

广州市电子政务数据中心 接入开发实施规范（试行）

V3.0.20110121

广州市科技和信息化局

2011 年 1 月

目录

1 前言	1
2 适用范围	2
3 交换体系	2
3.1 基本原则	2
3.2 运行架构	2
3.3 管理边界	4
3.4 交换方式	4
4 消息交换	6
4.1 安装 SDK	6
4.2 发送消息	9
4.3 接收消息	16
4.4 交换审计	22
4.5 异常处理	25
4.6 交换接口	28
5 数据库交换	35
5.1 提供共享数据	35
5.2 获取共享数据	40
5.3 交换接口	43
6 WEB 服务交换	53
6.1 提供 WEB 服务	53
6.2 使用 WEB 服务	59
6.3 交换接口	61
7 实施要求	63
7.1 配置要求	63
7.2 网络要求	64

7.3	安全措施.....	65
7.4	安装部署.....	65
7.5	运行管理.....	67
8	附录	67
8.1	术语定义.....	67
8.2	规范性引用	68
8.3	交换节点编码	70

版本历史

版本号	修改日期	作者	修改描述
V1.0	2007-3-28	PJ10 项目组	征求意见稿
V1.1	2007-4-4	PJ10 项目组	根据专家评审意见修订，增加异常处理
V1.2	2007-4-8	PJ10 项目组	改用 timestamp 数据类型提高时间字段精度
	2007-4-10	PJ10 项目组	改进数据提供流程图、数据获取流程图
V1.3	2007-4-25	PJ10 项目组	根据劳社的反馈建议，完善数据提供接口，增加事后填写批次数据量的功能设计； “问题解答”中增加在开发调试阶段，交换桥接对带 TEST 前缀的数据接口的使用说明。
V1.4	2007-4-29	PJ10 项目组	增加对使用 Oracle 数据链路的说明，在调用数据接口后必须提交事务。
V1.5	2007-6-4	PJ10 项目组	修改数据提供的流程，把“获取出错信息”和“重置数据提供”整合到“开始数据提供”。
	2007-6-25	PJ10 项目组	修改前置机配置要求，增加面向电子监察专题的 FAQ 说明
V1.6	2007-10-20	PJ10 项目组	完善政务服务中心专题说明
V1.7	2007-11-16	PJ10 项目组	修订四处错别字
V1.8	2008-8-18	PJ10 项目组	完善文件交换方式技术要求、交换接口 API、增加对 JSR105 (XML 数字签名)、JSR208 (Java 业务集成) 标准的引用说明
	2008-08-25	PJ10 项目组	根据数据部 0822 意见修改； 增加交换节点列表。
V1.9	2008-11-5	PJ10 项目组	按数据库交换、文件交换、Web 服务交换来组织文档内容。 补充 Web 服务交换方式的技术要求和实施要求。
	2008-11-18	PJ10 项目组	调整章节顺序，从部门的角色出发，先说明技术要求，再补充数据中心提供什么接口。
V2.0	2010-2-26	PJ10 项目组	数据库交换部分增加 SQL SERVER 数据库的示例代码
V3.0	2011-1-18	市数据中心运维组	增加消息交换规范，将文件交换合并到消息交换。

1 前言

根据国家政务信息资源交换体系的要求，市电子政务数据中心信息共享交换平台为各单位提供统一的交换服务，业务应用系统通过交换桥接系统实现与交换平台的对接。为使政务信息资源共享有序开展，实现政务部门之间的数据共享交换，广州市科技和信息化局组织编写了《广州市电子政务数据中心接入开发实施规范》，以指导各政务部门与数据中心交换平台对接的开发和实施工作。

本规范共分为以下四个部分：

第一部分：适用范围。说明本规范的适用范围。

第二部分：交换体系。概括描述由数据中心交换节点和各政务部门交换节点组成的交换体系结构。

第三部分：交换接口。详细定义了交换平台提供的交换服务接口（消息交换、数据库交换、Web 服务交换）及其数据结构。

第四部分：实施要求。说明了交换桥接的系统配置、安装部署、运行管理等方面在实施过程中需要注意的事项。政务部门应按照“[实施要求](#)”进行部署实施。

2 适用范围

基于市区两级数据共享框架，各政务部门通过交换桥接系统的方式接入数据中心交换平台，以实现政务部门之间的数据共享。本规范规定了各政务部门建设交换桥接系统必须遵循的技术和实施要求，适用于面向信息共享的交换和面向业务协同的交换。各政务部门应参照本规范的规定和要求进行交换桥接系统的开发和实施工作。

3 交换体系

3.1 基本原则

条块结合，建立市区两级政务数据共享框架：由市数据中心和各区（县级市）数据中心组成市区两级政务数据共享框架；市级部门的数据共享交换由市数据中心提供支撑，区（县级市）一级部门的数据共享交换由区（县级市）数据中心提供支撑；市级部门的共享数据通过市数据中心返还到各区（县级市）数据中心；政务部门内部市区两级的应用系统的数据交换由部门自行解决，也可以考虑通过市区两级政务数据共享框架支撑。

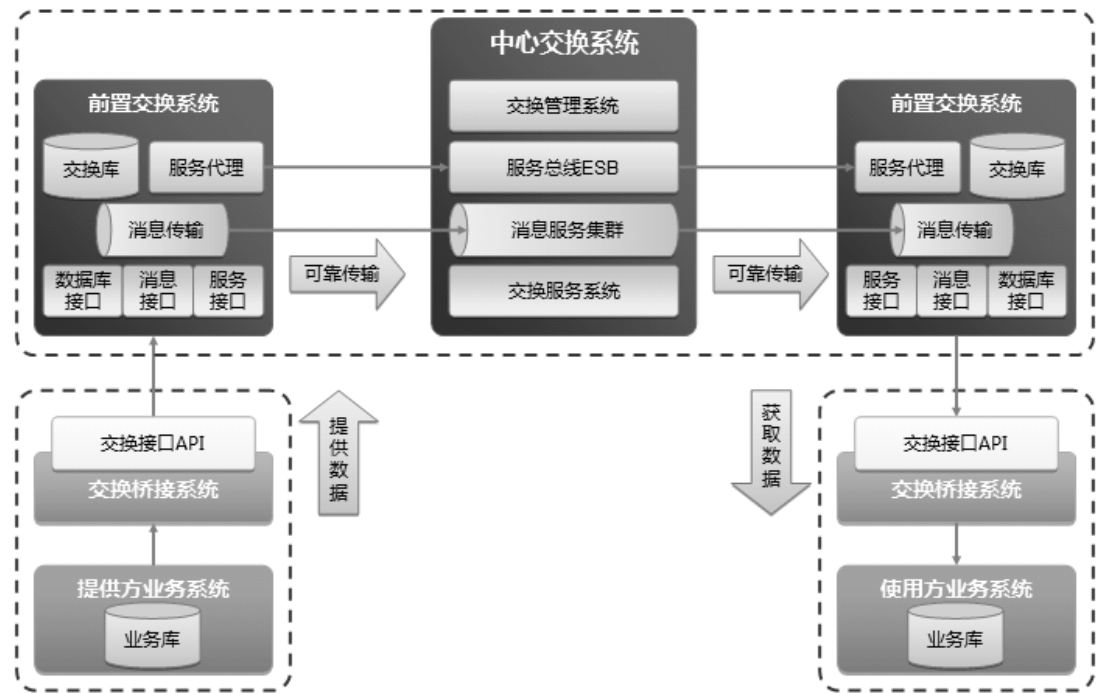
总体规划，坚持四个统一：一是统一交换平台，基于市电子政务数据中心交换平台实现各单位之间的数据交换；二是统一接口规范，各单位面向市数据中心交换平台的一套接口规范进行接口开发和实施；三是统一数据标准，业务数据标准由各业务单位制定，共享数据标准由市政务信息资源目录管理机构制定，遵循统一的数据标准进行共享交换；四是统一前置机，充分整合已有资源，各单位通过一台前置机实现接入。

3.2 运行架构

广州市电子政务数据中心信息共享交换平台（以下简称“交换平台”）由中心交换节点（中心交换系统）和端交换节点（前置交换系统）组成，依托统一的电子政务外网，通过采用统一的交换协议，实现跨地区、跨部门应用系统之

间的数据交换。

市电子政务数据中心作为中心交换节点，各前置交换系统作为端交换节点。政务部门业务系统通过交换桥接系统实现与前置交换系统互联，接入交换平台。参照国家政务信息资源交换体系标准，市电子政务数据中心交换平台的技术架构如下图所示：



图表 3-1 交换平台技术架构

业务库是由各政务部门产生和管理的政务信息资源库。交换库是政务部门提供本部门交换数据、获取其他部门交换数据的临时存储库。

中心交换节点提供交换数据的可靠传输服务，实现交换节点之间的数据路由和传送功能。端交换节点提供交换数据的临时存储、数据写入和读取接口，业务系统与端交换节点之间通过交换桥接系统进行交换数据的提供和获取，并实现业务数据和交换数据之间的转换。

端交换节点是交换平台中数据交换的起点或终点。数据交换过程可分为数据提供、数据采集、数据分发和数据获取四个阶段：

- （1）数据提供：提供方业务系统通过交换桥接系统提供数据到提供方前置交换系统；
- （2）数据采集：数据从提供方的前置交换系统传输到中心交换系统；
- （3）数据分发：数据从中心交换系统传输到使用方的前置交换系统；

(4) 数据获取：使用方业务系统通过交换桥接系统从使用方的前置交换系统获取数据到使用方的业务系统。

其中，数据提供和数据获取流程由政务部门的交换桥接系统负责。

对于面向信息共享的交换，采取集中式的交换模式。各政务部门通过交换桥接系统实现对前置交换系统的对接，向数据中心提供数据，并从数据中心获取所需数据。市数据中心负责从各政务部门增量采集数据，并按需分发数据到各政务部门的前置机交换系统。政务部门只需面向市数据中心交换平台开发实施一套数据交换接口。

对于面向业务协同的交换，采取集中与分布相结合的交换模式。交换双方分别是业务服务提供者（Service Provider）和业务服务使用者（Service Consumer），信息交换表现为交换双方业务服务的交互调用。市数据中心交换平台作为服务中介，提供统一的 ESB 服务以及服务代理。在遵循 SOA 体系架构和 Web Service 技术规范的前提下，交换双方可以使用交换平台通用的交换服务接口，也可以自行协商确定业务服务接口，按要求注册到 UDDI 目录。

3.3 管理边界

以前置机为交换边界，数据提供和获取过程是交换桥接系统的职责，数据传输过程是交换平台的职责。数据中心和政务部门的职责分工如下：

（一）数据中心

- 负责中心交换系统和前置交换系统建设、运维和管理。
- 负责交换节点网络运维和管理。
- 负责各前置交换系统之间数据安全、可靠和稳定交换。

（二）政务部门

- 负责交换桥接系统建设、运维和管理。
- 负责部门接入网络运维和管理。
- 负责通过交换桥接系统实现部门业务系统与前置交换系统之间数据安全、可靠和稳定交换。

3.4 交换方式

交换平台支持的交换方式包括：消息交换方式、数据库交换方式和服务交换方式。三种交换方式适用于不同的交换场景，各有优缺点和约束条件。各政务部门应根据业务需求和应用特点选择合适的交换方式。

（一）消息交换方式

消息交换方式	
数据提供	提供方所通过交换接口 API 提供的数据是轻量级消息。
数据获取	使用方通过交换接口 API 获取的数据是轻量级消息。
适用场景	<ul style="list-style-type: none"> ● 轻量级的数据交换。 ● 交换内容格式灵活多变，如需要交换一个正文加多个附件的数据包。 ● 对实时性要求较高。 ● 要求进行交换审计。
约束条件	<ul style="list-style-type: none"> ● 对交换双方都有较高的技术要求。 ● 不适宜大批量的数据交换。 ● 交换内容一般不需要交换平台进行解析处理，由消息发送方和接收方进行打包和解析处理。 ● 每一个交换任务必须有明确的接收方。

（二）数据库交换方式

数据库交换方式	
数据提供	提供方所提供的数据存放在业务数据库中，通过交换接口 API 向交换平台提供数据
数据获取	使用方通过交换桥接从前置机交换库获取到数据，数据的表示形式是数据表（对应.Net 的 DataTable，Java 的 ResultSet）
适用场景	<ul style="list-style-type: none"> ● 交换结构化数据（存放在数据库中的数据） ● 定期批量交换，实时性要求不高的交换任务

约束条件	<ul style="list-style-type: none"> ● 适用于交换能存放到数据库的结构化数据 ● 不适宜对所有数据内容进行完整性校验
------	---

（三）Web 服务交换方式

Web 服务交换方式	
数据提供	提供方把提供数据共享 Web 服务注册到数据中心 ESB 系统
数据获取	使用方在数据中心查询到提供方发布的服务，直接调用提供方提供的数据共享 Web 服务，或调用数据中心的代理来使用提供方的服务。
适用场景	<ul style="list-style-type: none"> ● 实时性要求高，但数据量不大的交换。 ● 使用方不需要所有数据，只需要按需求获取的应答式交换。
约束条件	<ul style="list-style-type: none"> ● 不适宜进行大数据量的交换。 ● 对使用方技术要求较高，必须具备 Web 服务及相关的开发技能。 ● 不支持“增量”数据交换。 ● 要求提供方提供的 Web 服务必须稳定，长期保持可用。 ● 对提供方节点的网络要求高，必须长时间保持稳定联通。

4 消息交换

4.1 安装 SDK

4.1.1 获取 SDK

政务部门可通过<http://dc.gz.gov.cn/services>或者联系数据中心管理员获取交换平台SDK。交换平台SDK主要由交换接口API类库以及配置文件、帮助文档、开发示例等组成，交换平台SDK目录结构如下所示：

序	目录路径	文件说明
1	\conf	配置文件，包括用户凭证和运行配置。

		用户凭证文件: <code>credentials.properties</code> 运行配置文件: <code>dc-client.properties</code>
2	<code>\data</code>	数据目录, 临时存储消息和日志。
2.1	<code>\data\msg</code>	消息文件目录
2.1.1	<code>\data\msg\pending</code>	挂起的消息 (应发未发)
2.1.2	<code>\data\msg\invalid</code>	非法的消息 (永不发送)
2.2	<code>\data\log</code>	交换审计日志目录
2.2.1	<code>\data\log\pending</code>	未同步的交换审计日志
3	<code>\doc</code>	文档目录, 包括接入开发实施规范以及 API 文档。
3.1	<code>\doc\java</code>	Java API 文档
3.2	<code>\doc\dotnet</code>	Dotnet API 文档
4	<code>\example</code>	示例程序, 包括 Java 和 Dotnet 版本示例。
4.1	<code>\example\java</code>	Java 示例程序
4.2	<code>\example\dotnet</code>	Dotnet 示例程序
5	<code>\lib</code>	交换接口 API 类库, 供交换桥接系统调用。
5.1	<code>\lib\java</code>	Java API 类库: <code>dc-client.jar</code>
5.2	<code>\lib\dotnet</code>	Dotnet API 类库: <code>dc-client.dll</code>

4.1.2 安装部署

交换平台 SDK 安装部署过程非常简单, 只需三步:

- (1) 把 SDK 解包到交换桥接系统所在机器的任一本地目录路径 {DIR}, 如: `C:\DatacenterClient`。
- (2) 增加系统环境变量 `DC_CLIENT`, 设置为 {DIR}。
- (3) 把 {DIR}\lib 目录下对应的 API 类库 (`dc-client.jar` 或 `dc-client.dll`) 引入到交换桥接系统可访问到的类库目录即可。

4.1.3 开发引用

在交换桥接系统程序开发中引用交换接口 API, 只需两行:

