp
driverClassName: com.mysql.jdbc.Driver
username: root
password: 123456
(.....)
```

**步骤5** 如果datarelease和governor或scheduler在同一台服务器上，则需要修改 DSPDataRelease/apache-tomcat-8.5.51/conf/server.xml中的端口号。

<Server port="8005" shutdown="SHUTDOWN"> (.....)	//0~65535 //0~65535
<Connector port="8080" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />	
(.....)	
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />	//0~65535

----结束

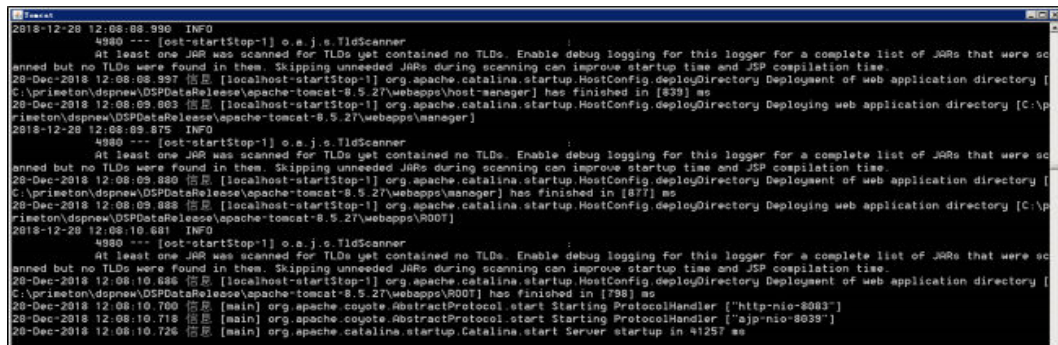
## 验证

在 “DSPDataRelease” 目录下，执行如下命令，启动DSP DataRelease。

- Linux  
#chmod -R +x  
#nohup ./startup.sh &
- Windows  
>startup.cmd

启动完成后，如下图所示。

图 4-14 启动 DSP DataRelease



## 卸载

**步骤1** 在“DSPDataRelease”目录下，执行如下命令，停止DSP DataRelease。

- Linux  
#./shutdown.sh
- Windows  
>shutdown.cmd

**步骤2** 停止服务后，直接删除“DSP DataRelease”目录来卸载。

----结束

## 4.2.7 DSP SSM

解压安装

### 安装与配置

**步骤1** 创建“DSPSSM”目录。

**步骤2** 解压安装包到“DSPSSM”目录中。

- Linux  
#tar -xvf Primeton\_DSP\_7.0\_SSM\_Linux\_64.tar -C DSPSSM
- Windows  
用解压工具将“Primeton\_DSP\_7.0\_SSM\_Windows\_64.zip”解压至“DSPSSM”目录

**步骤3** 将数据库驱动包复制到“DSPSSM\ssm\lib”目录下。

**步骤4** 在“DSPSSM\ssm\instances\default\conf\ssm.user.properties”文件，参照如下表格配置参数。

```
loader.log.file.directory=D: /DSP/datarelease-logs  
(.....)  
dispatcher.statistics.db.driver=com.mysql.jdbc.Driver  
dispatcher.statistics.db.url=jdbc: mysql: //127.0.0.1: 3306/dsp  
dispatcher.statistics.db.user=root  
dispatcher.statistics.db.password=123456  
dispatcher.statistics.db.maxactive=10  
dispatcher.statistics.db.maxidle=5  
dispatcher.statistics.db.maxwait=35  
dispatcher.statistics.exception.enable=true
```

```
dispatcher.statistics.top.enable=true  
dispatcher.statistics.time.enable=true  
dispatcher.statistics.all.enable=false  
mBeanServer.bindPort=6234  
mBeanServer.bindIp=127.0.0.1  
(.....)
