

物流信息交换基础网络
共建指导性文件
第 2 部分：交换接入

Logistics information exchange basic network

common construction guidance documents

Part2: exchange access

(Version 1.0)

目 次

1 范围.....	2
2 规范性引用文件.....	2
3 概述.....	2
4 交换认证服务.....	2
4.1 安全认证服务.....	2
4.2 退出授权服务.....	3
5 数据交换服务.....	3
5.1 数据发送服务(send).....	3
5.2 数据接收服务(receive).....	4
5.3 接收确认服务(confirmReceive).....	5
5.4 业务代理服务(proxySend).....	6
6 数据类型定义.....	6
6.1 简单数据类型.....	6
6.2 复杂数据类型.....	7
7 数据格式.....	8
7.1 交换数据包.....	8
7.2 数据包单元.....	9
7.3 国标业务数据.....	10
7.4 数据文件.....	11
7.5 简单数据.....	12
7.6 CSV 数据文件.....	12
8 传输错误代码.....	12
附录 A (规范性附录) WSDL 和 XML Schema 文件.....	15
A.1 交换同步传输服务接口定义(ExchangeTransportService.wsdl).....	15
A.2 交换安全服务接口定义(exchange-security.wsdl).....	19
A.3 业务代理服务接口定义(ExchangeBusinessProxyService.wsdl).....	20
A.4 交换同步传输服务 XML 消息定义(ExchangeTransportSystem.xsd).....	22
A.5 业务代理服务 XML 消息定义(ExchangeBusinessProxy-data-format.xsd).....	26
A.6 交换安全服务 XML 消息定义(exchange-security.xsd).....	27
A.7 基本数据类型 XML 消息定义(coreDataTypes.xsd).....	28
A.8 数据格式 XML 消息定义(exchange-data-format.xsd).....	29
A.9 业务数据表示格式 XML 消息定义(egs-org-cn-businessdata.xsd).....	30
附录 B (资料性附录) API 接口说明.....	33
B.1 Cuteinfo 客户端 API 接口说明(.Net 版).....	33
B.2 Cuteinfo 客户端 API 接口说明(Java 版).....	33
附录 C (资料性附录) 交换\应用服务器列表.....	34

1 范围

本部分规定了物流信息交换基础网络的安全和交换传输的技术要求。

本标准适用于物流行业相关的公共信息平台、电子数据交换与信息管理系统等的设计与开发应用。

2 规范性引用文件

《GB/T21062 政务信息资源交换体系 第1部分 总体框架》

3 概述

本文件是数据交换的技术规范，规定了企业接入数据交换中心所遵循的技术要求和约束，包括交换认证、交换传输、数据类型、数据格式和错误代码等内容。

交换基础网络采用 Web service技术，遵循的技术规范有：WSDL、SOAP、WS-Security等，并且提供一整套API，使物流通用软件或外部系统快速接入交换基础网络。

4 交换认证服务

该服务是企业接入交换中心的一种安全保障机制，为交换双方提供身份安全认证和交换合法性认证，包括安全认证操作和退出授权操作。

企业在进行数据交换前，首先调用身份安全认证服务，该安全认证服务返回给企业一个令牌标识。

企业进行数据交换时，物流通用软件或外部系统应在请求消息头中加入认证时获取的令牌信息。

令牌信息在消息头中通过ws-security格式来描述，示例如下：

```
<wsse:Security>
<wsse:UsernameToken>
<wsse:Username>userName</wsse:Username>
<wsse:UserTokenID>pwdvalue</wsse: UserTokenID >
</wsse:UsernameToken>
</wsse:Security>
```

企业进行用户认证时，用户名和密码是企业通过交换基础网络根服务器上申请的物流交换代码和填写的密码。

申请物流交换代码见《交通物流信息交换基础网络技术规范 第1部分：物流交换代码》

4.1 安全认证服务

4.1.1 概述

安全认证服务是一种基于web的服务，从输入消息和输出消息两部分服务属性进行约定，完成对用户身份的安全认证。

4.1.2 请求消息

用户安全认证消息以XML方式描述了应进行认证的用户信息。认证请求消息包括用户名和密码

属性名	XML元素名	数据类型	约束	说明
账号名称	Username	属性/String	1..1	

账号密码	Password	属性/String	1..1	
------	----------	-----------	------	--

4.1.3 响应消息

用户认证操作将用户的令牌标识以XML格式封装返回给用户。如果认证失败，返回用户错误代码和错误原因。

属性名	XML元素名	数据类型	约束	默认值
用户令牌标识	UserTokenID	属性/String	0..1	
认证失败	GenericFaul	元素/复杂数据类型	0..1	