- (1) Java 版本

```
import gov.datacenter.messages;  
import gov.datacenter.sdk;
```

(2) Dotnet 版本

```
using Gov.Datacenter.Messages;  
using Gov.Datacenter.Sdk;
```

4.1.4 运行配置

交换接口 API 运行配置主要由两个文件决定：dc-client.properties 和 credentials.properties。

(1) 运行配置文件 “dc-client.properties”

```
# 交换平台节点URI  
ServerNode.URI=tcp://192.168.0.92:61616  
  
# 交换平台节点名称  
ServerNode.Name=GONGSHANG  
  
# 客户端应用唯一标识  
Client.AppID=GONGSHANG-APP  
  
# 运行模式包括PRD（生产环境）、DEV（开发环境）  
Client.Mode=PRD  
  
# 自动重发挂起未发送的消息  
Client.AutoSendPendingMessages=true  
  
# 自动同步交换审计日志  
Client.AutoSendPendingLogs=true  
  
# 平台连接超时设置（以毫秒为单位，1小时）  
Client.Connection.Timeout=3600000  
  
# 接收消息超时设置（以毫秒为单位，5秒）  
Client.Receive.Timeout=5000
```

配置正确的交换平台节点 URI，在开发阶段配置 Client.Mode=DEV，正式上线运行时配置 Client.Mode=PRD，其他一般采用默认配置即可。

（2）用户凭证文件“credentials.properties”

```
# 用户名和密码
Client.User=GONGSHANG
Client.Password=ENC(mYRkg+4Q4hua1kvpCCI2hg==)
```

4.2 消息结构

消息数据包由消息头（Header）、消息属性（Properties）、消息内容（Body）及附件（Attachments）组成。

4.2.1 消息头（Header）

属性	类型	说明
MessageID	String	消息标识（自动生成）
CorrelationID	String	当前消息相应的原消息ID
MessageType	String	消息类型（自动生成）
SourceNode	String	来源节点
DestinationNode	String	目标节点
BizID	String	业务主键（可以为复合主键，用 分割）
BizTime	DateTime	业务时间
SubjectName	String	消息对应数据主题名称
SubjectCode	String	消息对应数据主题唯一标识
AppID	String	分配给客户端调用的系统ID
Version	String	接口版本信息（预留）
Priority	Int	优先级。
GroupID	String	消息分组ID（用于消息分组传输）
GroupSeq	Int	分组排序（最后一条消息序号为-1）
Sign	String	签名结果
SignMethod	String	签名方法
SignType	Int	签名类型（0不签名，1内容签名，2内容加参数签名）

CreationTime	DateTime	消息创建时间（API自动设置）
IsRedelivered	boolean	是否API重发消息（API自动设置）
DeliveredTime	DateTime	消息发送时间（API自动设置）

4.2.2 消息属性（Properties）

属性	类型	说明
Properties	HashMap<String, Object>	不同消息类型有不同的定义属性；也可由交换桥接系统设置的自定义消息属性（NameValue形式）

4.2.3 消息内容（Body）

交换平台 SDK 支持四种消息内容类型：

序	类型	适用场景
1	Text 文本消息	消息的数据内容是字符串类型
2	Object 对象消息	消息的数据内容是可序列化的对象
3	Blob 文件消息	消息的数据内容是一个文件
4	Composite 复合消息	消息的数据内容是由一个正文和多个附件组成

4.2.3.1 文本消息（TextMessage）

文本消息的内容是字符串类型。

属性	类型	说明
Text	String	消息文本

4.2.3.2 对象消息（ObjectMessage）

对象消息的内容是可序列化的对象类型。

属性	类型	说明
Object	Serializable	可序列化的对象类型（Java POJO或者.Net可序列对象）

4.2.3.3 文件消息（BlobMessage）

文件消息的内容是一个文件。文件消息的属性包括：

属性	类型	说明
FileName	String	文件名
FileSize	Long	文件大小
MimeType	String	文件类型
BinaryData	byte[]	文件二进制
FileProperties	HashMap<String, Object>	文件自定义属性

4.2.3.4 复合消息（CompositeMessage）

复合消息的内容是一个正文对象+多个文件附件。

属性	类型	说明
Object	Serializable	可序列化的对象类型（Java POJO或者.Net可序列对象）
Attachments	ArrayList<Attachment>	附件列表

复合消息的每个文件附件属性包括：

属性	类型	说明
FileName	String	附件名
FileSize	Long	附件大小
MimeType	String	附件类型
BinaryData	byte[]	文件二进制
FileProperties	HashMap<String, Object>	附件自定义属性

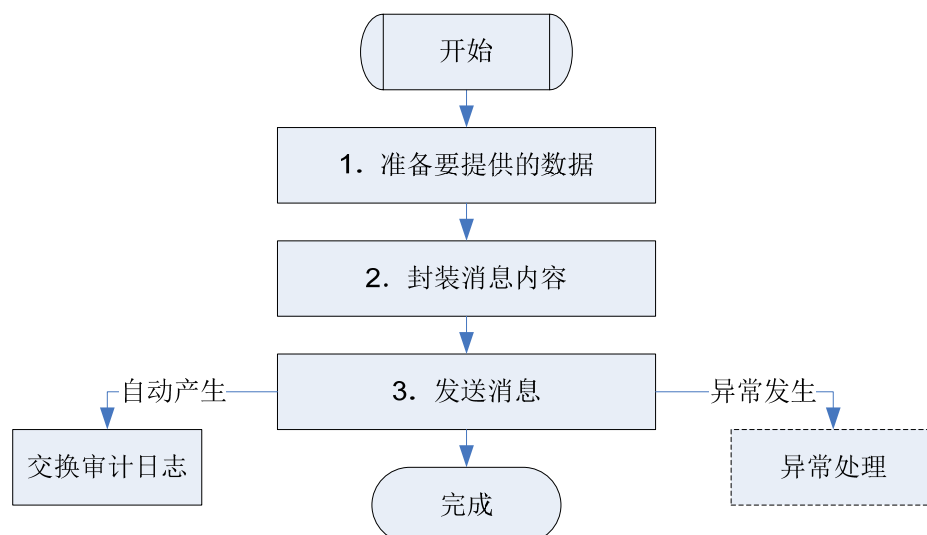
4.3 发送消息

4.3.1 操作流程

政务部门通过消息交换方式向数据中心提供数据，交换桥接系统只需按主

题把数据封装成为消息数据包，调用交换接口 API 发送到前置交换系统即可。由交换平台将消息即时传输到数据中心，并按指定的目标发送到接收消息的政务部门前置交换系统。

交换桥接通过消息方式提供数据时，应参照消息发送流程进行，消息发送操作的详细步骤如下：



（1）准备要提供的数据：交换桥接系统从部门的业务库（业务视图）抽取作为消息内容的业务数据。

（2）封转消息内容：填充消息交换的数据内容，并指定消息交换的目标。

（3）发送消息：交换桥接调用交换接口 API，发送消息。交换接口 API 自动记录交换审计日志（日志详细数据结构参考“4.4.2 日志结构”）。

4.3.2 开发示例

（1）发送消息的.NET（C#）开发示例

```
// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.Open();

// 3. 准备发送消息
TextMessage message = client.CreateTextMessage();
// 设置目的节点
message.DestinationNode = "DISHUI";
// 设置消息内容
message.Text = "这是一个示例";
```



```

// TODO 设置其它属性

// 4. 发送消息
try
{
    client.Send(message);
}
catch (Exception ex)
{
    // TODO: 异常处理
}

// 5. 关闭平台连接
client.Close();

```

(2) 发送消息的 Java 开发示例

```

// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.open();

// 3. 准备发送消息
TextMessage message = client.createTextMessage();

// 设置目的节点
message.setDestinationNode("DISHUI");
// 设置消息内容
message.setText("这是一个示例");
// TODO 设置其它属性

// 4. 发送消息
try
{
    client.send(message);
}
catch (DatacenterException ex)
{
    // TODO: 异常处理
}

// 5. 关闭平台连接
client.close();

```

(3) 发送复合消息的.NET (C#) 开发示例

```
// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.Open();

// 3. 准备发送消息
CompositeMessage message = client.CreateCompositeMessage();
// 设置目的节点
message.DestinationNode = "DISHUI";
// 设置消息内容
// 平台注册对象类型
BizObject bizObject = new BizObject();
...
message.Object = bizObject;
// TODO 设置其它属性

// 添加附件
message.AddAttachment("D://Send//附件一.txt");

// 4. 发送消息
try
{
    client.Send(message);
}
catch (Exception ex)
{
    // TODO: 异常处理
}

// 5. 关闭平台连接
client.Close();
```

(4) 发送复合消息的 Java 开发示例

```
// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.open();
```

```

// 3. 准备发送消息
CompositeMessage message = client.createCompositeMessage();

// 设置目的节点
message.setDestinationNode("DISHUI");
// 设置消息内容
// 平台注册对象类型
BizObject bizObject = new BizObject();
message.setObject(bizObject);
// 添加附件
message.addAttachement("D://Send//附件一.txt");
// TODO 设置其它属性

// 4. 发送消息
try
{
    client.send(message);
}
catch (DatacenterException ex)
{
    // TODO: 异常处理
}

// 5. 关闭平台连接
client.close();

```

4.3.3 相关说明

（一）封装消息

交换桥接在发送消息时通过设置消息 DestinationNode 指定消息发送目标节点，发送目标使用全市统一的编码表示。编码表由市数据中心交换平台提供，详见附录。通过交换平台发送的消息只能指定一个目标，如同一业务数据需要发送给多个目标时，则需发送多条内容相同目标不同的消息。通过 SubjectCode 指明消息对应的数据主题，通过 BizID 和 BizTime 设置业务唯一标识和业务发生时间。

（二）异常处理

交换桥接在调用接口 API（send）发送消息时，会发生非法消息发送异常、

服务不可用异常等异常。

（1）对于非法消息发送异常，API 会在审计日志中记录异常信息，交换桥接需要定时检查这些日志，修正消息并重新发送。详细的处理方法见“4.6.3 非法消息发送异常处理”。

（2）对于服务不可用异常，API 会在审计日志中记录异常信息。政务部门只需确保交换桥接与前置机网络连通，在交换桥接下一次连接交换平台是，接口 API 自动重传由于该类异常造成发送失败的消息，政务部门也可通过修改配置，手动重传消息。详细的处理方法见“4.6.2 服务不可用异常”。

（三）衔接业务系统

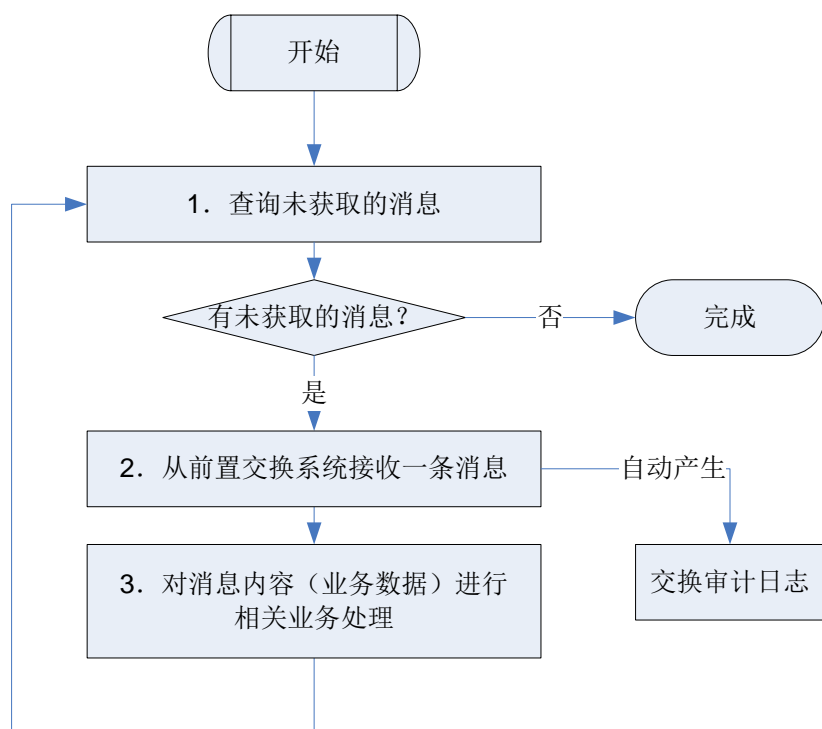
交换桥接系统须实现从业务库即时抽取业务数据。一般情况下，交换桥接系统可以通过以下方式与业务系统衔接：

- （1）定期轮询业务数据库，抽取新增的业务数据发送；
- （2）通过数据库触发器即时发现新增的业务数据并发送；
- （3）为保持业务系统与交换平台的松耦合，不建议业务环节（业务系统内部）调用交换接口 API 发送消息。

4.4 接收消息

4.4.1 操作流程

使用方的交换桥接系统应根据自己业务需求，实时或定时从前置交换系统接收消息。接收消息的操作步骤如下：



(1) 查询未获取的消息：交换桥接系统通过 API 查询前置交换系统中未获取的消息数目。

(2) 从前置交换系统接收一条消息：交换桥接系统从前置机交换系统接收一条消息，解包消息并获取消息内容（业务数据）。交换接口 API 自动记录交换审计日志（日志详细数据结构参考“4.4.2 日志结构”）。

(3) 进行业务处理：对获取的业务数据进行相关业务处理。

4.4.2 开发示例

4.4.2.1 同步接收消息

(1) 同步接收消息的.NET (C#) 开发示例

```

// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.Open();

// 3. 获取消息
try
{
    Message message = client.Receive();
    // 4. TODO: 处理获取消息，例如写入业务库
}
  
```

```

        if (message is CompositeMessage)
        {
            CompositeMessage compMessage = (CompositeMessage)message;
            // TODO 保存复合消息信息
            foreach (Attachment attachment in compMessage.Attachments)
            {
                // TODO 保存附件
            }
        }
        else if (message is BlobMessage)
        {
            BlobMessage blobMessage = (BlobMessage)message;
            // TODO 保存文件消息信息
        }
        else if (message is ObjectMessage)
        {
            ObjectMessage objMessage = (ObjectMessage)message;
            // TODO 保存对象消息信息
            BizObject biz = objMessage.GetObject(typeof(BizObject)) as
BizObject;

        }
        else if (message is TextMessage)
        {
            TextMessage textMessage = (TextMessage)message;
            // TODO 保存文本消息信息
        }
    }
    catch (Exception ex)
    {
        // TODO: 异常处理
    }

    // 5. 关闭平台连接
    client.Close();

```

(2) 同步接收消息的 Java 开发示例

```

// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.open();

```

```

// 3. 获取消息
try
{
    Message message = client.receive();
    // 4. TODO: 处理获取消息，例如写入业务库
    if (message instanceof CompositeMessage)
    {
        CompositeMessage compMessage = (CompositeMessage) message;
        // TODO 保存复合消息信息
        for (Attachment attach: compMessage.getAttachments()) {
            // TODO 保存附件
        }
    }
    else if (message instanceof ObjectMessage)
    {
        ObjectMessage objectMessage = (ObjectMessage) message;
        // TODO 保存对象消息信息

    }
    else if (message instanceof BlobMessage)
    {
        BlobMessage blobMessage = (BlobMessage) message;
        // TODO 保存文件消息信息
    }
    else if (message instanceof TextMessage)
    {
        TextMessage textMessage = (TextMessage) message;
        // TODO 保存文本消息信息
    }
}
catch (Exception ex)
{
    // TODO: 异常处理
}

// 5. 关闭平台连接
client.close();

```

4.4.2.2 循环接收消息

(1) 循环同步接收消息的.NET (C#) 开发示例

```

// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