```

表 4-8 配置参数

参数	说明
loader.log.file.directory	数据发布组件的日志路径，例如：D: /DSP/ datarelease-logs 数据发布成功后，会生成和datar atarelease-logs目录，配置为datarelease-logs所在目录即可。
dispatcher.statistics.db.driver	数据库驱动。
dispatcher.statistics.db.url	访问数据库的 JDBC URL，例如：jdbc: mysql: // 127.0.0.1 : 3306/dsp
dispatcher.statistics.db.user	数据库用户名。例如：root
dispatcher.statistics.db.password	数据库用户密码。例如：123456

#### 📖 说明

DSP SSM组件必须要与DSP DataRelease安装在同一台服务器中。

#### ----结束

## 验证

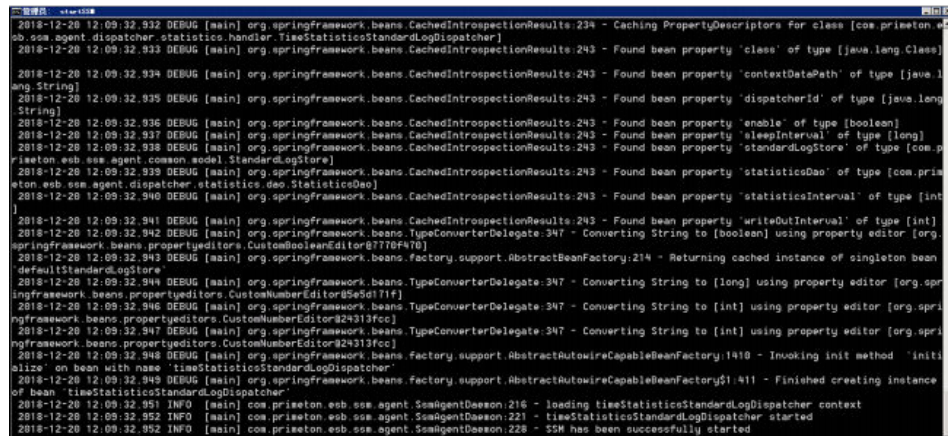
**步骤1** 在“DSPSSM”目录下，执行如下命令，启动DSP SSM。

- Linux  
#chmod -R +x  
#nohup ./startSSM.sh &
- Windows  
>startSSM.cmd

**步骤2** 当命令行出现“SSM has been successfully started”，表示DSP SSM启动成功



图 4-15 DSP SSM 启动



```
2018-12-20 12:09:32.932 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:234 - Caching PropertyDescriptors for class [com.primeton.esb.ssm.agent.dispatcher.statistics.handler.TimeStatisticsStandardLogDispatcher]
2018-12-20 12:09:32.933 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'class' of type [java.lang.Class]
2018-12-20 12:09:32.934 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'contextDataPath' of type [java.lang.String]
2018-12-20 12:09:32.935 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'dispatcherId' of type [java.lang.String]
2018-12-20 12:09:32.936 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'enable' of type [boolean]
2018-12-20 12:09:32.937 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'sleepInterval' of type [long]
2018-12-20 12:09:32.938 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'standardLogStore' of type [com.primeton.esb.ssm.agent.common.model.StandardLogStore]
2018-12-20 12:09:32.939 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'statisticsDao' of type [com.primeton.esb.ssm.agent.dispatcher.statistics.dao.StatisticsDao]
2018-12-20 12:09:32.940 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'statisticsInterval' of type [int]
2018-12-20 12:09:32.941 DEBUG [main] org.springframework.beans.CachedIntrospectionResults:243 - Found bean property 'writeOutInterval' of type [int]
2018-12-20 12:09:32.942 DEBUG [main] org.springframework.beans.TypeConverterDelegate:347 - Converting String to [boolean] using property editor [org.springframework.beans.propertyeditors.CustomBooleanEditor@7776f470]
2018-12-20 12:09:32.943 DEBUG [main] org.springframework.beans.factory.support.AbstractBeanFactory:214 - Returning cached instance of singleton bean [defaultStandardLogStore]