➤ 认证失败（GenericFaul）：是复杂数据类型，由数据类型进行完整描述，见本部分中 6.2.3 通用异常数据类型。

4.2 退出授权服务

4.2.1 概述

退出授权服务是一种基于web的服务，从输入消息和输出消息两部分服务属性进行约定，完成对用户退出登录的功能。

4.2.2 请求消息

用户注销登陆请求消息以XML方式描述了用户名称和用户令牌标识。

属性名	XML元素名	数据类型	约束	说明
用户名	UserName	属性/String	1..1	
用户令牌标识	UserTokenID	属性/String	1..1	

4.2.3 响应消息

用户退出授权服务请求以XML方式返回用户注销操作的结果。

属性名	XML元素名	数据类型	约束	说明
退出结果	LogoutResult	属性/boolean	1..1	

5 数据交换服务

支持分布或异构应用系统，通过交换平台进行信息交换的、面向SOA（面向服务架构）的一种信息系统服务模式，目前提供发送、接收、接收确认、业务代理服务。

4.3 数据发送服务(send)

5.1.1 概述

数据发送服务是一种基于web的服务，从输入消息和输出消息两部分服务属性进行约定，为企业提供了发送数据的功能，采用异步方式的交换传输技术规范。

5.1.2 请求消息

存储输入数据消息以XML方式描述了通用的存储请求。请求消息由数据目标地址、交换事件是否

同步发送属性组成，请求消息主要由下面的属性组成：

属性名	XML元素名	数据类型	约束	默认值
目标地址	ToAddress	元素/String	1..n	
交换事件	ExchangeEvent	交换事件数据类型	1..n	

其中：

- 目标地址（ToAddress）为：数据事件发送的目的地址。这里地址类型可以单个用户地址，也可以群组地址；
- 交换事件数据类型：需要发送的数据进行封装，由交换事件数据类型进行完整描述，见本部分 6.2.1 交换事件数据类型。

5.1.3 响应消息

响应消息是将操作结果封装。响应消息是发送结果类型。由执行状态，发送失败集组成。

属性名	XML元素名	数据类型	约束	默认值
发送结果	sendResult	boolean	1..1	
失败信息	GenericFault	复杂数据类型	0..n	

其中：

- 发送结果：整数类型。表示发送操作是否执行成功；
- 失败信息：执行发送失败的复杂类型。包括错误代码和错误原因，见本部分中 6.2.3 通用异常数据类型。

4.4 数据接收服务(receive)

5.2.1 概述

数据接收服务是一种基于web的服务，从输入消息和输出消息两部分服务属性进行约定，为企业提供了接收数据的功能，采用异步方式的交换传输技术规范。

5.2.2 请求消息

接收请求消息以XML方式描述了操作的请求数据。消息内容包括通道名称，超时时间和是否阻塞。

属性名	XML元素名	数据类型	约束	默认值
接收地址	Address	元素/String	1..1	
事件数量	receiveNumber	元素/int	1..1	1
超时时间	timeout	元素/long	1..1	3000ms
是否阻塞	isBlocked	元素/boolean	1..1	false

其中，

- 接收地址：请求消息中可以指定从哪个地址获取交换事件；
- 事件数量：接收事件的最大数目。服务端的数目如果小于最大数据，则返回所有事件；
- 超时时间：指定接收消息的超时时间。在指定的时间内没有接收到消息则操作返回；
- 是否阻塞：是否阻塞操作的执行直到有消息收到。默认为 false，收不到消息立即返回。

5.2.3 响应消息

接收操作的响应消息封装了接收操作执行的返回结果，响应消息包括的属性如下：

属性名	XML元素名	数据类型	约束	默认值
未处理事件数	TotalEventCount	String	1..1	0
接收事件	ReceiveExchangeEvent	交换事件数据类型	0..n	
异常信息	GenericFault	通用异常类型	0..1	

其中，

- 未处理事件数：指当前地址未被处理的事件数；
- 接收事件：交换事件类型封装了交换的数据事件，具体参照交换事件数据类型定义，见本部分 6.2.1 交换事件数据类型；
- 异常信息：在捕获输出数据时所发生的所有异常信息。见 6.2.3 通用异常数据类型。

4.5 接收确认服务(confirmReceive)

5.3.1 概述

接收确认服务是一种基于web的服务，从输入消息和输出消息两部分服务属性进行约定，为企业提供了确认消息的正确接收，并通知交换中心对消息进行清除处理，采用异步方式的交换传输技术规范。

5.3.2 请求消息

属性名	XML元素名	数据类型	约束	说明
事件标识	EventID	属性/string	1..n	
接收地址	Address	属性/string	1..1	
是否正确接收	isSuccessReceive	属性/boolean	1..1	
是否删除消息	isDeleteMsg	属性/boolean	0..1	

其中，

- 事件标识：交换事件的唯一标识；
- 接收地址：接收地址名称；
- 是否正确接收：标识物流通用软件和重要物流及相关信息系统是否正确接收到消息；
- 是否删除消息：通知交换中心是否删除此消息。