```

```

// 2. 打开交换平台连接
client.Open();

// 3. 获取消息
try
{
    // 4. 检查是否有更多消息
    while (client.HasMoreMessages())
    {
        Message message = client.Receive();
        // 5. TODO: 处理获取消息, 例如写入业务库

        // 等待5秒后继续
        Thread.Sleep(5000);
        Console.WriteLine("输入y回车继续, 请输入");
        if (!Console.ReadLine().Equals("y"))
            break;
    }
}
catch (Exception ex)
{
    // TODO: 异常处理
}

// 6. 关闭平台连接
client.Close();

```

(2) 循环同步接收消息的 Java 开发示例

```

// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.open();

// 3. 获取消息
try
{
    while (client.hasMoreMessages())
    {
        Message message = client.receive();
        // 4. TODO: 处理获取消息, 例如写入业务库
    }
}

```



```

        // 等待5秒后继续
        Thread.Sleep(5000);

    }
}
catch (Exception ex)
{
    // TODO: 异常处理
}

// 5. 关闭平台连接
client.close();

```

4.4.2.3 监听接收消息

(1) 监听接收消息的.NET (C#) 开发示例

```

// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.Open();

try
{
    // 3. 注册监听程序
    MessageListener listener = new MessageListener();
    client.Listener += listener.OnMessage;
    client.StartListener();

}
catch (Exception ex)
{
    // TODO: 异常处理
}

```

实现事件委托

```

/// <summary>
/// 数据获取事件委托
/// </summary>
public override void OnMessage(Message message)
{
    // TODO : 处理获取消息，例如写入业务库
}

```

（2）监听获取消息的 Java 开发示例

```
// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.open();

// 3. 注册监听程序
try
{
    client.startListener(new MessageListener());
}
catch (Exception ex)
{
    // TODO: 异常处理
}
```

实现事件监听接口

```
public void onMessage(Message message) {
    // TODO : 处理获取消息，例如写入业务库
}
```

4.4.3 相关说明

（一）接收方式

可供交换桥接系统选择的消息接收模式包括三种：

（1）receive 方式，在限定时间内同步接收一条消息。如果在指定时间内没有消息到达，系统将返回空值。如果指定时间为 0 时，则系统将一直等待直到有消息到达。

（2）receiveNoWait 方式，接收下一条即时消息。如果当前没有可接收的消息，系统将立即返回空值。

（3）onMessage 方式，有消息到则通知触发接收消息。

详见“交换接口”的“接收消息”方法说明。

（二）识别消息

交换桥接系统接收到消息时，通过 SourceNode 获知消息来源，通过

SubjectCode 获取消息对应的数据主题，通过 BizID 和 BizTime 获知对应的业务唯一标识和业务发生时间。

（三）衔接业务系统

交换桥接系统接收到消息后，需实现业务相关的消息内容处理（写入业务库或调用相关业务服务）。

4.5 交换审计

4.5.1 审计内容

一般情况下，交换数据会经过提供方业务系统、提供方交换桥接、提供方前置机、中心节点、获取方前置机、获取方交换桥接、获取方业务系统七个节点。交换审计就是对账数据从产生到被利用全过程的传输一致性。其中提供方业务系统与提供方交换桥接，获取方交换桥接与获取方业务系统的对账由政务部门负责；交换平台提供数据从提供方交换桥接到获取方交换桥接的审计对账。

4.5.2 日志结构

字段名	数据类型	含义说明
MessageID	String	消息唯一标识（平台自动生成）
CorrelationID	String	相关消息标识
MessageType	String	消息类型
SourceNode	String	来源节点
DestinationNode	String	目标节点
CurrentNode	String	当前节点
BizID	String	业务主键（可以为复合主键，用 分割）
BizTime	DateTime	业务时间
SubjectName	String	消息对应数据主题名称
SubjectCode	String	消息对应数据主题唯一标识
AppID	String	分配给客户端调用的系统 ID
Version	String	接口版本信息（预留）
Priority	Int	优先级。

CreationTime	DateTime	消息创建时间
IsRedelivered	boolean	是否 API 重发消息
DeliveredTime	DateTime	消息发送时间
LogTime	DateTime	日志时间
LogPhase	String	日志阶段
LogStatus	Int	运行结果（成功 0，失败 1）
ErrorCode	String	错误类型
ErrorMessage	String	错误信息

审计日志使用文件记录，文件名为“yyyy-mm-dd.log”，存储路径为 auditlog。
审计日志的字段和内容字段说明如下：

（1）MessageID 字段作为消息的物理唯一标识，由交换接口 API 自动生成。物理唯一标识与业务上的唯一标识不同，同一交换桥接发出的消息可以存在物理唯一标识不同但业务唯一标识相同的数据，两者是“一对一”或“多对一”关系。MessageID 字段无需业务部门的交换桥接维护。

（2）SourceNode、DestinationNode、CurrentNode 字段是消息的路由信息，SourceNode 是消息发出的源节点，DestinationNode 是消息要传送到的目标节点，CurrentNode 是记录本审计日志的节点。交换桥接在发送消息时，需填写以上三个路由信息。

（3）SubjectName、SubjectCode 字段是消息内容对应的数据主题。交换桥接系统需要交换的业务记录须先在数据中心建立主题。

（4）Priority 消息发送时的优先级。前置交换系统会根据优先级和写入时间进行排队发送。

（5）BizTime、CreationTime、DeliveredTime、LogTime 字段是分别是消息内容（业务数据）的发生时间、消息的创建时间、消息被成功发送的时间、记录日志的时间。首先业务数据在业务系统生产，记录业务发生时间；然后交换桥接抽取业务数据，封装为消息，记录消息创建时间；成功向下一节点发送消息，记录发送时间，若第一次发送消息成功，发送时间与创建时间一致，若消息经过重发，则发送时间比创建时间大。BizTime 需交换桥接填写，其他三个时间均由交换接口 API 生产，无需交换桥接维护。

（6）ErrorCode、ErrorMessage 字段是记录消息发送过程中的异常信息，包

括异常类型和详细异常情况。

4.5.3 审计实施

审计内容按审计分工，可分为两部分：一是由政务部门负责的业务系统与交换桥接的审计；二是由交换平台负责的从提供方交换桥接到获取方交换桥接的传输过程审计。

(1) 政务部门的审计

通过比对交换桥接记录的审计日志与业务系统数据量的比对，政务部门可以审计交换桥接发送的数据是否与业务系统产生的数据一致。交换接口 API 提供的审计日志的查询方法，详见“4.5.6 交换审计”。

(2) 交换平台的审计

消息传输的每一个环节均记录审计日志，交换平台通过对审计日志的比对，实现从提供方桥接程序到获取方桥接程序过程的审计对账。政务部门可以通过电子政务外网访问广州市电子政务数据中心系统查看审计结果。

系统访问地址：<http://dc.gz.gov.cn/services>

系统截图如下：

交换服务

消息审计

?

提供单位

数据主题

统计

发送目标

发送时间

只显示对账异常的统计记录

提供单位	数据主题	发送目标	提供消息		提供方前置机	中心节点	获取方前置机	获取消息	状态
			计划提供消息	成功提供消息					
	行政		10	10	10	10	--	10	● 正常
			8	8	8	8	8	7	● 获取过程异常
	登记		8	8	8	8	8	8	● 正常
			10	9	9	9	9	9	● 提供过程异常

4.6 异常处理

4.6.1 异常分类

消息交换过程中，一般会发生三类异常：

序	分类	说明
1	服务不可用异常	交换桥接无法正常打开前置机的消息队列。交换接口 API 会抛出异常并记录日志。
2	非法消息发送异常	交换桥接发送的消息内容不符合数据标准规范或指定了非法的目标。交换接口 API 会抛出异常并记录日志。
3	数据量变化异常	交换桥接向前置机发送的消息数量较平均值出现大幅波动（放大或缩小 10 倍），或消息数量为 0。由数据中心通过人工监控手段发现异常。

4.6.2 服务不可用异常

（一）异常的发生

在交换桥接调用交换接口 API 发送消息时，交换桥接和前置机网络不可连通或前置机发生故障都可能导致服务不可用的异常的发生。

同样，交换桥接调用交换接口 API 获取消息时也会产生此类异常。

（二）交换平台提供的支撑

发生异常时，交换接口 API 会抛出异常，并在审计日志中标记交换失败的消息以及记录异常信息。

交换接口 API 提供了自动重传消息和手动重传消息的方法，详见异常处理过程。

（三）异常处理过程

（1）服务不可用异常主要由网络断开或前置机故障引发，因此该类故障发生时，政务部门应检查交换桥接与前置机间的网络环境，保障网络连通性，并联系数据中心管理员检查前置机，以保障前置机正常。

（2）在服务可用后，交换桥接需要重新传输由于异常导致发送失败的消息或重新获取消息。对于重传消息，交换接口 API 提供自动重传消息和手工重传消息两种方式，其中前者是默认方式；对于重新获取消息，获取方交换桥接程序只需再次运行即可完成消息的重新获取。

自动重传消息：

修改配置文件“config.xml”的“AutoSendPendingMessages=true”。在交换

桥接系统下一次打开平台连接时，交换接口 API 就会自动查询待重传的消息并完成重传。

手工重传消息：

修改配置文件“config.xml”的“AutoSendPendingMessages=false”。交换桥接系统可以按需调用交换接口 API 完成消息重传。

手工重传消息的.NET 开发示例如下：

```
// 1. 初始化交换接口
DatacenterClient client = new DatacenterClient();

// 2. 打开交换平台连接
client.Open();

// 3. 重传挂起消息
try
{
    if (client.ExistsPendingMessage())
        client.SendPendingMessages();
}
catch (Exception ex)
{
    // TODO: 异常处理
}

// 4. 关闭平台连接
client.Close();
```

4.6.3 非法消息发送异常

（一）异常的发生

交换桥接调用交换接口 API 发送消息时，交换内容（业务数据）不符合数据规范或者指定了非法的目标，就会导致非法消息发送异常。

交换接口 API 每一次发送消息时，都会将消息中的数据与事前已在数据中心注册的数据主题（数据标准）进行比对，检查数据项（字段）取值的合法性，并拒绝发送取值不合法的数据。

交换桥接在封装消息时指定的目标超出广州市消息交换目标编码表（见附录 3）的取值范围，在消息发送时会抛出非法消息发送异常。

（二）交换平台提供的支撑

发生异常时，交换接口 API 会抛出异常，并在审计日志中标记交换失败的消息以及记录异常信息。

（三）异常处理过程

（1）异常发生时，政务部门需要分析审计日志中的异常信息，定位造成异常的原因，修正消息内容或发送目标。修正消息内容一般由业务人员通过业务系统完成。

（2）交换桥接重新封装消息并发送消息。

4.6.4 数据量变化异常

（一）异常的发生

数据量变化异常主要由交换平台通过交换监控发现。

交换平台对消息交换过程实施每天监控，当政务部门消息交换量为 0 时，视为数据量变化异常。例如交换桥接运行失败、服务长期不可用、当天实际业务量为 0 都会引发异常。

交换平台对消息交换过程实施每天监控，当政务部门消息交换量放大或缩小 10 倍时，视为数据量变化异常。例如业务高峰日、连续发生非法消息发送异常都会引发数据量变化异常。

（二）交换平台提供的支撑

交换平台对消息交换过程实施每天监控。对消息交换量为 0 或消息交换量急剧变动的政务部门，通过邮件发送预警提示。

（三）异常处理过程

政务部门在收到预警提示时，应按照提示检查交换桥接运行情况及业务系统运行情况。若发现异常应及时处理，并将检查处理情况邮件回复交换平台管理员；若交换桥接和业务系统运行均正常，则回复交换平台管理员，说明数据变动属正常情况。

4.7 交换接口

交换接口 API 提供了平台连接、发送消息、接收消息、异常处理、交换审计等五类 API：

序	类型	方法
1	平台连接	open close
2	发送消息	send
3	接收消息	receive hasMoreMessages
4	异常处理	existsPendingMessage sendPendingMessages
5	交换审计	getAuditLogs

4.7.1 平台连接

- **open**

打开平台连接（Java）

方法声明	<code>public int open() throws DatacenterException</code>
功能描述	打开交换平台连接，进行用户验证和资源配置。
输入参数	无
返回值	状态代码
异常抛出	<p>（1）DatacenterException：由于内部错误导致打开交换平台连接失败。</p> <p>（2）InvalidClientException：非法用户导致验证失败，应检查交换接口 API 配置文件的用户和密码设置。</p> <p>（3）InvalidNodeException：非法交换节点配置导致，应检查交换接口 API 配置文件的交换节点设置。</p>
调用示例	<code>DatacenterClient client = new DatacenterClient(); client.open();</code>
备注	

- **close**

关闭平台连接（Java）

方法声明	<code>public void close()</code>
功能描述	关闭交换平台连接，释放所有占用资源。
输入参数	无
返回值	无
异常抛出	无
调用示例	<pre>DatacenterClient client = new DatacenterClient(); client.open(); client.close();</pre>
备注	可在配置文件中设置是否在关闭平台连接前自动同步未提交给前置交换系统的交换审计日志以及挂起的消息。

4.7.2 发送消息

- **send**

发送消息（Java）

方法声明	<code>public void send(Message message) throws DatacenterException</code>
功能描述	发送消息数据包到指定的目标节点。目标节点在消息的 <code>DestinationNode</code> 属性声明。
输入参数	message: 要发送的消息数据包。
返回值	无。
异常抛出	<code>DatacenterException</code> : 由于内部错误导致发送消息失败。 <code>InvalidDestinationException</code> : 非法目标节点，消息中声明的目标节点无效。

	InvalidMessageException: 非法消息。
调用示例	<pre> DatacenterClient client = new DatacenterClient(); client.open(); Message message = client.createTextMessage(); message.DestinationNode = "GONGSHANG"; message.Text = "消息内容"; client.send(message); client.close(); </pre>
备注	

4.7.3 接收消息

● receive

同步接收下一条消息（Java）

方法声明	<pre> public message receive() throws DatacenterException </pre>
功能描述	同步接收下一条消息。
输入参数	无。
返回值	下一条要接收的消息。
异常抛出	DatacenterException: 由于内部错误导致发送消息失败。
调用示例	<pre> DatacenterClient client = new DatacenterClient(); client.open(); Message message = client.receive(); Client.close(); </pre>
备注	该方法调用直到有消息要接收或者交换平台连接已关闭才返回。

● receive

在指定时限内接收下一条消息（Java）

方法声明	<pre>public message receive(long timeout) throws DatacenterException</pre>
功能描述	在指定时限内接收下一条消息。
输入参数	timeout: 超时时间（以毫秒为单位），0 表示永不超时（即同步接收）。
返回值	下一条要接收的消息。
异常抛出	DatacenterException: 由于内部错误导致发送消息失败。
调用示例	<pre>DatacenterClient client = new DatacenterClient(); client.open(); Message message = client.receive(1000); Client.close();</pre>
备注	

● receiveNoWait

接收下一条即时消息（Java）

方法声明	<pre>public message receiveNoWait() throws DatacenterException</pre>
功能描述	接收下一条即时消息（如果没有消息不等待）。
输入参数	无。
返回值	返回下一条即时要接收的消息。 当没有消息可即时接收时返回 null。
异常抛出	DatacenterException: 由于内部错误导致发送消息失败。 InvalidDestinationException: 非法目标节点，消息中声明的目标节点无效。
调用示例	<pre>DatacenterClient client = new DatacenterClient(); client.open(); Message message = client.receiveNoWait();</pre>