2018-12-20 12:09:32.944 DEBUG [main] org.springframework.beans.TypeConverterDelegate:347 - Converting String to [long] using property editor [org.springframework.beans.propertyeditors.CustomNumberEditor@5e5d71f1]
2018-12-20 12:09:32.945 DEBUG [main] org.springframework.beans.TypeConverterDelegate:347 - Converting String to [int] using property editor [org.springframework.beans.propertyeditors.CustomNumberEditor@24313fcc]
2018-12-20 12:09:32.947 DEBUG [main] org.springframework.beans.TypeConverterDelegate:347 - Converting String to [int] using property editor [org.springframework.beans.propertyeditors.CustomNumberEditor@24313fcc]
2018-12-20 12:09:32.948 DEBUG [main] org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory:1410 - Invoking init method 'initialize' on bean with name 'timeStatisticsStandardLogDispatcher'
2018-12-20 12:09:32.949 DEBUG [main] org.springframework.beans.factory.support.AbstractAutowireCapableBeanFactory:1411 - Finished creating instance of bean 'timeStatisticsStandardLogDispatcher'
2018-12-20 12:09:32.951 INFO [main] com.primeton.esb.ssm.agent.SsmAgentDeamon:216 - Loading timeStatisticsStandardLogDispatcher context
2018-12-20 12:09:32.952 INFO [main] com.primeton.esb.ssm.agent.SsmAgentDeamon:221 - timeStatisticsStandardLogDispatcher started
2018-12-20 12:09:32.952 INFO [main] com.primeton.esb.ssm.agent.SsmAgentDeamon:228 - SSM has been successfully started
```

----结束

## 卸载

步骤1 在“DSPSSM”目录下，执行如下命令，停止DSP SSM。

- Linux  
#./stopSSM.sh
- Windows  
>stopSSM.cmd

步骤2 停止服务后，可以直接删除“DSPSSM”目录来卸载。

----结束

## 4.2.8 DSP Agent

解压安装

### 安装与配置

步骤1 创建“DSPAgent”目录。

步骤2 解压安装包到“DSPAgent”目录中。

- Linux  
#tar -xvf Primeton\_DSP\_7.0\_Agent\_Linux\_64.tar -C DSPAgent
- Windows

用解压工具将“Primeton\_DSP\_7.0\_Agent\_Windows\_64.zip”解压至“DSPAgent”目录

步骤3 将数据库驱动包复制到“DSPAgent\dspagent\lib”目录下。

步骤4 在“DSPAgent\dspagent\config\di-startup-switch.xml”文件，修改DSP Server的配置。

<pre>&lt;bigData&gt; &lt;DI-Server enable="true"/&gt; &lt;BFT-Server enable="true"/&gt; &lt;BFT-Agent enable="true"/&gt; &lt;/bigData&gt;</pre>	<pre>//DSP ServerDSP Server //true</pre>
---	--

**步骤5** 在“DSPAgent\dspagent\config\bft-agent-config.xml”文件，修改如下配置。

<pre>&lt;agent&gt; &lt;name&gt;defaultAgent&lt;/name&gt; &lt;password&gt;00000000&lt;/password&gt; &lt;/agent&gt; (.....) &lt;nettyconfig&gt; &lt;ip&gt;127.0.0.1&lt;/ip&gt;</pre>	<pre>//BFT //BFT8 //DSP</pre>