5.3.3 响应消息

接收操作的响应消息封装了接收操作执行的返回结果，响应消息包括的属性如下：

属性名	XML元素名	数据类型	约束	说明
确认结果	confirmResult	属性/boolean	0..n	
异常信息	GenericFault	通用异常类型	0..1	

其中：

- 确认结果：标识确认是否成功；
- 异常信息：异常信息的详细内容。见本部分 6.2.3。

4.6 业务代理服务(proxySend)

5.4.1 概述

此服务主要是为物流通用软件提供跨中心的非交换数据的信息交互服务。该服务是一种基于web的服务，从输入消息和输出消息两部分服务属性进行约定。

业务代理服务是一种基于web的信息交互服务，从输入消息和输出消息两部分服务属性进行约定。物流通用软件根据请求的业务服务类型，由交换中心通过业务代理自行组合相关信息的操作和返回，为调用方提供相应的信息服务。

采用同步方式的交换传输技术规范。

5.4.2 请求消息

属性名	XML元素名	数据类型	约束	默认值
事件标识	ProxyID	属性/String	1..1	
操作类型	ActionType	属性/String	1..1	
消息正文	BusinessXML	子元素/base64Binary	1..1	

其中，

- 事件标识 (ProxyID)：唯一标识一类业务代理；
- 操作类型 (ActionType)：唯一标识交换活动的事件活动类型；
- 消息正文 (BusinessXML)：以 base64Binary 格式标识具体代理的业务相关内容。

5.4.3 响应消息

属性名	XML元素名	数据类型	约束	说明
服务代理结果	Result	属性/boolean	1..1	
服务代理内容	ResultXML	属性/ base64Binary	0..1	
错误消息	GenericFault	元素/通用异常类型	0..n	

其中：

- 服务代理结果：标识代理服务调用是否成功；
- 服务代理内容：调用代理服务所获取的内容；
- 错误消息：异常消息的详细内容。

6 数据类型定义

6.1 简单数据类型

6.1.1 字符型

表示为：string

6.1.2 整数型

表示为：int

6.1.3 日期型

表示为: date

6.1.4 数据变化类型

在请求增量数据或者存储数据的时候都需要指定数据变化的类型, 数据变化的类型包括: 新增、修改和删除三种, 数据的变化类型被定义为一个简单类型, 它不包括任何属性和子元素, 但是对其取值范围进行了限定, 分别是: Insert, Update, Delete。

6.2 复杂数据类型

6.2.1 交换事件数据类型

属性名	XML元素名	数据类型	约束	默认值
事件标识	EventID	String	1..1	
操作类型	ActionType	String	1..1	
交换数据包	ExchangeDataPackage	交换数据包数据类型	1..1	
扩展属性	ExtendAttribute	anyType	0..1	

➤ 事件标识

事件的唯一标识。此标识要求在时间和空间上全局唯一。

➤ 操作类型

交换操作的类型。可以是转发数据或者调用服务。

➤ 交换数据包数据类型

详细信息参见本部分 7.1 交换数据包。

➤ 扩展属性

物流通用软件可自定义任何类型进行属性设置。

6.2.2 接收事件数据类型

接收事件类型基础交换事件类型, 在交换事件类型的基础上增加了接收时间属性。

属性名	XML元素名	数据类型	约束	默认值
事件接收时间	ReceivedTime	String	1..1	

6.2.3 通用异常数据类型

通用异常数据类型规定了一般的异常表示方法, 本规范规定通用异常包括异常编码和异常错误信息, 它的属性定义如下:

属性名	XML元素名	数据类型	约束	默认值
异常编码	Code	string	1..1	
异常信息	ErrorMessage	string	1..1	

➤ 异常编码

异常的错误代码。本规范不限定任何错误代码。

➤ 异常信息

异常发生的原因的详细信息。

7 数据格式

数据格式定义了系统间进行数据交换时基于消息机制的数据表示格式。

本部分在使用国标已经定义的数据接口模型的基础上，进行补充、细化，以增强表达大量数据集合、非结构化数据文件以及多种格式数据的能力，同时扩充了交换的数据的元数据信息。

业务接口报文格式采用XML格式，同时平台也支持文本文件、Excel文件等数据格式文件，用于完成各种异构数据的采集、转换、加载和发布功能。

7.1 交换数据包

7.1.1 概述

交换数据包是数据交换中具有完整业务含义的政务信息资源集合单位。完整业务含义是从业务角度来定义的，通常由若干种类型的数据包单元组成，如某次交换的企业基本信息、企业年检信息以及企业处罚信息。

交换数据包由下面的属性组成：

属性名	XML元素名	类型	是否必须	最大出现次数
交换数据报ID	packageID	属性/string	是	1
交易ID	transactionID	属性/string	是	1
创建时间	startTime	属性/datetime	是	1
失效时间	endTime	属性/datetime	否	1
标题	title	属性/string	是	1
数据包单元	ExchangeDataPackageUnit	子元素/ ExchangeDataPackageUnit	是	不限