	<code>client.close();</code>
备注	

● **hasMoreMessages**

检查当前是否还有未接收的消息（Java）

方法声明	<pre>public boolean hasMoreMessages() throws DatacenterException</pre>
功能描述	检查当前是否还有未接收的消息。
输入参数	无。
返回值	<p>true: 存在未接收的消息。</p> <p>false: 不存在未接收的消息。</p>
异常抛出	DatacenterException: 由于内部错误导致发送消息失败。
调用示例	<pre>if(client.hasMoreMessages()) Message msg = receiveNoWait();</pre>
备注	

4.7.4 异常处理

● **existsPendingMessage()**

检查是否存在挂起的本地消息（Java）

方法声明	<pre>public boolean existsPendingMessage() throws DatacenterException</pre>
功能描述	检查是否存在挂起在本地的消息。

输入参数	无。
返回值	true: 当存在挂起的本地消息。 false: 当没有挂起的本地消息。
异常抛出	DatacenterException: 由于内部错误导致检查失败。
调用示例	<pre>DatacenterClient client = new DatacenterClient(); client.open(); if(existsPendingMessage()){ client.sendPendingMessage(); }</pre>
备注	

● sendPendingMessages()

发送所有挂起的本地消息（Java）

方法声明	<pre>public int sendPendingMessages() throws DatacenterException</pre>
功能描述	发送所有挂起的本地消息。
输入参数	无。
返回值	成功发送的消息数量。
异常抛出	DatacenterException: 由于内部错误导致发送消息失败。
调用示例	<pre>DatacenterClient client = new DatacenterClient(); client.open(); if(existsPendingMessage()){ client.sendPendingMessages(); }</pre>
备注	

4.7.5 交换审计

● **getAuditLogs(Date begin, Date end)**

获取指定时间段的交换审计日志（Java）

方法声明	<code>public List<AuditLog> getAuditLogs(Date begin, Date end) throws DatacenterException</code>
功能描述	获取指定时间段的交换审计日志。
输入参数	begin: 审计开始时间。 end: 审计结束时间。
返回值	指定时间段的交换审计日志列表。
异常抛出	DatacenterException: 由于内部错误导致获取失败。
调用示例	<code>DatacenterClient client = new DatacenterClient(); client.open(); List<AuditLog> logs = client.getAuditLogs();</code>
备注	

5 数据库交换

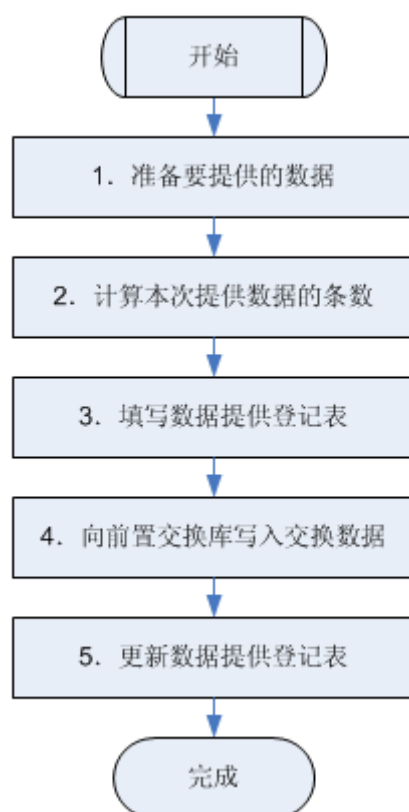
5.1 提供共享数据

5.1.1 操作流程

政务部门通过数据库方式向数据中心提供共享数据，交换桥接只需要定期按主题按批次，把新增数据通过交换库的提供数据接口写入前置交换库即可。由交换平台增量采集到数据中心，并按需分发到各授权使用方。

交换桥接向前置机交换库添加交换数据时，应参照数据提供流程进行，数

据提供操作的详细步骤如下：



图表 5-1 通过交换库接口提供数据的步骤

1. 准备要提供的数据：交换桥接从业务库（业务视图）增量抽取将要写入前置机交换库的业务数据。

2. 计算本批次提供数据的记录条数：交换桥接计算本批次所抽取出的业务数据的总条数 N。

3. 填写数据提供登记表：交换桥接按规定向数据提供登记表添加一条记录，该记录包含了批次号、交换数据表名、提供的数据记录总条数 N（其他字段使用缺省值）。

4. 向前置交换库写入交换数据：交换桥接把准备好的交换数据按标准写入前置机交换库。

5. 更新数据提供登记表：交换桥接更新第 3 步中所添加的数据提供登记记录，表示该批次操作结束，操作结束的时间将记录在 S_LAST_UPDATED 字段中。

5.1.2 开发示例

提供数据操作的 Oracle 示例脚本如下（以工商提供开业登记数据主题为例）：

```
declare
    v_sid varchar2(38); /* 批次号 */
    v_result int; /* 操作执行结果 */
begin
    /* 准备要提供的开业登记业务数据，假定记录条数为 N */

    /* 调用开始数据提供接口，准备提供 N 条数据 */
    EXDB.STP_BEGIN_PROVIDE('EX_GONGSHANG_0001_KYDJ', N, v_sid);
    commit;

    /* 如果 v_sid 不为空，则表示可以开始提供数据，否则表示可能最近一次提供数据的过程尚未完成，不能开始新的一次数据提供 */
    if (v_sid is not null) then
        /* 向 EX_GONGSHANG_0001_KYDJ 表添加 N 条交换数据，参照“添加交换数据” */
        */
        .....
    else
        /* 异常处理 */
    end if;

    /* 调用结束数据提供接口，通知交换平台数据提供完成 */
    EXDB.STP_END_PROVIDE(v_sid, -1, v_result);
    commit;
end;
```

提供数据操作的 SQL SERVER 示例脚本如下（以工商提供开业登记数据主题为例）：

```
declare @v_sid varchar(40);--批次号
declare @v_result int; --本批次数据提供是否出现异常
declare @v_count int; --本批次需要提供的数据量

set @v_count = 100;
--假设本批次提供的数据量为100条。数量可以预先计算也可以在写入数据后计算

exec STP_BEGIN_PROVIDE 'EX_GONGSHANG_01_KYDJ', -1, @v_sid output;

if @v_sid is not null
begin
    /** 写入本批次需要提供的数据，写入的数据量为@v_count条**/
    insert into EX_GONGSHANG_01_KYDJ(QYZCH, QYMC, FDDBR, ...)
    ... (VALUES语句或者子查询，参考[添加交换数据])
```

```

        /** 提供完成以后需要通知交换平台**/
        exec STP_END_PROVIDE @v_sid, @v_count, @v_result output;
    end
else
    begin
        select '上一个批次数据提供没有完成,暂时不能提供数据';
        /** 如果您确认当前没有其他程序在提供数据,则可能是上次
            数据提供操作事务异常中止,没有通知交换平台数据已经
            提供完成.这时需要重置上次数据提供操作**/
        exec STP_GET_LAST_ERROR 'EX_GONGSHANG_01_KYDJ',
            @v_sid output, @v_result output;
        if @v_sid is not null
            exec STP_RESET_PROVIDE @v_sid,@v_result output;
    end
end

```

5.1.3 实施说明

(1) 异常处理

当调用 STP_BEGIN_PROVIDE 接口, 输出参数 P_SID 为空值时, 表示最近一次的数据提供操作仍未结束或者有异常情况发生, 需要进行异常处理: 调用 STP_GET_LAST_ERROR 接口, 如果输出参数 P_RESULT > 0, 则表示数据提供操作已超时或者出错, 应调用 STP_RESET_PROVIDE 接口进行重置操作, 该操作中已写入的数据将自动被废置; 如果输出参数 P_RESULT < 0, 表示任务正在运行, 需要等待操作结束; 如果输出参数 P_RESULT = 0, 表示可以正常开始数据提供操作。异常处理流程图如下所示:

异常处理的 Oracle 示例脚本如下 (以检查工商提供的开业登记数据主题为例):

```

declare
    v_last_sid varchar2(38);  /* 最近一次数据提供的批次号 */
    v_result int;             /* 最近一次数据提供的错误代码 */
    v_reset_result int;       /* 重置操作执行结果 */
begin
    /* 获取上一次数据提供的错误信息 */
    EXDB.STP_GET_LAST_ERROR('EX_GONGSHANG_0001_KYDJ', v_last_sid,
        v_result);
    commit;
    if (v_result > 0) then
        EXDB.STP_RESET_PROVIDE(v_last_sid, v_reset_result);
    end if;
end

```

```

        commit;
    elsif (v_result < 0) then
        /* 上一次操作未完成，需要继续等待 */
    else
        /* 可以开始数据提供操作 */
    end if;
end;

```

（2）添加交换数据

一般情况，交换桥接把业务数据写入前置机交换库都是添加操作。交换桥接只需把业务数据写入交换数据表的内容字段，对内建字段只需采用缺省值。

Oracle 示例脚本如下：

```

insert into EXDB.EX_GONGSHANG_0001_KYDJ
(QYZCH, QYMC, FDDBR, SFZJHM, ZS, ...)
values
('注册登记号', '企业名称', '法定代表人', '身份证件号码', '住所', ...)

```

SQL SERVER 示例脚本如下：

```

insert into EX_GONGSHANG_01_KYDJ(QYZCH, QYMC, FDDBR,...)
values('企业注册号','企业名称','法定代表人',...);

```

（3）修改交换数据

只有对已添加到前置机交换库的业务数据记录确认为有错误时，才能执行修改交换数据操作，此外，不应执行修改交换数据操作。修改交换数据时，应根据业务唯一标识号 BIZ_ID 定位有错误的数据记录，更新错误数据为正确数据，把数据状态字段 S_STATUS 的值设为 1（修改），把 S_LAST_UPDATED 字段设为当前时间。

Oracle 示例脚本如下：

```

update EXDB.EX_GONGSHANG_0001_KYDJ
set
QYZCH = '注册登记号', QYMC = '企业名称 1', FDDBR = '正确的法定代表人 1', ...,
S_STATUS = 1, S_LAST_UPDATED = systimestamp
where BIZ_ID = '唯一业务流水号'

```

SQL SERVER 示例脚本如下：

```

update EX_GONGSHANG_01_KYDJ set

```

```

QYZCH = '企业注册号',
QYMC = '企业名称',
...
S_STATUS=1,
S_LAST_UPDATED = getdate(),
where BIZ_ID = '唯一业务流水号'

```

（4）废置交换数据

当已添加到前置机交换库的业务数据被确认为无效记录时，可以执行废置交换数据操作来进行逻辑删除（不允许执行物理删除）。废置交换数据时，应根据业务唯一标识号 BIZ_ID 定位确认为无效的数据记录，把数据状态 S_STATUS 字段的值设为 2（废置），并把 S_LAST_UPDATED 字段设为当前时间。

Oracle 示例脚本如下：

```

update EXDB.EX_GONGSHANG_0001_KYDJ
set
S_STATUS = 2, S_LAST_UPDATED = systimestamp
where BIZ_ID = '唯一业务流水号'

```

SQL SERVER 示例脚本如下：

```

update EX_GONGSHANG_01_KYDJ set
S_STATUS=2,
S_LAST_UPDATED = getdate(),
where BIZ_ID = '唯一业务流水号'

```

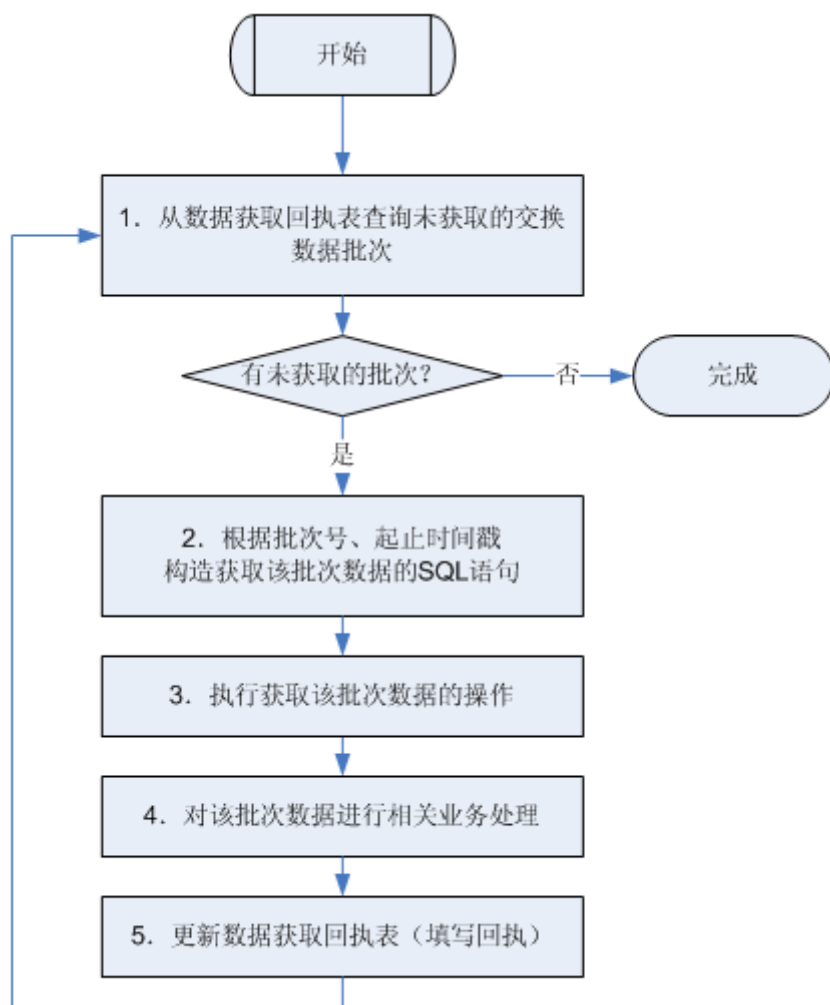
（5）增量提供数据

交换桥接必须实现由业务库增量抽取业务数据到前置机交换库的功能。一般情况下可以通过比较业务发生时间实现增量抽取业务数据。采取这种做法，交换桥接必须自行记录上次提供数据的业务发生时间。

5.2 获取共享数据

5.2.1 操作流程

使用方的交换桥接应根据数据获取回执表，按批次获取由数据中心分发到前置机交换库的交换数据。按批次获取交换数据的操作步骤如下：



图表 5-2 通过交换库接口获取交换数据的步骤

1. 查询批次：从数据获取回执表查询未获取的交换数据批次。
2. 构造 SQL 语句：根据批次号、起止时间戳构造获取该批次数据的 SQL 语句。
3. 执行获取操作：执行第 2 步构造的 SQL 语句，获取该批次数据。
4. 进行业务处理：对获取的批次数据进行业务相关处理。
5. 填写获取回执：更新该批次回执，表示该批次获取操作结束，操作结束的时间将记录在 S_LAST_UPDATED 字段中。