<pre>Serverbft-server-config.xmlnettyconfigIP &lt;port&gt;8091&lt;/port&gt; &lt;reConnectTime&gt;5&lt;/reConnectTime&gt; &lt;/nettyconfig&gt; &lt;receiveport&gt;7081&lt;/receiveport&gt; // BFTDSP AgentDSP Server0~65535</pre>	

**步骤6** 在“DSPAgent\dspagent\config\bft-agent-jetty.xml”文件，修改jetty的http端口号。

<pre>(.....) &lt;Set name="port"&gt; &lt;SystemProperty name="jetty.http.port" default="8070" /&gt; //BFTjettyDSP AgentDSP Server0~65535 &lt;/Set&gt; (.....)</pre>
<p>说明 如果DSP Agent和DSP Server安装在同一台服务器上，在“DSPAgent\dspagent\config\di-jmx-config.xml”文件，修改jmx管理端口号。</p> <pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;&lt;configuration xmlns="http://www.primeton.com/"&gt;&lt;jmxconfig&gt;&lt;host&gt;127.0.0.1&lt;/host&gt;&lt;rmiPort&gt;6300&lt;/rmiPort&gt;&lt;/jmxconfig&gt;&lt;/configuration&gt;</pre>

----结束

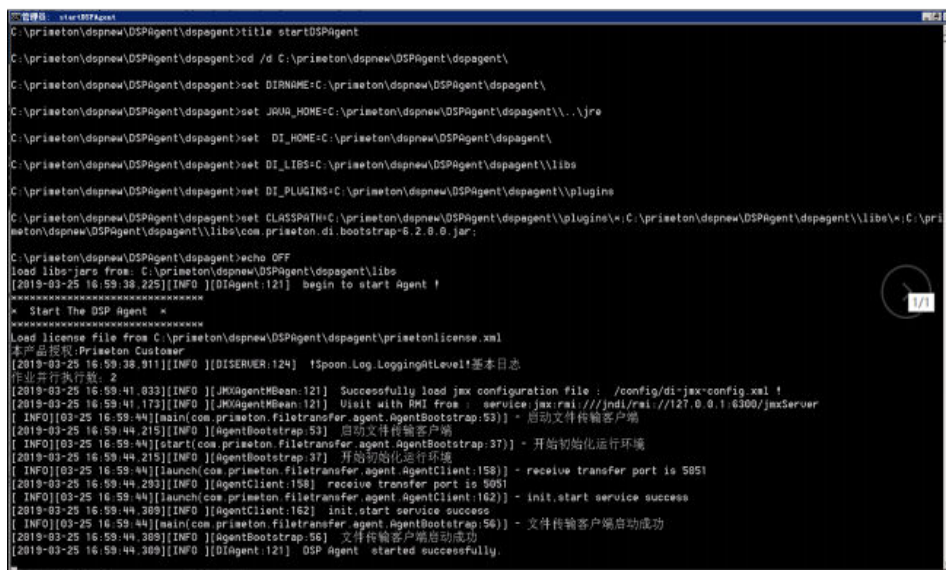
## 验证

**步骤1** 在“DSPAgent”目录下，执行如下命令，启动DSP Agent。

- Linux  
#chmod -R +x  
#nohup ./startDSPAgent.sh &
- Windows  
>startDSPAgent.cmd

**步骤2** 当命令行出现“文件传输客户端启动成功”，表示DSP Agent启动成功。

图 4-16 DSP Agent 启动



```
C:\prisonet\dspnew\DSPAgent\dspagent>title startDSPAgent
C:\prisonet\dspnew\DSPAgent\dspagent>cd /d C:\prisonet\dspnew\DSPAgent\dspagent\
C:\prisonet\dspnew\DSPAgent\dspagent>set DIRNAME=C:\prisonet\dspnew\DSPAgent\dspagent\
C:\prisonet\dspnew\DSPAgent\dspagent>set JAVA_HOME=C:\prisonet\dspnew\DSPAgent\dspagent\.. \jre
C:\prisonet\dspnew\DSPAgent\dspagent>set DI_HOME=C:\prisonet\dspnew\DSPAgent\dspagent\
C:\prisonet\dspnew\DSPAgent\dspagent>set DI_LIBS=C:\prisonet\dspnew\DSPAgent\dspagent\libs
C:\prisonet\dspnew\DSPAgent\dspagent>set DI_PLUGINS=C:\prisonet\dspnew\DSPAgent\dspagent\plugins
C:\prisonet\dspnew\DSPAgent\dspagent>set CLASSPATH=C:\prisonet\dspnew\DSPAgent\dspagent\plugins\*.C:\prisonet\dspnew\DSPAgent\dspagent\libs\*.C:\prisonet\dspnew\DSPAgent\dspagent\libs\com.priseton.di.bootstrap-6.2.8.0.jar;
C:\prisonet\dspnew\DSPAgent\dspagent>echo OFF
load libor.jar from C:\prisonet\dspnew\DSPAgent\dspagent\libs
[2019-03-25 16:59:38.225][INFO][DIAGENT.121] begin to start Agent !
*****
* Start The DSP Agent *
*****
Load license file from C:\prisonet\dspnew\DSPAgent\dspagent\prisonetlicense.xml
本产品授权:Prisonet Customer
[2019-03-25 16:59:38.911][INFO][DISERVER.124] Ispoon.Log.Logging@Level:基本日志
作业并行执行数: 2
[2019-03-25 16:59:41.833][INFO][JMXAgentMBean.121] Successfully load jmx configuration file : /config/di-jmx-config.xml !