7.1.2 属性描述

7.1.3 交换数据包 ID

唯一标识一个数据交换包的ID，要求在时间上和空间上唯一。

7.1.4 交易 ID

唯一标识一次数据交换交易的ID，要求在时间上和空间上唯一。交易ID标识了交换数据包之间的业务关联性，体现了业务完整性。

7.1.5 创建时间

交换数据包创建的时间。

7.1.6 失效时间

交换数据包数据有效的时间。不指定该时间表示永久有效。

7.1.7 标题

交换数据包的描述性名称。

7.1.8 数据包单元

特定结构的数据集合。交换数据包中的数据可以包含多种格式的数据，这些数据的组织方式以数据包为单位确保相同格式的数据集中的存放在一起。

7.2 数据包单元

7.1.9 概述

数据包单元是特定结构的数据集合，由描述信息和数据包组成。通过分组标识，可将不同的数据包组织在一起，在相同组中的数据包可以拥有相同的或不同的数据结构。在数据包中兼容了国标中定义的数据接口模型，同时也针对数据文件定义了相应的表示方法。

数据包单元包括的属性如下表：

属性名	XML元素名	类型	是否必须	最大出现次数
分组标识	groupID	属性/string	是	1
分组大小	groupSize	属性/int	是	1
组内序号	sequenceInGroup	属性/int	是	1
包单元ID	unitID	属性/string	是	1
数据来源	source	属性/string	是	1
实体标识名	entityIDName	属性/string	是	1
实体显示名	entityDispName	属性/string	否	1
创建时间	createDate	属性/datetime	是	1
失效时间	expireDate	属性/datetime	否	1
国标业务数据	BusinessData	子元素 /egsbd:BusinessData	否	1
数据文件	DataFile	子元素/DataFile	否	1

7.1.10 属性描述

7.1.11 分组标识

在一个交换数据包中唯一标识一个分组的标识符。

7.1.12 分组大小

表示组成员的数量。数据包单元在组织形式上以组为单位进行存放，相同组的数据包单元应当连续存放。

7.1.13 组内序号

数据包单元作为组成员的顺序号。

7.1.14 包单元 ID

数据包单元的唯一标识符。该标识符要求在一个交换数据包中不重复。

7.1.15 数据来源

表示数据包单元的数据来源的描述性信息。

7.1.16 实体标识名

确定一个数据实体的唯一逻辑标识符。对于没有预定义的数据实体，以unkown字符串来表示

7.1.17 实体显示名

数据实体用于显示的名称，要求不同结构的数据实体必须采用不同的显示名称。

7.1.18 创建时间

数据包单元的创建时间。

7.1.19 失效时间

数据包单元数据失效的时间。不指定该属性表示永不失效。

7.1.20 国标业务数据

基于国标交换体系的数据接口规范定义数据的表示。本规范通过引入的该结构兼容了国标中对交换数据的格式要求。

7.1.21 数据文件

数据文件是指数据包单元中的数据是以文件方式存在。结构化和非结构化数据都可以采用数据文件作为数据包中数据的组织方式。

7.1.22 简单数据集

以数据名称和数据值组成的数据值对方式进行的简单数据的集合。

7.3 国标业务数据

7.1.23 概述

国标业务数据是指国标交换体系中定义的数据接口模型。该模型用于在不同系统间进行信息交换时封装交换信息内容，可支持结构化数据、非结构化数据的封装。

数据接口模型由数据结构、数据集、附件集组成。数据结构是可选元素，元素名称是DataStructure，用来描述交换信息内容的结构信息。数据集是可选元素，元素名称是DataSet，用来封装结构化数据。附件集是可选元素，元素名称是Attachments，用来表示非结构化数据。数据集和附件集可以同时出现，也可以单独出现。

详细的定义参见《GB/T21062 政务信息资源交换体系 第3部分：数据接口规范》。

7.1.24 属性描述

略。参见《GB/T21062 政务信息资源交换体系 第3部分:数据接口规范》。

7.4 数据文件

7.1.25 概述

数据文件是数据包单元中封装交换数据包的一种方式。这种方式封装的结果是将数据（包括结构化数据）以文件的方式表示，因此称为“数据文件”。

非结构化数据本来就是以文件方式存在，此时封装仅指提供一些描述信息，并不处理文件内容。结构化数据一种是采用国标业务数据的表示方式，另一种是将其序列化为某种格式的数据文件，然后再进行交换。

数据文件的属性有：

属性名	XML元素名	类型	是否必须	最大出现次数
文件标识	dataFileID	属性/string	是	1
文件名	fileName	属性/string	是	1
文件格式	dataFileFormat	属性/string	是	1
嵌入式内容	Base64EncodedData	子元素/base64Binary	是	1
数据附件	AttachmentData	子元素/binary	是	1