5.2.2 开发示例

按批次获取数据的 Oracle 示例脚本片段如下（以劳社获取公安的出生登记为例）：

```
/* 从前置机交换库中，获取第一批未获取的数据 */
```

```

declare
    type t_cursor is ref cursor;
    v_sid varchar2(38);          /* 批次号 */
    v_begin_stamp timestamp;      /* 批次的开始时间戳 */
    v_end_stamp timestamp;        /* 批次的结束时间戳 */
    v_datas t_cursor;            /* 第一批未获取的交换数据 */
    v_script varchar2(1024);      /* 用于生成脚本 */
    v_count int;                 /* 应该获取到的数据条数 */
    v_check_count int;           /* 实际获取到的数据条数 */
    v_result int;                /* 操作执行结果 */
begin
    /* 调用开始数据获取接口，获取一个批次数据的信息 */
    EXDB.STP_BEGIN_ACQUIRE('EX_GONGAN_0001_CSDJ', v_sid, v_begin
_stamp, v_end_stamp, v_count);
    commit;

    while (v_sid is not null) loop
    begin
        /* 构造获取该批次交换数据的语句 */
        v_script := 'select * from EXDB.EX_GONGAN_0001_CSDJ
            where S_LAST_UPDATED >= :begin_stamp and S_LAST_UPDATED <=
:end_stamp';

        /* 获取该批次的交换数据 */
        open v_datas for v_script using v_begin_stamp, v_end_stamp

        /* 开始使用 v_datas 中的数据，例如经过转换后存放到业务库中 */
        /* 计算实际获取到的记录数，并填到变量 v_check_count 中 */
        /* ... */

        /* 调用结束数据获取接口，更新数据获取回执表 */
        EXDB.STP_END_ACQUIRE(v_sid, v_check_count, v_result);
        commit;

        /* 调用开始数据获取接口，获取下一个批次数据的信息 */
        EXDB.STP_BEGIN_ACQUIRE('EX_GONGAN_0001_CSDJ', v_sid, v_beg
in_stamp, v_end_stamp, v_count);
        commit;
    end loop;
end;

```

按批次获取数据的 SQL SERVER 示例脚本片段如下（以劳社获取公安的出生登记为例）：

```

declare @v_sid varchar(40);--批次号
declare @v_begin_stamp datetime; --批次的开始时间戳
declare @v_end_stamp datetime;--批次的结束时间戳
declare @v_count int; --批次的数据量
declare @v_check_count int; --实际获取到的数据量
declare @v_result int; --获取结果

exec STP_BEGIN_ACQUIRE 'EX_GONGAN_01_CSDJ',
    @v_sid output, @v_begin_stamp output,
    @v_end_stamp output, @v_count output;

if @v_sid is not null
begin
    /** 读取本批次新数据(需要自定义开发代码) **/
    select * from EX_GONGAN_01_CSDJ
    where S_LAST_UPDATED >= @v_begin_stamp
        and S_LAST_UPDATED <= @v_end_stamp;

    /** 计算实际获取到的数据量(需要自定义开发代码) **/
    set @v_check_count=@v_count;

    /** 通知交换平台获取完成**/
    exec STP_END_ACQUIRE @v_sid, @v_check_count, @v_result output;
end
else
begin
    select '没有新数据';
    return;
end

```

5.3 交换接口

5.3.1 数据库结构

本节定义了前置机交换库的数据表结构。前置机交换库的表是由数据中心交换平台管理系统自动创建的，是采用交换库方式接口的交换桥接设计开发的主要依据。前置机交换库主要有交换数据表、数据提供登记表、数据获取回执表。

- 交换数据表。用于存储本部门提供的交换数据以及由数据中心分发的其他部门提供的交换数据。
- 数据提供登记表。提供方使用该表登记执行数据提供操作的信息。
- 数据获取回执表。使用方根据该表信息按批次获取数据，并使用该表填写数据获取回执。

5.3.1.1 交换数据表

交换数据表的命名格式为：EX_提供方交换节点拼音_四位编号_数据主题拼音简写。

交换数据表包括内建字段和内容字段两部分，表结构如下：

交换数据表结构（EX_提供方交换节点拼音简写_4位编号_数据主题拼音简写）				
字段名	数据类型	允许空	缺省值	含义说明
S_GUID	VARCHAR2(38)	否	sys_guid()	数据记录的主键（物理唯一标识）
S_CREATION_TIME	TIMESTAMP	否	systimestamp	数据记录在当前节点交换库的物理创建时间
S_LAST_UPDATED	TIMESTAMP	否	systimestamp	数据记录在提供方交换库的最后修改时间
S_STATUS	VARCHAR2(1)	否	0	数据记录的状态 0：添加 1：修改 2：废置
S_SRC_NODE	VARCHAR2(80)	是		来源节点名称 注：在数据提供或数据获取时，该字段取空值。
内容字段应包含： 唯一的业务流水号 字段：BIZ_ID 业务发生时间 字段：BIZ_TIME （这两个字段命名可以根据实际调整）				

表格 5-1 交换数据表结构

交换数据表的内建字段和内容字段说明如下（以Oracle为例说明）：

1. S_GUID 字段作为交换数据记录的物理唯一标识，缺省值为 sys_guid()。物理唯一标识与业务上的唯一标识不同，同一个交换表中可以存在物理唯一标识不同但业务唯一标识相同的数据。S_GUID 字段无需业务部门的交换桥接维

护。

2. S_CREATION_TIME 是交换数据记录在当前节点交换库的物理创建时间，缺省值为 systimestamp。S_CREATION_TIME 字段无需业务部门的交换桥接维护。

3. S_LAST_UPDATED 是交换数据记录在提供方交换库的最后修改时间。需要由提供方的交换桥接维护。

4. S_STATUS 是交换数据记录的状态，取值范围是 0、1、2。S_STATUS 字段的默认值是 0，需要由提供方的交换桥接维护。

数据状态 S_STATUS 的含义			
状态名	状态值	含义	允许的状态切换
添加	0	交换数据记录初始写入交换库，在当前交换库的创建时间为 S_CREATION_TIME，此时 S_LAST_UPDATED 等于 S_CREATION_TIME	修改、废置
修改	1	提供方在其交换库修改了该交换数据记录，修改时间为 S_LAST_UPDATED	废置
废置	2	提供方已把该交换数据记录标记为无效，标记时间为 S_LAST_UPDATED	无

表格 5-2 数据状态 S_STATUS 的含义

5. 内容字段根据各政务部门的实际业务信息确定，原则上至少应包含唯一的业务流水号字段和业务发生时间字段。

6. S_SRC_NODE 是来源节点名称。在发送数据时，由发送方的交换桥接填写其对应交换节点的值。

提供方交换桥接对提供方交换数据表的权限规定如下：

提供方交换桥接对提供方交换数据表的权限					
对象（表或字段）	SELECT	INSERT	UPDATE	DELETE	权限说明
提供方交换数据表	√	√	√	×	不允许物理删除交换数据表中的记录，只能修改路由状态字段标记数据“已接收”。
S_GUID	√	×	×	×	插入时取缺省值 sys_guid()，不允许修改
S_CREATION_TIME	√	×	×	×	插入时取缺省值

					sysimestamp, 不允许修改
S_LAST_UPDATED	√	×	√	×	插入时取缺省值 sysimestamp, 按规定修改
S_STATUS	√	×	√	×	插入时取缺省值 0, 按规定修改
S_SRC_NODE	√	√	×	×	由交换桥接维护
内容字段	√	√	√	×	内容字段的信息由交换桥接维护

表格 5-3 提供方交换桥接对提供方交换数据表的权限

使用方交换桥接对提供方交换数据表的权限规定如下：

使用方交换桥接对提供方交换数据表的权限					
对象	SELECT	INSERT	UPDATE	DELETE	权限说明
提供方交换数据表	√	×	×	×	使用方对其前置机交换库上的、由数据中心分发的交换数据表只有查询权限

表格 5-4 使用方交换桥接对提供方交换数据表的权限

5.3.1.2 数据提供登记表

数据提供登记表的命名格式为：DC_提供方交换节点拼音_PROVIDE。数据提供登记表包括内建字段（与交换数据表的含义相同）和内容字段两部分，表结构如下：

注：交换桥接一般只需要调用交换平台提供的数据库存储过程接口，而无需直接对该表进行操作。

数据提供登记表结构（DC_提供方交换节点拼音_PROVIDE）				
字段名	数据类型	允许空	缺省值	含义说明
S_GUID	VARCHAR2(38)	否	sys_guid()	数据记录的主键（物理唯一标识）
S_CREATION_TIME	TIMESTAMP	否	sysimestamp	数据记录在当前节点交换库的物理创建时间
S_LAST_UPDATED	TIMESTAMP	否	sysimestamp	数据记录在提供方交换库的最后更新时间

S_STATUS	VARCHAR2 (1)	否	0	数据记录的状态 0: 添加 1: 修改 2: 废置
BEGIN_STAMP	TIMESTAMP			在该批次数据记录中， 按 最 后 更 新 时 间 (S_LAST_UPDATED) 截取的最小时间戳。
END_STAMP	TIMESTAMP			在该批次数据记录中， 按 最 后 更 新 时 间 (S_LAST_UPDATED) 截取的最大时间戳。
EX_TABLE_NAME	VARCHAR2(80)	否		交换数据表的名称
RECORD_COUNT	INT	否		提供方登记该批次提供的 记录条数
CHECK_RESULT	INT	否	0	数据中心对该批次数据 提供进行检查的结果 0: 未检查 1: 数据记录条数一致 2: 数据记录条数不一致
CHECK_TIME	DATE			数据中心对该批次数据 提供进行检查的时间
CHECK_COUNT	INT			数据中心检查得出的该 批次提供的实际记录条 数

表格 5-5 数据提供登记表结构

提供方交换桥接对数据提供方登记表的权限规定如下：

提供方交换桥接对数据提供登记表的权限					
对象（表或字段）	SELECT	INSERT	UPDATE	DELETE	权限说明
数据提供登记表	✓	✓	✓	✗	不允许物理删除数据交 换表中的记录，但可以 通过标记数据状态为 “废置”来逻辑删除
S_GUID	✓	✓	✗	✗	插 入 时 取 缺 省 值 sys_guid(), 不允许修改
S_CREATION_TIME	✓	✗	✗	✗	插 入 时 取 缺 省 值 sys_timestamp, 不允许修 改

S_LAST_UPDATED	√	×	√	×	插入时取缺省值 sysimestamp, 按规定修改
S_STATUS	√	×	√	×	插入时取缺省值 0, 按规定修改
BEGIN_STAMP	√	×	×	×	插入时取缺省值 sysimestamp, 不允许修改
END_STAMP	√	×	√	×	插入时取缺省值 sysimestamp, 按规定修改
EX_TABLE_NAME	√	√	×	×	不允许修改
RECORD_COUNT	√	√	×	×	不允许修改
CHECK_RESULT	√	×	×	×	由数据中心维护
CHECK_TIME	√	×	×	×	由数据中心维护
CHECK_COUNT	√	×	×	×	由数据中心维护

表格 5-6 提供方交换桥接对数据提供登记表的权限

5.3.1.3 数据获取回执表

数据获取回执表的命名格式为：DC_使用方交换节点拼音_ACQUIRE。数据获取回执表包括内建字段（与交换数据表的含义相同）和内容字段两部分，表结构如下：

注：交换桥接一般只需要调用交换平台提供的数据库存储过程接口，而无需直接对该表进行操作。

数据获取回执表结构（DC_使用方交换节点拼音_ACQUIRE）				
字段名	数据类型	允许空	缺省值	含义说明
S_GUID	VARCHAR2(38)	否	sys_guid()	数据记录的主键（物理唯一标识）
S_CREATION_TIME	TIMESTAMP	否	sysimestamp	数据记录在当前节点交换库的物理创建时间
S_LAST_UPDATED	TIMESTAMP	否	sysimestamp	数据记录在提供方交换库的最后更新时间
S_STATUS	VARCHAR2 (1)	否	0	数据记录的状态 0：添加 1：修改

				2: 废置
BEGIN_STAMP	TIMESTAMP	否		在该批次数据记录中，按最后更新时间(S_LAST_UPDATED)截取的最小时间戳
END_STAMP	TIMESTAMP			在该批次数据记录中，按最后更新时间(S_LAST_UPDATED)截取的最大时间戳
EX_TABLE_NAME	VARCHAR2(80)	否		交换数据表的名称
RECORD_COUNT	INT	否		提供方登记该批次提供的记录条数
CHECK_RESULT	INT	否	0	使用方对该批次数据获取情况进行标记（数据获取回执标记），由使用方交换桥接维护该字段。 0: 未获取 1: 获取成功 2: 获取失败
CHECK_TIME	DATE			使用方对该批次数据获取的时间（数据获取时间），由使用方交换桥接维护该字段。
CHECK_COUNT	INT			使用方对该批次数据获取的实际记录条数

表格 5-7 数据获取回执表结构

使用方交换桥接对数据获取回执表的权限规定如下：

使用方交换桥接对数据获取回执表的权限					
对象（表或字段）	SELECT	INSERT	UPDATE	DELETE	权限说明
数据获取回执表	✓	✗	✓	✗	不允许物理删除数据交换表中的记录，但可以通过标记数据状态为“废置”来逻辑删除
S_GUID	✓	✗	✗	✗	插入时取缺省值 sys_guid(), 不允许修改
S_CREATION_TIME	✓	✗	✗	✗	插入时取缺省值 systimestamp, 不允许修

					改
S_LAST_UPDATED	√	×	√	×	插入时取缺省值 systimestamp, 按规定修改
S_STATUS	√	×	√	×	插入时取缺省值 0, 按规定修改
BEGIN_STAMP	√	×	×	×	不允许修改
END_STAMP	√	×	×	×	不允许修改
EX_TABLE_NAME	√	×	×	×	不允许修改
RECORD_COUNT	√	×	×	×	不允许修改
CHECK_RESULT	√	×	√	×	由使用方交换桥接执行获取数据操作后修改
CHECK_TIME	√	×	√	×	由使用方交换桥接执行获取数据操作后修改
CHECK_COUNT	√	×	√	×	由使用方交换桥接执行获取数据操作后修改

表格 5-8 使用方交换桥接对数据获取回执表的权限

5.3.2 数据提供

5.3.2.1 开始数据提供

接口名称	STP_BEGIN_PROVIDE	
功能	向数据提供登记表写入一条记录，通知交换平台，交换桥接已开始一次数据提供操作。	
输入参数		
参数名	类型	说明
P_TABLE_NAME	VARCHAR2（80）	准备写入的交换数据表名
P_RECORD_COUNT	INT	<ul style="list-style-type: none">● 值大于等于 0 时，表示本次数据提供操作准备写入的数据量；● 值为-1 时，表示交换桥接应在完成数据提供时计算并填写本批次提供的数据量；● 其他值无效。
输出参数		
参数名	类型	说明
P_SID	VARCHAR2（38）	批次号。

		如果为空值，则表示对应交换数据表还有其它数据提供的操作未完成，暂时不能开始数据提供操作，应等待上次数据提供操作结束。 否则，则表示可以开始写入数据。
Oracle 调用示例	call EXDB.STP_BEGIN_PROVIDE('EX_GONGSHANG_0001_KYDJ', 100, v_sid); 或 call EXDB.STP_BEGIN_PROVIDE('EX_GONGSHANG_0001_KYDJ', -1, v_sid);	
SQL SERVER 调用示例	exec STP_BEGIN_PROVIDE 'EX_GONGSHANG_01_KYDJ', -1, @v_sid output ;	

5.3.2.2 结束数据提供

接口名称	STP_END_PROVIDE	
功能	更新数据提供登记表记录，通知交换平台，交换桥接已完成一次数据提供操作。	
输入参数		
参数名	类型	说明
P_SID	VARCHAR2（38）	要结束的数据提供操作的批次号
P_RECORD_COUNT	INT	<ul style="list-style-type: none">● 值为-1 时，表示本次提供的数据量以调用 STP_BEGIN_PROVIDE 时提供的参数值为准；● 值大于等于 0 时，表示本次数据提供操作中已写入的数据量（如果交换桥接在调用 STP_BEGIN_PROVIDE 时提供的 P_RECORD_COUNT 参数值大于等于 0，则本参数值视为无效）；● 其他值无效。
输出参数		
参数名	类型	说明
P_RESULT	INT	执行结果状态。 0：表示操作成功。 1：表示操作失败，所指定的批次号无效。此时桥接程序应该中止正常的处理流程，进行程序出错处理。
Oracle 调用示例	call EXDB.STP_END_PROVIDE(v_sid, -1, v_result); 或	

	call EXDB.STP_END_PROVIDE(v_sid, 100, v_result);
SQL SERVER 调用示例	exec STP_END_PROVIDE @v_sid, @v_count, @v_result output ;