[2019-03-25 16:59:41.173][INFO][JMXAgentMBean.121] Visit with RMI from : service:jmx:rmi:///jndi/rmi://127.0.0.1:6300/jmxServer
[2019-03-25 16:59:44.215][INFO][main[com.priseton.filetransfer.agent.AgentBootstrap:53]] - 启动文件传输客户端
[2019-03-25 16:59:44.389][INFO][AgentBootstrap:53] 启动文件传输客户端
[2019-03-25 16:59:44.389][INFO][AgentBootstrap:37] - 开始初始化运行环境
[2019-03-25 16:59:44.215][INFO][AgentBootstrap:37] 开始初始化运行环境
[2019-03-25 16:59:44.389][INFO][main[com.priseton.filetransfer.agent.AgentClient:158]] - receive transfer port is 5851
[2019-03-25 16:59:44.283][INFO][AgentClient:158] receive transfer port is 5051
[2019-03-25 16:59:44.389][INFO][main[com.priseton.filetransfer.agent.AgentClient:162]] - init start service success
[2019-03-25 16:59:44.389][INFO][AgentClient:162] init start service success
[2019-03-25 16:59:44.389][INFO][main[com.priseton.filetransfer.agent.AgentBootstrap:56]] - 文件传输客户端启动成功
[2019-03-25 16:59:44.389][INFO][AgentBootstrap:56] 文件传输客户端启动成功
[2019-03-25 16:59:44.389][INFO][DIAGENT.121] DSP Agent started successfully.
```

---结束

## 卸载

**步骤1** 在“DSPAgent”目录下，执行如下命令，停止DSP Agent。

- Linux  
#./stopDSPAgent.sh
- Windows  
>stopDSPAgent.cmd

**步骤2** 停止服务后，可以直接删除“DSPAgent”目录来卸载。

---结束

## 4.2.9 DSP GatherClient

解压安装

## 安装与配置

**步骤1** 创建“DSPGatherClient”目录。

**步骤2** 解压安装包到“DSPGatherClient”目录中。

- Linux  
#tar -xvf Primeton\_DSP\_7.0\_GatherClient\_Linux\_64.tar -C DSPGatherClient
- Windows  
用解压工具将“Primeton\_DSP\_7.0\_GatherClient\_Windows\_64.zip”解压至“DSPGatherClient”目录

**步骤3** 将数据库驱动包复制到“DSPGatherClient\lib”目录下。

**步骤4** 打开“DSPGatherClient\startup.conf”文件，按照如下说明配置。

DEVICEID=mydevice SERVERIP=127.0.0.1 SERVERPORT=8000	
参数	说明
DEVICEID	设备ID，和服务引擎管理中注册的物理设备“名称”一致，也就是3.4.1.1 物理设备管理中添加的物理设备的“名称”。注册设备时名称不能为中文，否则监测不到。
SERVERIP	采集客户端连接采集服务器的IP地址，集成在DSP Governor中。
SERVERPORT	采集客户端连接采集服务器的端口号。需要与“DSPGovernor\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\_srv\config\user-config.xml”中的采集服务器的端口号8000保持一致。

**步骤5** 在需要进行引擎监控的服务启动脚本的“-classpath”参数前添加jvm参数Dengineld和DengineType。启动脚本中参数说明如下所示。

**表 4-9** 参数说明

参数	说明
Dengineld	和“服务引擎管理”中注册的各个服务引擎的名称一致，也就是3.4.1.2 作业服务器、3.4.1.3 代理服务器、3.4.1.4 调度服务器、3.4.1.5 文件传输服务器、3.4.1.6 文件传输节点中的“引擎名称”。注册服务引擎名称不能为中文，否则监测不到。
DengineType	服务引擎类型，与业务字典中DSP_DEVICE_RESOURCE_TYPE一致，各个服务器引擎类型如下表格第四列所示。

----结束

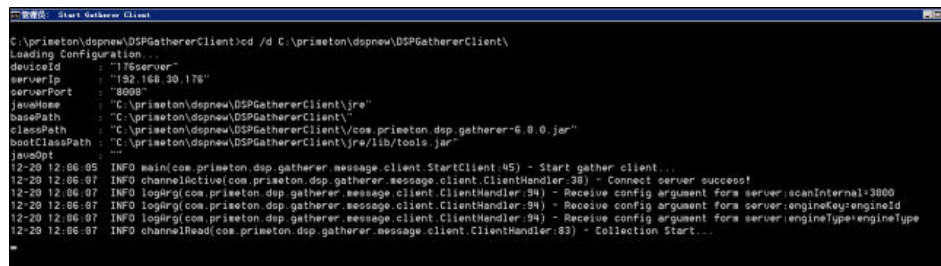
## 验证

**步骤1** 在“DSPGatherClient”目录下，执行如下命令，启动DSP GatherClient。

- Linux  
#chmod -R +x  
#nohup ./startClient.sh &
- Windows  
>startClient.bat

**步骤2** 启动完成后，如下图所示。

图 4-17 启动 DSP GatherClient



```
Start Gather Client
E:\primeton\dspnew\dspgatherclient>cd /d C:\primeton\dspnew\dspgatherclient\
Loading Configuration...