7.1.26 属性描述

7.1.27 文件标识

在一个交换数据包中唯一标识一个数据文件的标识符。

7.1.28 文件名

数据文件的文件名称。此文件名称不携带任何路径信息。

7.1.29 文件格式

文件所采用的格式。对于非结构化数据文件可以参考MIME中规定的类型。对于结构化数据文件可以在交换中协商确定。

7.1.30 嵌入式内容

以Base64编码方式将数据文件内容直接存放在此元素下。对于超过一定大小的数据文件不建议使用此种方式，而应使用消息附件方式。

7.1.31 数据附件

文件以消息附件的方式进行引用。

7.5 简单数据

7.1.32 概述

简单数据类型以数据名称和数据值对的方式来描述数据。数据值的数据类型只能是字符串类型。

属性名	XML元素名	类型	是否必须	最大出现次数
名称	FieldName	属性/string	是	1
值	FieldValue	属性/string	是	1

7.1.33 属性描述

7.1.34 数据名称

数据的名称。

7.1.35 数据值

数据的具体值。只能为字符串类型。

7.6 CSV 数据文件

7.1.36 概述

CSV即Comma Separate Values，这种文件格式经常用来作为不同程序之间的数据交互的格式。当交换数据包中的交换数据文件格式为CSV时，CSV以ZIP压缩后的格式作为数据包中的文件进行传输。

7.1.37 格式描述

- ✓ 每条记录占一行
- ✓ 以逗号为分隔符
- ✓ 逗号前后的空格会被忽略
- ✓ 字段中包含有逗号，该字段必须用双引号括起来
- ✓ 字段中包含有换行符，该字段必须用双引号括起来
- ✓ 字段前后包含有空格，该字段必须用双引号括起来
- ✓ 字段中的双引号用两个双引号表示
- ✓ 字段中如果有双引号，该字段必须用双引号括起来
- ✓ 第一条记录，可以是字段名。

如果某自端的类型为二进制类型数据，则在CSV文件中描述二进制类型数据文件在zip包中的相对路径，二进制文件在和CSV文件在zip包中一起传输。

8 传输错误代码

物流通用软件或外部系统调用传输服务时，可能会在响应消息里收到异常代码，以及物流通用软件或外部系统应如何处理这些异常代码，错误代码见下表：

异常编码 (Code)	异常信息 (ErrorMessage)	描述	处理方式
0	服务端运行异常		
1003	重复事件错误	重复事件错误，表明事件ID已经被使用	

2000	无效交换用户或密码	无效用户或密码，表明认证信息错误	
2001	认证令牌超时	令牌超时失效	
2002	无效认证令牌		
2003	消息目的地被屏蔽		
3001	无效消息标识		
5002	无效的组或用户	无效的组或用户标识，表明使用的交换用户或者组是非法名称	
5003	无效消息目的地	无效的目标地址，表明目标地址不存在	
6001	无法参与交换		
6002	无接收功能		
9002	消息身份验证失败		
9003	消息目的地验证失败		
9004	消息防火墙验证失败		
9005	消息存储失败		
9006	消息标识验证失败		
9007	消息接收失败		
9008	消息接收确认失败		
7001	业务代理失败		
PY2000	the wldm is null	传过来的物流交换代码为空	
PY2001	the user *** password is not right	物流交换代码的密码错误	
PY2002	the user *** is not exit	激活的物流交换代码不存在	
PY2003	the user is not active	激活失败，用户信息更新失败	
PY2004	*** is not a active group	需要添加的群组无效	
PY2005	apply userservice exception	开通用户服务异常	
PY2006	***:***	物流代码所在服务器与群组所在服务器不是同一个（：前面为物流交换代码所在的服务器名称，：后面为群组所在服务器名称）	

附录A
(规范性附录)
WSDL和XML Schema文件

WSDL 是对交换服务接口的定义，包括3个文件，Schema是对交换服务接口请求/响应消息的XML Schema定义，包括6个文件，定义如下：

序号	文件名	文件描述
1	ExchangeTransPortService.wsdl	交换服务的Web Service接口定义，包括发送、接收、接收确认三个。
2	exchange-security.wsdl	交换安全服务的Web Service接口定义
3	ExchangeBusinessProxyService.wsdl	业务代理服务的Web Service接口定义
1	ExchangeTransportSystem.xsd	交换服务接口的请求/响应消息的XML Schema定义，包括发送、接收、接收确认三个。
2	exchange-security.xsd	交换安全服务接口的请求/响应消息的XML Schema定义
3	coreDataTypes.xsd	描述了基础的数据表示格式，是exchange-data-format.xsd数据格式定义文件和egs-org-cn-businessdata.xsd数据格式定义的基础
4	exchange-data-format.xsd	定义了系统间基于消息机制进行数据交换时数据的表示格式
5	egs-org-cn-businessdata.xsd	描述了一个统一的业务数据表示格式
6	ExchangeBusinessProxy-data-format.xsd	业务代理服务接口的请求/响应消息的XML Schema定义