5.3.3 数据获取

5.3.3.1 开始数据获取

接口名称	STP_BEGIN_ACQUIRE	
功能	查询数据获取回执表，找出指定主题的未获取的一批数据，获取该批数据的批次信息。	
输入参数		
参数名	类型	说明
P_TABLE_NAME	VARCHAR2（80）	准备获取数据的交换数据表名
输出参数		
参数名	类型	说明
P_SID	VARCHAR2（38）	批次号。如果为空值，则表示目标主题没有未获取的数据；否则，则表示可以开始获取数据。
P_BEGIN_STAMP	TIMESTAMP	待获取数据批次的开始时间戳
P_END_STAMP	TIMESTAMP	待获取数据批次的结束时间戳
P_COUNT	INT	待获取数据批次的记录数
Oracle 调用示例	call EXDB.STP_BEGIN_ACQUIRE('EX_GONGAN_0001_CSDJ', v_sid, v_begin_stamp, v_end_stamp, v_count);	
SQL SERVER 调用示例	exec STP_BEGIN_ACQUIRE 'EX_HZCHUZU_03_DWXX', @v_sid output, @v_begin_stamp output, @v_end_stamp output, @v_count output;	

5.3.3.2 结束数据获取

接口名称	STP_END_ACQUIRE	
功能	更新数据获取回执表中的一条记录，标记该批次的数据已获取完毕。	
输入参数		
参数名	类型	说明
P_SID	VARCHAR2（38）	获取数据的批次号
P_CHECK_COUNT	INT	实际成功获取的记录数
输出参数		

参数名	类型	说明
P_RESULT	INT	执行结果状态。 0: 表示操作成功。 1: 表示操作失败, 所指定的批次号无效。此时桥接程序应该中止正常的处理流程, 进行程序出错处理。
Oracle 调用示例	call EXDB.STP_END_ACQUIRE(v_sid, v_check_count, v_result);	
SQL SERVER 调用示例	exec STP_END_ACQUIRE @v_sid, @v_check_count, @v_result output ;	

6 Web 服务交换

为实现 Web 服务统一身份验证, 保障 Web 服务会话的安全性, 市数据中心交换平台为服务提供方和服务使用方提供了安全令牌服务 (SecurityTokenService, 简称 STS)。市数据中心分别与各成员单位 (使用方或提供方) 建立信任关系, 使用方凭市数据中心颁发的安全令牌可实现对提供方 Web 服务的安全访问, 提供方根据安全令牌可验证使用方的身份并在安全会话中回复服务请求。

6.1 提供 Web 服务

各成员单位通过 Web 服务方式来提供对本单位共享数据的访问, 需要满足两个要求: (一) 所开发的 Web 服务必须满足按本节定义的规范要求; (二) 将本单位的 Web 服务注册到中心 ESB 系统, 并经数据中心审核发布。

使用交换平台的安全令牌服务, 要求服务提供方的 Web 服务策略配置如下:

```
<sp:IssuedToken
sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient">
  <Issuer xmlns="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
    <Address
xmlns="http://www.w3.org/2005/08/addressing">http://dc.gz.gov.cn/ws/STS</Address>
    <Metadata xmlns="http://www.w3.org/2005/08/addressing">
      <Metadata xmlns="http://schemas.xmlsoap.org/ws/2004/09/mex">
```

```

        <wsx:MetadataSection>
            <wsx:MetadataReference>
                <Address
xmlns="http://www.w3.org/2005/08/addressing">http://dc.gz.gov.cn/ws/STS/MEX<
/Address>
            </wsx:MetadataReference>
        </wsx:MetadataSection>
    </Metadata>
</Metadata>
</Issuer>
<sp:RequestSecurityTokenTemplate>
    <t:TokenType
xmlns:t="http://schemas.xmlsoap.org/ws/2005/02/trust">http://docs.oasis-open.org/w
ss/oasis-wss-saml-token-profile-1.1#SAMLV1.1</t:TokenType>
    <t:KeyType
xmlns:t="http://schemas.xmlsoap.org/ws/2005/02/trust">http://schemas.xmlsoap.org/
ws/2005/02/trust/SymmetricKey</t:KeyType>
    <t:KeySize
xmlns:t="http://schemas.xmlsoap.org/ws/2005/02/trust">256</t:KeySize>
</sp:RequestSecurityTokenTemplate>
<wsp:Policy>
    <sp:RequireDerivedKeys/>
    <sp:RequireInternalReference/>
</wsp:Policy>
</sp:IssuedToken>

```

各成员单位提供的 Web 服务交换接口必须遵循的标准规范如下：

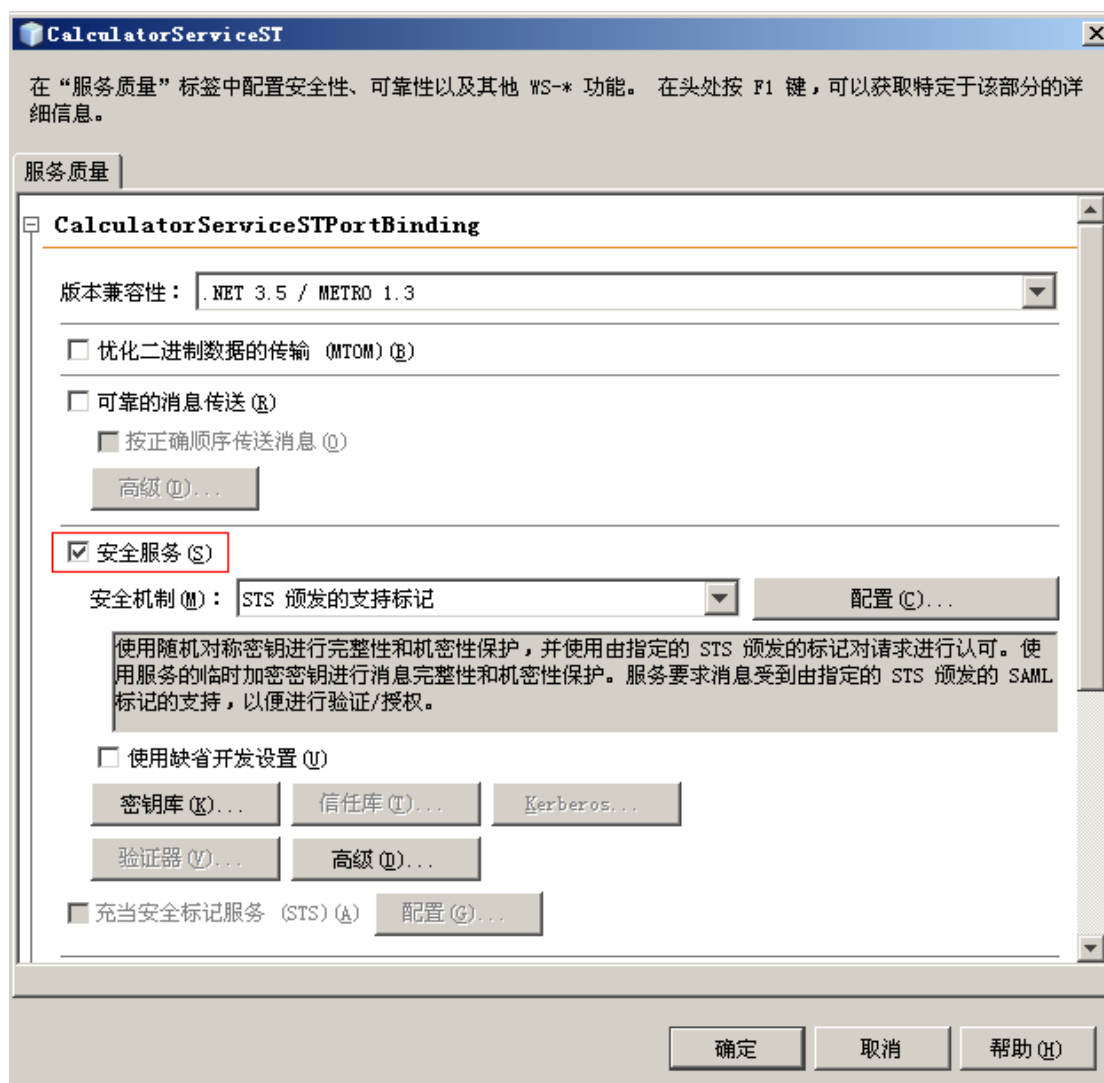
序号	规范名称及标准号	链接
1.	WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1.html
1.	WS-Security	http://docs.oasis-open.org/wss/2004/01/
2.	WS-Policy	http://schemas.xmlsoap.org/ws/2004/09/policy
3.	WS-SecureConversation	http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512
4.	WS-Trust	http://schemas.xmlsoap.org/ws/2005/02/trust

6.1.1 服务配置

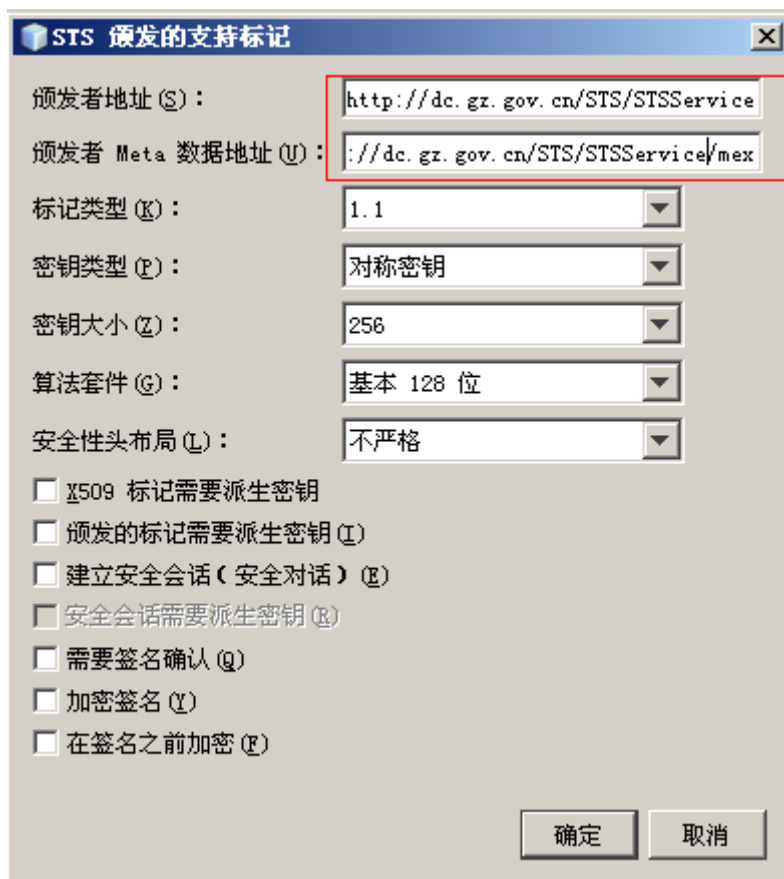
本节以 NetBeans6.5 为例，说明 Java 开发的 Web 服务如何配置。

一、首先打开 NetBeans6.5IDE，在项目中展开并选中需要配置的 Web 服务，从右键菜单中选“编辑 Web 服务属性”，弹出服务属性页，勾选“安全服务”

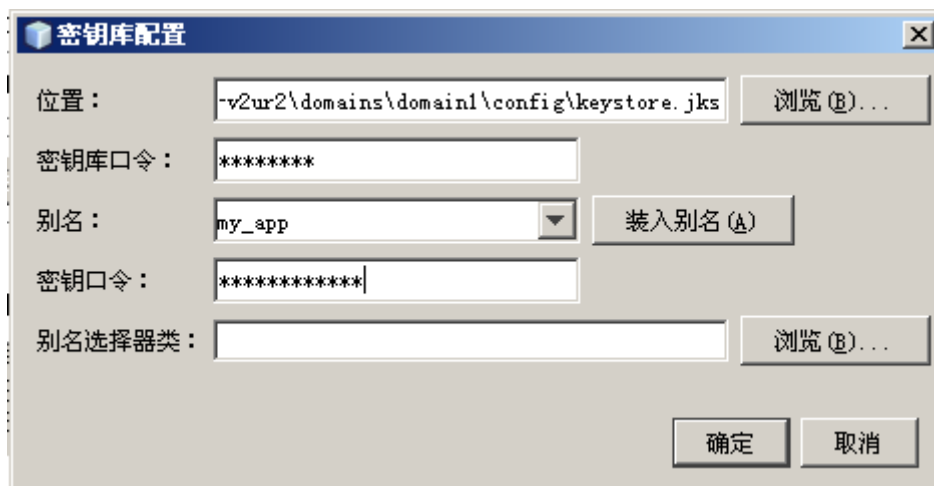
选项，如下图：



二、然后点击“配置”，弹出配置页，填写“颁发者地址”为 <http://dc.gz.gov.cn/STS/STSService>，颁发者 Meta 数据地址为 <http://dc.gz.gov.cn/STS/STSService/mex>，如下图。



三、点击密钥库，配置密钥，配置信息请参看一下节证书安装，配置后如下图所示：



6.1.2 证书安装

使用 Java 开发的 Web 服务，需要把证书导入到 KeyStore 中，导入的方法是通过 JDK 提供的 KeyTool 工具导入，步骤如下：

一、生成密钥对

```
keytool -genkey -alias my_node -validity 365 -keyalg RSA -keysize 1024 -keypass mykeypass -storepass mystorepass -keystore my_keystore
```

其中，my_node 是本单位简称，即接入到数据中心时使用的单位拼简写，如市公安局为“GONGAN”，市地税局为“DISHUI”。

mykeypass 和 mystorepass 是密钥库的密码，在后面的步骤中会使用到。

My_keystore 是 keystore 的名称，可自定义，在后面的步骤中会使用到。

二、导入数据中心根证书：

```
keytool -import -v -trustcacerts -alias dc_ca_root -file ca.crt -storepass mystorepass -keystore my_keystore
```

其中，ca.crt 是数据中心根证书。

三、导入本单位证书：

```
keytool -import -v -alias my_app -file my_app.crt -storepass mystorepass -keystore my_keystore
```

其中，my_app.crt 是为应用程序分派的服务端证书。

6.1.3 服务注册

交换平台基于服务总线（Service Bus）实现对 Web 服务交换的支撑。数据中心、各成员单位都可以作为服务提供方（Service Provider）。所有成员单位提供的服务都必须注册到数据中心 ESB 系统，符合服务目录的规范要求。交换平台提供 Web 服务代理，各服务使用方（Service Consumer）只需面向统一的交换平台调用服务。

6.1.4 如何获取调用方身份

服务提供方可能从请求上下文的 SecurityTokens 获取调用方的安全令牌，示例如下：

采用注入的方式得到 Web 服务上下文对象：

```
@Resource
private WebServiceContext context;
```

从 Web 服务上下文对象中获取调用方的身份标识：

```
/**
 * 从 Web 服务上下文获取调用方的身份信息
```

```

* @param context
* @return
* @throws java.security.cert.CertificateException
*/
private static String getClient(WebServiceContext context) throws CertificateException {
    try {
        Subject subj = SubjectAccessor.getRequesterSubject(context);
        Set<Object> set = subj.getPublicCredentials();
        Element samlAssertion = null;
        for (Object obj : set) {
            if (obj instanceof XMLStreamReader) {
                XMLStreamReader reader = (XMLStreamReader) obj;
                samlAssertion = SAMLUtil.createSAMLAssertion(reader);
                NodeList list = samlAssertion.getElementsByTagName("ds:X509Certificate");
                if (list.getLength() > 0) {
                    // 得到 x.509 证书的内容
                    String certValue = list.item(0).getTextContent();
                    CertificateFactory cf = CertificateFactory.getInstance("X.509");

                    BASE64Decoder decoder = new BASE64Decoder();
                    byte[] buffer = decoder.decodeBuffer(certValue);
                    InputStream fisCert = new ByteArrayInputStream(buffer);

                    // 得到 x.509 证书对象
                    X509Certificate cert = (X509Certificate) cf.generateCertificate(fisCert);
                    // 得到证书主体标识名
                    return cert.getIssuerDN().getName();
                }
            }
        }
    } catch (Exception ex) {
        // 异常处理
    }
    return null;
}