deviceId      : "176server"
serverIp      : "192.168.30.176"
serverPort    : "8088"
javaHome      : "C:\primeton\dspnew\dspgatherclient\jre"
basePath      : "C:\primeton\dspnew\dspgatherclient\"
classpath     : "C:\primeton\dspnew\dspgatherclient\cos.primeton.dsp.gatherer-6.0.0.jar"
bootClassPath : "C:\primeton\dspnew\dspgatherclient\jre\lib\tools.jar"
javaOpt       : ""
12-29 12:06:05 INFO main(cos.primeton.dsp.gatherer.message.client.StartClient:45) - Start gather client...
12-29 12:06:07 INFO channelActive(cos.primeton.dsp.gatherer.message.client.ClientHandler:28) - Connect server success!
12-29 12:06:07 INFO logArg(cos.primeton.dsp.gatherer.message.client.ClientHandler:94) - Receive config argument from server:scanInterval:3000
12-29 12:06:07 INFO logArg(cos.primeton.dsp.gatherer.message.client.ClientHandler:94) - Receive config argument from server:engineKey:engineId
12-29 12:06:07 INFO logArg(cos.primeton.dsp.gatherer.message.client.ClientHandler:94) - Receive config argument from server:engineType:engineType
12-29 12:06:07 INFO channelRead(cos.primeton.dsp.gatherer.message.client.ClientHandler:83) - Collection Start...
```

----结束

## 卸载

**步骤1** 在“DSPGatherClient”目录下，执行如下命令，停止DSP GatherClient。

- Linux  
#./stop.sh
- Windows  
>stop.bat

**步骤2** 停止服务后，直接删除“DSP GatherClient”目录来卸载。

----结束

## 4.2.10 DSP Studio

### 安装与配置

**步骤1** 创建“DSPStudio”目录。

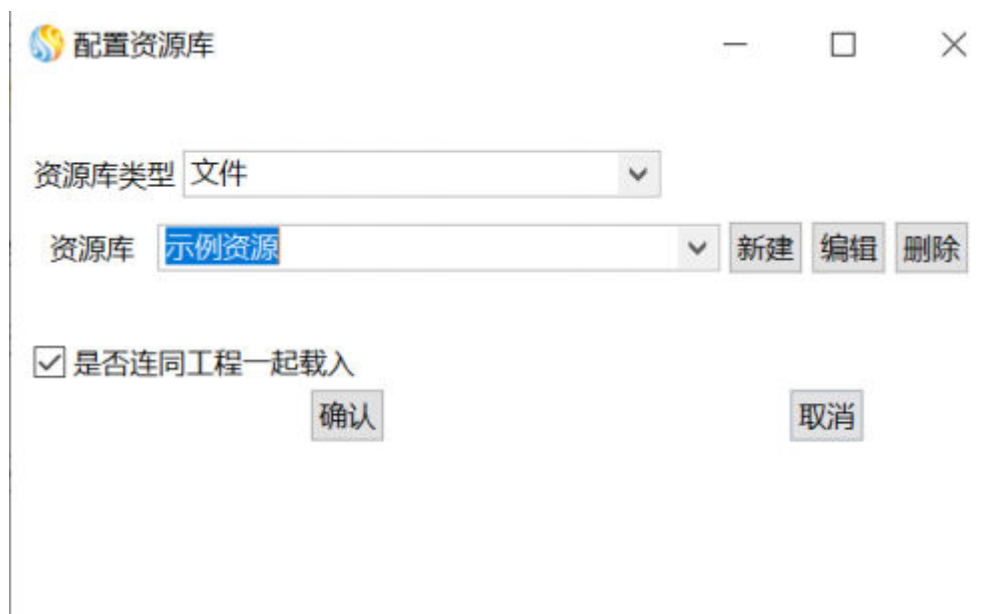
**步骤2** 用解压工具将“Primeton\_DSP\_7.0\_Studio\_Windows\_64.zip”安装包解压至“DSPStudio”目录中。

----结束

## 验证

在“DSPStudio”目录中，双击“startDSPStudio.cmd”，启动DSP Studio，如下图所示。

图 4-18 启动 DSP Studio



## 卸载

关闭DSP Studio后，直接删除“DSPStudio”目录来卸载。

# 5 附录

## 背景信息

表 5-1 背景信息

术语	产品
DSP	数据共享交换管理
taglib	数据标签库管理

## 常见问题

### 数据标签库管理常见问题

- 问题1：修改应用默认端口
  - a. 停止应用
  - b. 修改taglibr/conf/server.xml中如下内容

图 5-1 修改应用默认端口

```
<?xml version="1.0" encoding="UTF-8"?>
<Server port="18089" shutdown="SHUT_DOWN">
  <Listener className="org.apache.catalina.startup.VersionLoggerListener" />
  <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
  <Listener className="org.apache.catalina.core.JreMemoryLeakPreventionListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.core.ThreadLocalLeakPreventionListener" />
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <Service name="Catalina">