A.1 交换同步传输服务接口定义 (ExchangeTransPortService.wsdl)

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://service.egs.org.cn/exchange/exchangetransport"
  xmlns:tra="http://schema.egs.org.cn/exchange/exchangetransport"
  xmlns:ns="http://schema.egs.org.cn/exchange/edf"
  xmlns:ns1="http://schema.egs.org.cn/exchange/cdt"
  xmlns:ns2="http://schemas.egs.org.cn/businessdata"
```

```

xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm/policy"
xmlns:wssu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://service.egs.org.cn/exchange/exchangetransport">
<wsdl:types>
  <xs:schema>
    <xs:import
      namespace="http://schema.egs.org.cn/exchange/exchangetransport"
      schemaLocation="schema/ExchangeTransportSystem.xsd" />
  </xs:schema>
</wsdl:types>
<message name="sendRequest">
  <part name="partSendRequest" element="tra:SendRequest" />
</message>
<message name="sendResponse">
  <part name="partSendResponse" element="tra:SendResponse" />
</message>
<message name="receiveRequest">
  <part name="partReceiveRequest" element="tra:ReceiveRequest" />
</message>
<message name="receiveResponse">
  <part name="receiveResult" element="tra:ReceiveResponse" />
</message>
<message name="confirmReceiveRequest">
  <part name="partConfirmReceiveRequest"
    element="tra:ConfirmReceiveRequest" />
</message>
<message name="confirmReceiveResponse">
  <part name="partConfirmReceiveResponse"
    element="tra:ConfirmReceiveResponse" />
</message>
  <message name="groupReceiveRequest">
    <part name="partGroupReceiveRequest" element="tra:GroupReceiveRequest"></part>
  </message>
  <message name="groupReceiveResponse">
    <part name="partGroupReceiveResponse" element="tra:GroupReceiveResponse"></part>
  </message>
  <wsdl:message name="confirmGroupReceiveRequest">
    <wsdl:part
      name="partConfirmGroupReceiveRequest"
      element="tra:ConfirmGroupReceiveRequest"></wsdl:part>
  </wsdl:message>
  <wsdl:message name="confirmGroupReceiveResponse">

```

```

        <wsdl:part name="partConfirmGroupReceiveResponse"
element="tra:ConfirmGroupReceiveResponse"></wsdl:part>
    </wsdl:message>
    <portType name="ExchangeTransportPort">
        <operation name="send">
            <input message="tns:sendRequest" />
            <output message="tns:sendResponse" />
        </operation>
        <operation name="receive">
            <input message="tns:receiveRequest" />
            <output message="tns:receiveResponse" />
        </operation>
        <operation name="confirmReceive">
            <input message="tns:confirmReceiveRequest" />
            <output message="tns:confirmReceiveResponse" />
        </operation>
        <operation name="groupReceive">
            <input message="tns:groupReceiveRequest"></input>
            <output message="tns:groupReceiveResponse"></output>
        </operation>
        <wsdl:operation name="confirmGroupReceive">
            <wsdl:input message="tns:confirmGroupReceiveRequest"></wsdl:input>
            <wsdl:output message="tns:confirmGroupReceiveResponse"></wsdl:output>
        </wsdl:operation>
    </portType>
    <binding name="MessageTransportBinding"
type="tns:ExchangeTransportPort">
        <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
        <operation name="send">
            <soap:operation soapAction="send" style="document" />
            <input>
                <soap:body use="literal" />
            </input>
            <output>
                <soap:body use="literal" />
            </output>
        </operation>
        <operation name="receive">
            <soap:operation soapAction="receive" style="document" />
            <input>
                <soap:body use="literal" />
            </input>

```