```

上述方法，会返回一个字符串中，它表示调用方的身份信息，格式为：
 CN={部门拼音简写},OU={部门拼音全称},O=dc.gov.gz.cn,C=CN。例如，调用方是市公安局，则上述方法得到的值为：
 CN=GONGAN,OU=GuangZhouShiGongAnJu,O=dc.gov.gz.cn,C=CN。

6.2 使用 Web 服务

服务使用方可从市政务信息资源共享目录检索得到所有可供共享的 Web 服务资源。

使用交换平台的安全令牌服务，要求服务使用方的服务策略配置如下：

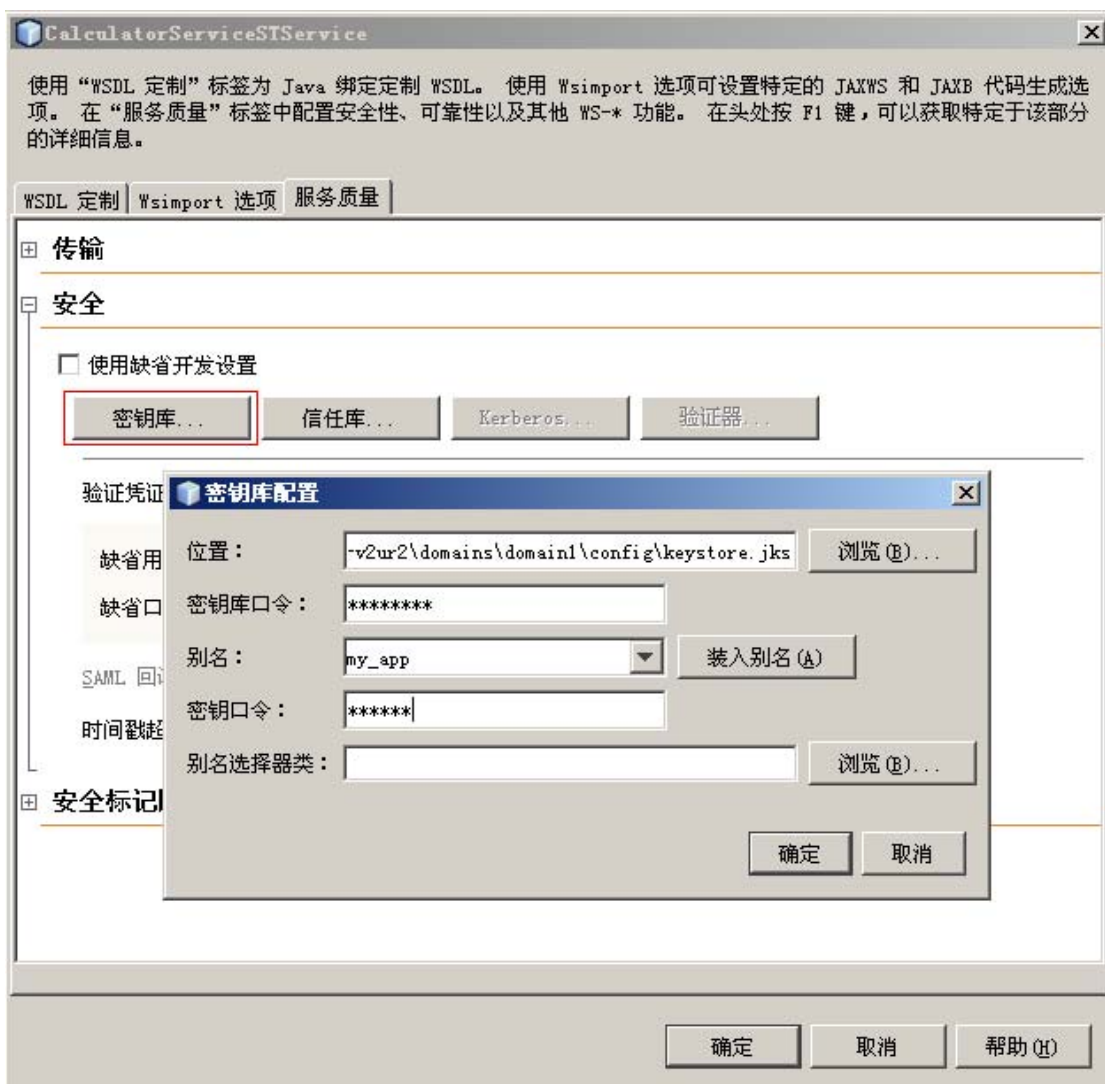
```
<tc:PreconfiguredSTS xmlns:tc="http://schemas.sun.com/ws/2006/05/trust/client"
endpoint="http://dc.gz.gov.cn/ws/STS" metadata="http://dc.gz.gov.cn/ws/STS"/>
```

使用交换平台的 Web 服务代理服务，服务使用方可按照标准的 Web 服务调用规范进行调用。

序号	服务目录	服务地址URL
2.	市数据中心提供的Web服务	http://dc.gz.gov.cn/ws/
3.	各成员单位注册的Web服务代理地址	http://dc.gz.gov.cn/ws/单位拼音简写/ 如地税局： http://dc.gz.gov.cn/ws/DISHUI/

6.2.1 服务配置

打开 NetBeans6.5IDE，在项目中展开并选中需要配置的 Web 服务引用，在“服务质量”页点击“密钥库”配置密钥信息，如下图：



6.2.2 证书安装

所有节点无论是 Web 服务提供方或是 Web 服务使用方，都需要安装两个证书，数据中心概证书和本节点的证书，安装方法参见 6.1.2 节。

6.2.3 如何找到需要的服务

所有通过数据中心交换平台向其他部门公开的 Web 服务，都需要作为 Web 服务资源注册到数据中心共享目录中，通过共享目录统一授权和管理。登录数据中心系统后，在资源目录模块下可以浏览和查询所有已注册的信息资源，其中类型为“服务资源”的即为 Web 服务资源。所有资源必须经过提供方授权才能使用，对未获得授权的资源，可使用部门管理员帐号向提供方发起申请授权，申请授权通过后，再向管理方提出使用申请，申请通过后即可使用。

6.3 交换接口

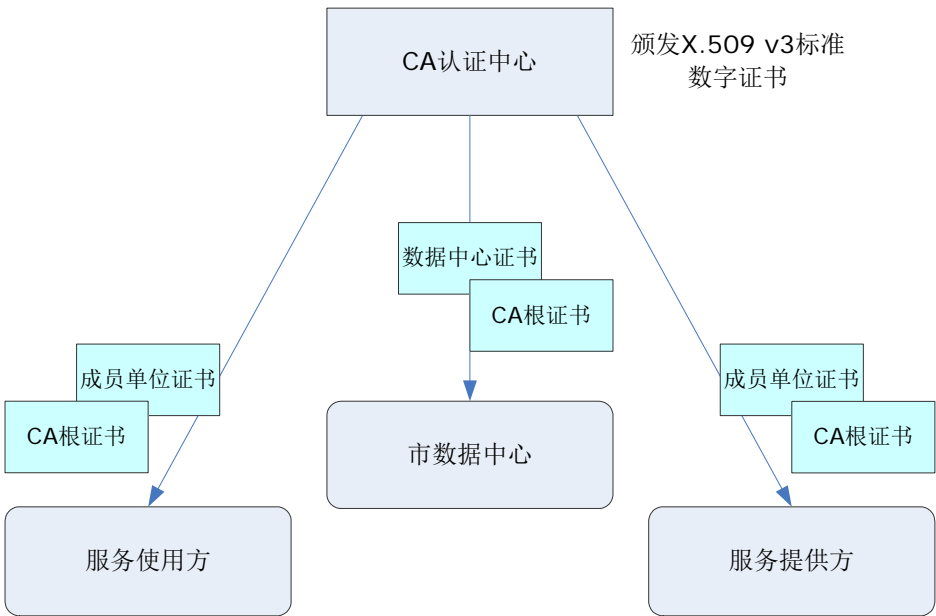
6.3.1 安全令牌服务

安全令牌服务关键在于解决两个问题：一、市数据中心如何与各成员单位建立信任关系；二、Web 服务调用如何实现身份验证并保证会话安全性。交换平台提供的安全令牌服务（STS），通过采用 X.509 v3 标准的数字证书，并遵循 WS-Trust 规范来保证 Web 服务安全性。

下面从信任关系建立和 Web 服务调用两个方面分别进行说明。

（一）信任关系基础

市数据中心和各成员单位的信任关系是基于 PKI 体系建立的。CA 认证中心向市数据中心和各成员单位（服务使用方或服务提供方）颁发 X.509 v3 标准的数字证书。市数据中心和各成员单位在应用环境分别部署从 CA 认证中心申请获取的数字证书以及 CA 根证书。市数据中心和各成员单位都经过统一的 CA 认证，作为相互信任关系的基础。



图表 6-1 基于 PKI 体系的信任关系建立

（二）安全服务会话

基于安全令牌 Web 服务调用分为四个主要步骤：

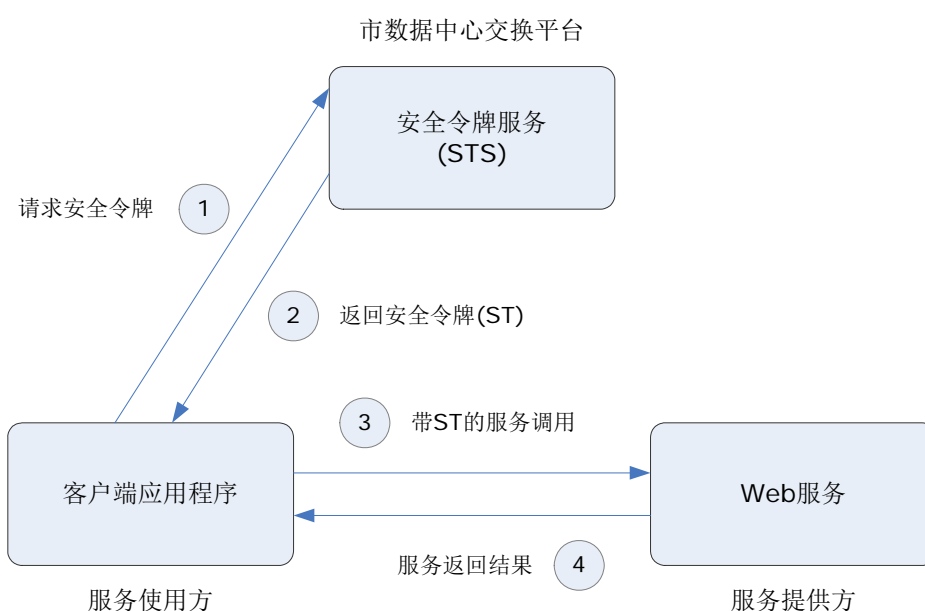
- 1、请求安全令牌。使用方的客户端应用程序根据已配置的安全策略，向交

换平台发送安全令牌请求（RST 消息）。

2、返回安全令牌。交换平台根据安全令牌请求消息，对使用方的身份进行验证，并返回安全令牌（RSTR 消息）。

3、带安全令牌的服务调用。使用方在服务调用中嵌入交换平台返回的安全令牌，向服务提供方发出安全 Web 服务调用请求。

4、Web 服务返回结果。服务提供方根据安全令牌，识别使用方的身份，在安全会话中处理服务请求，向使用方返回结果。



图表 6-2 基于安全令牌 Web 服务调用过程

由于市数据中心、服务使用方和服务提供方具有 CA 颁发的数字证书，并遵循 WS-Trust 规范，因此上述服务调用过程都是安全服务会话。

（三）服务策略配置

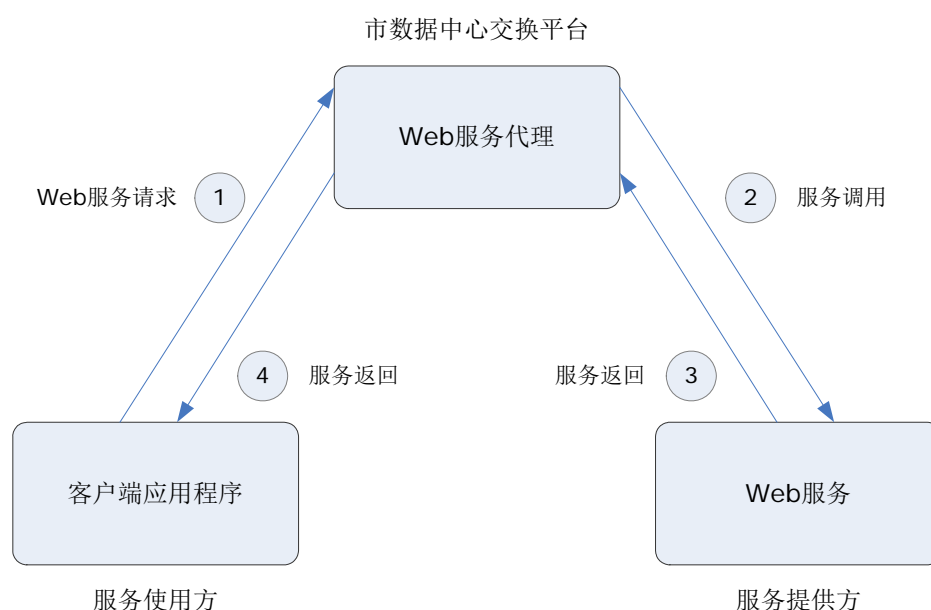
市数据中心交换平台的安全令牌服务（STS）的服务策略配置如下：

```
<tc:STSTConfiguration xmlns:tc="http://schemas.sun.com/ws/2006/05/trust/server"
encryptIssuedKey="true" encryptIssuedToken="false">
  <tc:LifeTime>36000</tc:LifeTime>
  <tc:Contract>gov.datacenter.ws.STS</tc:Contract>
  <tc:Issuer>GZDC</tc:Issuer>
  <tc:ServiceProviders>
    <tc:ServiceProvider endPoint="http://dc.gz.gov.cn/ws/STS">
      <tc:CertAlias>GZDC</tc:CertAlias>
    </tc:ServiceProvider>
  </tc:ServiceProviders>
  <tc:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile1.1#
SAMLV1.1</tc:TokenType>
```

```
</tc:ServiceProvider>
</tc:ServiceProviders>
</tc:STSTConfiguration>
```

6.3.2 Web 服务代理

数据中心中心提供 Web 服务代理,服务提供方可以把服务注册到数据中心,数据中心根据服务定义生成 Web 服务代理程序。服务使用方调用数据中心的服
务代理,由中心服务代理再调用服务提供方的 Web 服务,把结果返回服务使用方。服务代理的运行机制如下图所示:



图表 6-3 Web 服务代理机制

使用服务代理,可在实施上减少对网络环境的要求,只要服务提供方和服务使用方都能与数据中心交换平台的服务器网络连通,即可实现服务调用。

7 实施要求

7.1 配置要求

根据交换数据量的大小和增长规模,估算前置机的性能和容量需求,预留足够处理能力和存储能力,确保前置机稳定、可靠、高效运行。

以下列出的是前置机配置的基本要求：

表格 7-1 前置机配置基本要求

CPU	二颗主频不少于 2.0GHz 的英特尔至强处理器
内存	不少于 2G 的物理内存
硬盘	不少于 300G 的可用硬盘空间 (≥10,000RPM)
网卡	100M 或以上的双网卡
操作系统	Red Flag DC Server 5 或以上版本或 Windows 2003 或以上版本
数据库	Oracle Database 10g Standard Edition One 或以上版本

7.2 网络要求

部门前置机应实现与政务外网的互联，保证与数据中心服务器的物理链路互通。部门前置机的网络端口应针对中心节点服务器和中心维护管理工作站开放数据库访问端口、远程控制端口、Web 服务访问端口、时间同步服务端口，保证与数据中心软件系统的应用链路互通。必须确保中心点节点服务器和中心维护管理工作站能访问前置机的以下端口：

表格 7-2 前置机对中心服务器和维护管理工作站开放的端口列表

序号	端口名称	端口号
1	数据库访问端口	SQL Server: 1433 Oracle: 1521
2	FTP(S)访问端口	FTP: 21 FTP(S): 989、900
3	远程控制端口	Telnet: 22 Windows: 3389
4	Web 服务访问端口 HTTP(S)	HTTP: 80、81 HTTP(S): 443
5	消息中间件	10240~10250 1099 1100

前置机要求能正常访问中心服务器的以下端口：

表格 7-3 前置机需要访问中心服务器的端口列表

序号	端口名称	端口号
1	时间服务	123
2	FTP(S)访问端口	FTP: 21 FTP(S): 989、900
3	Web 服务访问端口 HTTP(S)	HTTP: 80、81 HTTP(S): 443
4	消息中间件	10240~10250 1099 1100

除中心节点服务器、中心维护管理工作站、部门的业务系统服务器、部门的交换桥接所部署的服务器、部门的维护管理工作站外，部门前置机应禁止其他 IP 访问。

7.3 安全措施

以前置机为交换边界，交换节点之间的数据交换安全性主要由交换平台保证，业务应用系统与前置机之间的传输安全性主要由业务部门保证。主要的安全措施要求如下：

（1）保障网络及主机安全。部门应指定前置机管理员，除采取“网络要求”一节的安全措施外，前置机管理员应结合部门的相关安全制度和措施防御非法访问和攻击，做好安全审计和监控，保障前置机网络及主机安全。

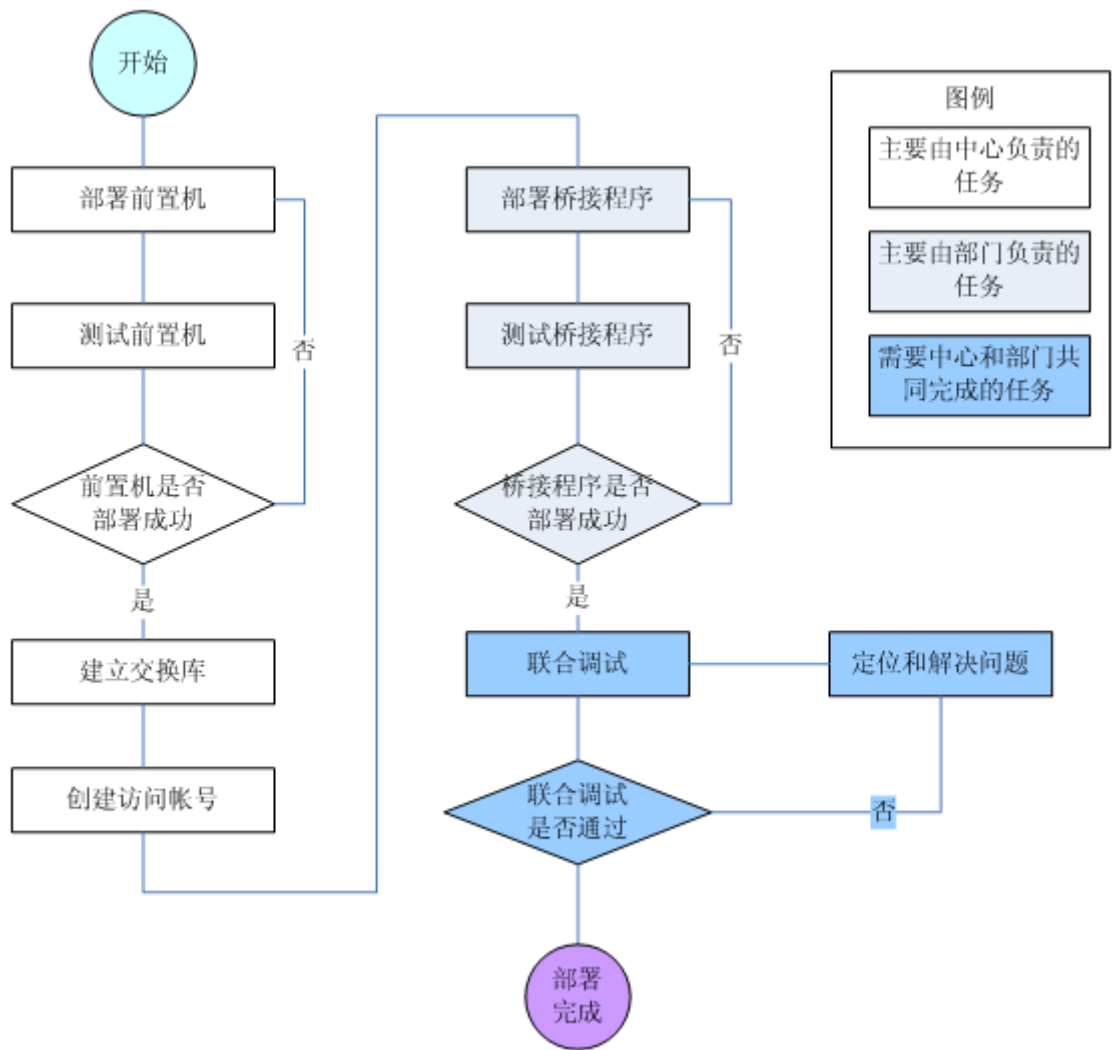
（2）按最小权限原则分配专用帐号。对前置机操作系统、交换数据库、FTP 服务等专用访问账户按最小权限原则分配。

（3）加强数据和应用安全。遵循 WS-Security 和 WS-SecureConversation 等 Web 服务安全规范，确保交换节点之间的业务服务交互的安全性，条件具备的部门应结合 CA 证书对关键数据进行签名、加密等处理。

7.4 安装部署

数据中心负责在部门部署安装前置机。各部门负责部署和实施本部门的业务系统与前置机之间的交换桥接。各部门确保交换桥接能正常运作，按照约定时间定期或实时把增量数据抽取到前置机交换库、按批次获取其他部门的交换

数据。在一个部门接入数据中心时，应该按以下步骤实施：



图表 7-4 交换桥接部署流程

1. 中心实施人员在部门部署前置机，安装并配置必须的系统软件，包括操作系统、数据库、应用服务器等。业务部门协助中心实施人员部署和调试前置机。
2. 前置机部署完毕后，数据中心管理员负责在前置交换库上开设专用的有读写权限的帐号，并提供给业务部门管理员。该访问帐号用于交换桥接向前置交换库提供数据以及从前置交换库获取数据。当需要更新该帐号密码时，业务部门管理员应通过数据中心管理系统进行更新。交换平台提供时间同步服务器，数据中心管理员在安装部署前置机时设置前置机定期与时间服务器进行时钟同步，确保交换平台中各个交换节点的数据库系统时间基本同步。
3. 业务部门部署交换桥接，交换桥接所部署的软硬件环境由业务部门提

供，交换桥接的日常维护和管理都由业务部门负责。特殊情况下，业务部门与交换中心协商一致后，也可以把交换桥接部署在部门前置机上。

4. 交换桥接部署完毕后，中心实施人员协助业务部门进行交换桥接联调，确保数据交换能正常运作。

7.5 运行管理

业务部门前置机是交换平台的组成部分，由数据中心管理员和部门管理共同维护和管理，在日常运作中前置机系统因硬件或软件原因出现问题，无法正常运作时，业务部门管理员应及时向数据中心管理员反映，并协助数据中心管理员定位问题和解决问题。

对业务部门管理员，日常数据交换管理的具体要求如下：

1. 定期查看数据交换情况。部门管理员需要定期以部门管理员帐号登录数据中心管理系统，查看本部门的数据交换情况。检查的频度应与本部门数据交换的频度为依据。

2. 向数据中心管理员反映数据交换异常。部门管理员如发现数据交换有异常，应以电话、邮件等方式及时向数据中心管理员反映情况。数据交换异常可能包括：数据交换量出现剧增或剧减；所有交换数据量都为零；交换的数据中出现大量疑似重复的数据；数据提供登记表中 CHECK_RESULT 为 2（数据记录条数不一致）的情况；数据获取回执表中 CHECK_RESULT 为 2（获取失败）的情况。

3. 协助数据中心管理员定位和解决问题。当确认数据交换出现问题时，部门管理员应协助数据中心管理员定位和解决问题，主要工作包括：检查交换桥接运行是否正常，网络运行是否正常，配合数据中心管理员进行联合调试等。

8 附录

8.1 术语定义

- 交换平台（Exchange Platform）：指广州市电子政务数据中心信息共享交换

平台。

- 交换桥接系统 (Adapter): 由政务部门建设, 政务部门通过交换桥接系统实现业务系统与前置交换系统之间的双向数据交换。
- 交换连接 (Connection): 交换桥接与交换平台的一次通信会话, 一次会话包括多个请求和响应。
- 交换请求 (Request): 交换桥接向交换平台发出交换请求消息。
- 交换响应 (Response): 交换桥接从交换平台收到交换响应消息。
- 交换回执 (Receipt): 提供方交换桥接收到的从提供方端节点、中心节点、使用方端节点、使用方交换桥接产生的交换回执信息, 交换回执信息主要说明何人何时发出回执、是否收到数据、收到数据是否完整等信息。
- 数据集 (DataSet): 指以数据库表 (DataTable) 或文件 (File) 方式存在的交换内容集合。
- 提供数据 (Provide): 提供方交换桥接调用交换接口 API 发送数据 (交换内容) 到提供方端节点的过程。
- 采集数据 (Gather): 中心节点从端节点采集数据的过程。
- 分发数据 (Distribute): 中心节点把数据分发到端节点的过程。
- 获取数据 (Acquire): 使用方交换桥接调用交换接口 API 从使用方端节点获取数据 (交换内容) 的过程。
- 交换方式 (Exchange Mode): 是指根据交换桥接系统提供、获取交换内容的数据存在形态来分类的交换情况, 包括三种交换方式: 消息交换、数据库交换、Web 服务交换。

8.2 规范性引用

凡是不注明日期的引用文件, 其最新版本适用于本部分。

序号	规范名称及标准号	链接
1.	RFC2616: Hypertext Transfer Protocol — HTTP/1.1	http://www.ietf.org/rfc/rfc2616
2.	RFC2965: HTTP State Management	http://www.ietf.org/rfc/rfc2965

	Mechanism	
3.	RFC2818: HTTP Over TLS	http://www.ietf.org/rfc/rfc281
4.	RFC2246: The TLS Protocol Version 1.0	http://www.ietf.org/rfc/rfc2246
5.	RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile	http://www.ietf.org/rfc/rfc2459
6.	The SSL Protocol Version 3.0	http://wp.netscape.com/eng/ssl3/draft302.txt
7.	Extensible Markup Language (XML) 1.0	http://www.w3.org/TR/xml/
8.	XML Schema Part 1: Structures	http://www.w3.org/TR/xmlschema-1
9.	XML Schema Part 2: Datatypes	http://www.w3.org/TR/xmlschema-2
10.	Namespaces in XML	http://www.w3.org/TR/xml-names
11.	XML Information Set	http://www.w3.org/TR/xml-infoset/
12.	Web Services Description Language (WSDL)	http://www.w3.org/TR/wsdl
13.	Simple Object Access Protocol (SOAP)	http://www.w3.org/TR/SOAP/
14.	Universal Description, Discovery and Integration (UDDI)	http://uddi.org/pubs/uddi_v3.htm
15.	SOAP Message Transmission Optimization Mechanism	http://www.w3.org/TR/soap12-mtom/
16.	XML-binary Optimized Packaging	http://www.w3.org/TR/xop10/
17.	WS-I Basic Profile 1.1	http://www.ws-i.org/Profiles/BasicProfile-1.1.html
18.	OASIS WS-Security	http://docs.oasis-open.org/wss/2004/01/
19.	OASIS WS-SecureConversation	http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512
20.	Web Services Addressing	http://www.w3.org/Submission/ws-addressing/

	(WS-Addressing)	
21.	Web Services Reliable Messaging (WS-ReliableMessaging)	http://docs.oasis-open.org/ws-rx/wsrn/200608
22.	Java™ Platform Enterprise Edition Specification, v1.5	http://java.sun.com/javae/
23.	JSR 105: XML Digital Signature APIs	http://jcp.org/en/jsr/detail?id=105
24.	JSR 208: Java™ Business Integration (JBI)	http://jcp.org/en/jsr/detail?id=208

8.3 交换节点编码

(1) 委办局节点

序号	政务部门简称	交换节点编码
1	市安监局	ANJIAN
2	市财政局	CAIZHENG
3	市残联	CANLIAN
4	市城管委	CHENGGUANWEI
5	市档案局	DAAN
6	市地税局	DISHUI
7	市发改委	FAGAIWEI
8	市防空办	FANGKONGBAN
9	市港务局	GANGWU
10	市公安局	GONGAN
11	市住房公积金管理中心	GONGJIJIN
12	市工商局	GONGSHANG
13	市规划局	GUIHUA
14	市国税局	GUOSHUI
15	市国土房管局	GUOTU
16	市环保局	HUANBAO

17	黄埔海关	HUANGPUHAIGUAN
18	市监察局	JIANCHA
19	市建委	JIANWEI
20	市交委	JIAOWEI
21	市教育局	JIAOYU
22	市经贸委	JINGMAOWEI
23	市科技局	KEJIJU
24	市林业和园林局	LINYE
25	市民政局	MINZHENG
26	市农业局	MONGYE
27	市气象局	QIXIANG
28	市人口计生局	RENKOUJISHENG
29	市人社局	RENSHE
30	市水务局	SHUIWU
31	市司法局	SIFA
32	市体育局	TIYU
33	市统计局	TONGJI
34	市外经贸局	WAIJINGMAO
35	市卫生局	WEISHENG
36	市物价局	WUJIA
37	市无线电管理办	WUXIANDIAN
38	市科信局	KEXIN
39	市药监局	YAOJIAN
40	市园林局	YUANLIN
41	市政务办	ZHENGWUBAN
42	市质监局	ZHIJIAN
43	市知识产权局	ZHISHICHANQUAN
44	市中院	ZHONGYUAN

(2) 区（县级市）节点

序号	政务部门简称	交换节点编码
1	白云区	BAIYUN
2	从化市	CONGHUA
3	海珠区	HAIZHU
4	黄埔区	HUANGPU
5	花都区	HUADU
6	荔湾区	LIWAN
7	萝岗区	LUOGANG
8	南沙区	NANSHA
9	番禺区	PANYU
10	天河区	TIANHE
11	越秀区	YUEXIU
12	增城市	ZENGCHENG

(3) 中心及其他节点

序号	政务部门简称	交换节点编码
1	市数据中心	DATACENTER
2	企业信用	QIYEXINYONG
3	市社保卡	SHEBAOKA
4	省信息中心	XINXIZHONGXIN
5	公用前置机	GONGYONG