    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="1800"
      maxThreads="200"
      KeepAliveTimeout="1800"
      redirectPort="8443"
      relaxedPathChars="|{}|,"
      relaxedQueryChars="|{}|," />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.LockOutRealm">
        <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
          resourceName="UserDatabase"/>
      </Realm>
      <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true">
        <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
          prefix="localhost_access_log" suffix=".txt"
          pattern="common" resolveHosts="false" />
        <!-- <Context path="/tagcalc" docBase="..hive_app/mrs_tagcalc" reloadable="true"/> -->
        <Context path="/tagcalc" docBase="..gauss_app/tagcalc" reloadable="true"/>
      </Host>
    </Engine>
  </Service>
</Server>
```

c. 启动应用

● 问题2：软件安装后无法启动？

查看应用安装路径下的bdatag/logs/catalins.out日志文件，具体分析。命令如下：

```
cat bdatag/logs/catalins.out
```

一般应用启动失败，较常见的情况是端口冲突，可通过日志具体分析。查看到具体时间那个端口冲突之后，则需要对端口进行修改：

```
vi bdatag/engine/tomcat/conf/server.xml
```

查找文件内容中的冲突端口，然后修改成新的端口。例如日志中显示8005端口冲突，则在文件内容中找到8005，修改成其他端口如8006，然后保存重启应用。

● 问题3：应用启动后无法访问

a. 检查应用与数据库是否正常启动

b. 查看服务器的防火墙是否关闭

linux 6及以下操作系统使用命令

```
service iptables status
```

linux 7及以上操作系统使用命令

```
systemctl status firewalld
```

如果防火墙是启动状态，则确认是否可以关闭防火墙，如果可以则关闭防火墙

linux 6及以下操作系统使用命令

```
service iptables stop
```

linux 7及以上操作系统使用命令

```
systemctl stop firewalld
```

如果不能关闭防火墙，则另外处理网络问题。

c. 检查浏览器端到应用服务器之间网络是否连通，使用ping跟telnet命令

ping 服务器IP



例：ping 192.168.10.31

telnet 服务器ip 应用端口

例：telnet 192.168.10.31 18089

ping返回结果失败，telnet也失败，网络不通。

ping返回结果失败，telnet成功，则访问地址http://IP: 端口/health.html，如果可以打开页面，则表示应用存在问题，先检查taglibr/bda-config/bda.conf中的配置是否正确，如果配置正确则检查应用服务器跟数据库服务器之间的网络是否连通，同时查看 taglibr/logs/catalins.out日志报错信息。

ping返回结果正常，但是telnet返回结果失败表示网络通，端口不通。此时去查看应用的使用端口，如果应用使用端口无误，则网络不通。

- **问题4：应用运行一段时间候，出现部分功能不正常或者所有页面均为空白页**  
磁盘空间可能被其他程序占用，此时可查看服务器磁盘剩余空间

```
df -h
```

如果有使用率达到百分百的，则需要删除一些无用的文件，以留出空间后再重启应用。

### 数据共享交换管理常见问题

- **问题5：端口列表及说明**

图 5-2 DSP Governor 用到的默认端口号

端口号	配置文件	说明
6200	{DSP Governor目录}\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\application.properties	EOS管理端
8080	{DSP Governor目录}\governor\apache-tomcat-8.5.51\conf\server.xml	DSP Governor访