```

        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="confirmReceive">
        <soap:operation soapAction="confirmReceive"
            style="document" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="groupReceive">
        <soap:operation
            soapAction="groupReceive" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
    <operation name="confirmGroupReceive">
        <soap:operation
            soapAction="confirmGroupReceive" />
        <input>
            <soap:body use="literal" />
        </input>
        <output>
            <soap:body use="literal" />
        </output>
    </operation>
</binding>
<service name="ExchangeTransportService">
    <port name="ExchangeTransportPort"
        binding="tns:MessageTransportBinding">
        <soap:address location="No Target Adress" />
    </port>
</service>
</wsdl:definitions>

```

A.2 交换安全服务接口定义 (exchange-security.wsdl)

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:eus="http://service.egs.org.cn/exchange/exchangesec"
  xmlns:eu="http://schemas.egs.org.cn/exchange/exchangesec"
  xmlns:ns="http://schema.egs.org.cn/exchange/cdt"
  targetNamespace="http://service.egs.org.cn/exchange/exchangesec">
  <types>
    <xs:schema>
      <xs:import
        namespace="http://schemas.egs.org.cn/exchange/exchangesec"
        schemaLocation="schema/exchange-security.xsd" />
    </xs:schema>
  </types>
  <message name="AuthenticateRequest">
    <part name="partAuthenticateRequest"
      element="eu:AuthenticateRequest" />
  </message>
  <message name="AuthenticateResponse">
    <part name="partAuthenticateResponse"
      element="eu:AuthenticateResponse" />
  </message>
  <message name="LogoutRequest">
    <part name="partLogoutRequest" element="eu:LogoutRequest" />
  </message>
  <message name="LogoutResponse">
    <part name="partLogoutResponse" element="eu:LogoutResponse" />
  </message>
  <portType name="ExchangeUserSecurityPortType">
    <operation name="authenticate">
      <input message="eus:AuthenticateRequest" />
      <output message="eus:AuthenticateResponse" />
    </operation>
    <operation name="logout">
      <input message="eus:LogoutRequest" />
      <output message="eus:LogoutResponse" />
    </operation>
  </portType>
</definitions>
```

```

</portType>
<binding name="ExchangeUserSecurityBinding"
  type="eus:ExchangeUserSecurityPortType">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="authenticate">
    <soap:operation soapAction="authenticate" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
  <operation name="logout">
    <soap:operation soapAction="logout" style="document" />
    <input>
      <soap:body use="literal" />
    </input>
    <output>
      <soap:body use="literal" />
    </output>
  </operation>
</binding>
<service name="ExchangeUserSecurity">
  <port name="ExchangeUserSecurityPort"
    binding="eus:ExchangeUserSecurityBinding">
    <soap:address
      location="http://localhost/ExchangeUserSecurity" />
  </port>
</service>
</definitions>

```

A.3 业务代理服务接口定义 (ExchangeBusinessProxyService.wsdl)

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:ebp="http://www.wondersgroup.com/ExchangeBusinessProxyService"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  name="ExchangeBusinessProxyService"
  targetNamespace="http://www.wondersgroup.com/ExchangeBusinessProxyService"

```

```

xmlns:ebpxsd="http://www.wondersgroup.com/ExchangeBusinessProxy-data-format"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsm="http://schemas.xmlsoap.org/ws/2005/02/rm/policy"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<wsdl:types>
  <xsd:schema>
    <xsd:import
      namespace="http://www.wondersgroup.com/ExchangeBusinessProxy-data-format"
      schemaLocation="schema/ExchangeBusinessProxy-data-format.xsd">
    </xsd:import>
  </xsd:schema>
</wsdl:types>
<wsdl:message name="proxySendRequest">
  <wsdl:part name="partBusinessProxyRequest"
    element="ebpxsd:BusinessProxyRequest">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="proxySendResponse">
  <wsdl:part name="partBusinessProxyResponse"
    element="ebpxsd:BusinessProxyResponse">
  </wsdl:part>
</wsdl:message>

<wsdl:portType name="ExchangeBusinessProxyService">
  <wsdl:operation name="proxySend">
    <wsdl:input message="ebp:proxySendRequest"></wsdl:input>
    <wsdl:output message="ebp:proxySendResponse"></wsdl:output>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ExchangeBusinessProxyServiceSOAP"
  type="ebp:ExchangeBusinessProxyService">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http" />
  <wsdl:operation name="proxySend">
    <soap:operation
      soapAction="http://www.wondersgroup.com/ExchangeBusinessProxyService/bpsend" />
    <wsdl:input>
      <soap:body use="literal" />
    </wsdl:input>
    <wsdl:output>
      <soap:body use="literal" />
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>

```

```

    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="ExchangeBusinessProxyService">
    <wsdl:port binding="ebp:ExchangeBusinessProxyServiceSOAP"
      name="ExchangeBusinessProxyServiceSOAP">
      <soap:address location="http://localhost" />
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>

```

A. 4 交换同步传输服务XML消息定义 (ExchangeTransportSystem.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://schema.egs.org.cn/exchange/exchangetransport"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:edf="http://schema.egs.org.cn/exchange/edf"
  xmlns:cdt="http://schema.egs.org.cn/exchange/cdt"
  targetNamespace="http://schema.egs.org.cn/exchange/exchangetransport"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:import namespace="http://schema.egs.org.cn/exchange/cdt"
    schemaLocation="coreDataTypes.xsd" />
  <xs:import namespace="http://schema.egs.org.cn/exchange/edf"
    schemaLocation="exchange-data-format.xsd" />
  <xs:complexType name="ExchangeEventType">
    <xs:sequence>
      <xs:element ref="edf:ExchangeDataPackage" />
      <xs:element name="ExtentAttribute" type="xs:anyType"
        minOccurs="0" />
    </xs:sequence>
    <xs:attribute name="EventID" type="xs:string" use="required" />
    <xs:attribute name="ActionType" type="xs:string" use="required" />
  </xs:complexType>
  <xs:element name="ExchangeEvent" type="ExchangeEventType" />
  <xs:complexType name="SendRequestType">
    <xs:sequence>
      <xs:element name="ToAddress" type="xs:string"
        maxOccurs="unbounded" />
      <xs:element ref="ExchangeEvent" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="SendResponseType">
    <xs:sequence>
      <xs:element name="SendResults"
        type="SendResponseResultsType" maxOccurs="unbounded"