9091	{DSP Governor目录}\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\_srv\config\user-config.xml	datarelea 端口号，需 ase端口保
9999	{DSP Governor目录}\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\_srv\config\user-config.xml	采集器服务
8000	{DSP Governor目录}\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\_srv\config\user-config.xml	采集服务器
8383	{DSP Governor目录}\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\_srv\config\user-config.xml	HBaseMapp

图 5-3 DSP Server 用到的默认端口号 1

端口号	配置文件	说明
6300	{DSP Server目录}\dspserver\config\di-jmx-config.xml	JMX管理端口号，DSP Server、 DSP Agent在同一台服务器中需要修改。
8091	{DSP Server目录}\dspserver\config\bft-server-config.xml	BFT服务引擎netty端口号。
6210	{DSP Server目录}\dspserver\config\bft-server-config.xml	BFT服务引擎的jmx端口号。
8170	{DSP Server目录}\dspserver\config\bft-server-jetty.xml	BFT服务引擎的jetty端口号。

图 5-4 DSP Server 用到的默认端口号 2

7081	{DSP Server目录}\dspserver\config\bft-agent-config.xml	BFT节点引擎端口号，DSP Server与DSP Agent在同一台服务器器时需要修改。
8091	{DSP Server目录}\dspserver\config\bft-agent-config.xml	BFT服务引擎netty端口号。
8070	{DSP Server目录}\dspserver\config\bft-agent-jetty.xml	BFT节点引擎的jetty端口号，DSP Agent与DSP Server在同一台服务器器时需要修改。

图 5-5 DSP Agent 用到的默认端口号

端口号	配置文件	说明
6300	{DSP Agent目录}\dspagent\config\di-jmx-config.xml	JMX管理端口号，DSP Agent与DSP Server在同一台服务器中需要修改。
8091	{DSP Agent目录}\dspagent\config\bft-agent-config.xml	BFT服务引擎 netty端口号。需要和“{DSP Server目录}\dspserver\config\bft-server-config.xml”中的8091端口号一致。
8070	{DSP Agent目录}\dspagent\config\bft-agent-jetty.xml	BFT节点引擎的 jetty端口号，DSP Agent与DSP Server在同一台服务器器时需要修改。
7081	{DSP Agent目录}\dspagent\config\bft-agent-config.xml	BFT节点引擎端口号，DSP Agent与DSP Server在同一台服务器器时需要修改。

图 5-6 DSP Scheduler 用到的默认端口号

端口号	配置文件	说明
8080	{DSP Scheduler目录}\scheduler\apache-tomcat-8.5.51\conf\server.xml	Scheduler的Tomcat端口号，DSP Scheduler、DSP DataRelease、DSP Governor在同一台服务器需要修改此端口号。

图 5-7 DSP DataRelease 用到的默认端口号

端口号	配置文件	说明
8080	{DSP DataRelease目录}\apache-tomcat-8.5.51\conf\server.xml	DataRelease的Tomcat端口号，DSP DataRelease、DSP Scheduler与DSP Governor在同一台服务器需要修改此端口号。

图 5-8 DSP GatherClient 用到的默认端口号

端口号	配置文件	说明
8000	{DSP GatherClient目录}\startup.conf	采集客户端连接采集服务器的端口号。需要与“{DSP Governor目录}\governor\apache-tomcat-8.5.51\webapps\dsp\WEB-INF\classes\META-INF\_srv\config\user-c”中的采集服务器的端口号8000保持一致。

# 6 修订记录

发布日期	修订记录
2024-04-25	规范词、敏感词专项处理，章节优化
2023-12-06	第一次正式发布。