```

```

        minOccurs="0">
        </xs:element>
        <xs:element name="SendResult" type="xs:boolean"></xs:element>
        <xs:element name="GenericFault" type="cdt:GenericFaultType" maxOccurs="1"
minOccurs="0"></xs:element>
    </xs:sequence>
</xs:complexType>
<xs:element name="SendResponse" type="SendResponseType" />
<xs:element name="SendRequest" type="SendRequestType" />
<xs:complexType name="ReceiveRequestType">
    <xs:attribute name="ReceiveNumber" type="xs:int" use="required" />
    <xs:attribute name="Timeout" type="xs:long" />
    <xs:attribute name="ISBloacked" type="xs:boolean" />
</xs:complexType>
<xs:element name="ReceiveRequest" type="ReceiveRequestType" />
<xs:complexType name="ReceiveResponseType">
    <xs:choice>
        <xs:element name="ReceiveExchangeEvent" minOccurs="0"
maxOccurs="unbounded">
            <xs:complexType>
                <xs:sequence>
                    <xs:element name="ReceivedEvent"
type="ExchangeEventType" />
                </xs:sequence>
                <xs:attribute name="ReceivedTime"
type="xs:dateTime" />
                <xs:attribute name="EventSender" type="xs:string" />
                <xs:attribute name="EventReceiver"
type="xs:string">
                    </xs:attribute>
                </xs:complexType>
            </xs:element>
            <xs:element name="GenericFault" type="cdt:GenericFaultType" maxOccurs="1"
minOccurs="0"></xs:element>
        </xs:choice>
        <xs:attribute name="TotalEventCount" type="xs:int" />
    </xs:complexType>
<xs:element name="ReceiveResponse" type="ReceiveResponseType" />
<xs:complexType name="ConfirmReceiveRequestType">
    <xs:sequence>
        <xs:element name="ConfirmEventElement"
type="ConfirmReceiveElementType" maxOccurs="unbounded" />
    </xs:sequence>

```

```

        <xs:attribute name="ISDeleteEvent" type="xs:boolean"
            use="required" />
    </xs:complexType>
    <xs:element name="ConfirmReceiveRequest"
        type="ConfirmReceiveRequestType" />
    <xs:complexType name="ConfirmReceiveResponseType">
        <xs:sequence>
            <xs:element name="confirmResult" type="xs:boolean" />
            <xs:element name="GenericFault" type="cdt:GenericFaultType"
                maxOccurs="1"
minOccurs="0"></xs:element>

            <xs:element name="confirmResults"
                type="ConfirmReceiveResponseResultsType"
                maxOccurs="unbounded"
minOccurs="0">
                </xs:element>
            </xs:sequence>
        </xs:complexType>
    <xs:element name="ConfirmReceiveResponse"
        type="ConfirmReceiveResponseType" />
    <xs:element name="SendResponseResults"
        type="SendResponseResultsType">
    </xs:element>
    <xs:complexType name="SendResponseResultsType">
        <xs:sequence>
            <xs:element name="EventID" type="xs:string"></xs:element>
            <xs:element name="Result" type="xs:boolean"></xs:element>
            <xs:element name="GenericFault"
                type="cdt:GenericFaultType" maxOccurs="1" minOccurs="0">
            </xs:element>
            <xs:element name="ToAddress" type="xs:string"></xs:element>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="GroupReceiveRequestType">

        <xs:attribute name="ReceiveNumber" type="xs:int"></xs:attribute>
        <xs:attribute name="Timeout" type="xs:long"></xs:attribute>
        <xs:attribute name="IsBlocked" type="xs:boolean"></xs:attribute>
    </xs:complexType>
    <xs:element name="GroupReceiveResponse"
        type="GroupReceiveResponseType">
    </xs:element>
    <xs:complexType name="GroupReceiveResponseType">
        <xs:choice>

```

```

    <xs:element name="ReceiveExchangeEvent" maxOccurs="unbounded"
      minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="ReceivedEvent"
            type="ExchangeEventType" />
        </xs:sequence>
        <xs:attribute name="ReceivedTime" type="xs:dateTime" />
        <xs:attribute name="EventSender" type="xs:string" />
        <xs:attribute name="EventReceiver" type="xs:string">
        </xs:attribute>
      </xs:complexType>
    </xs:element>
    <xs:element name="GenericFault" type="cdt:GenericFaultType" maxOccurs="1"
minOccurs="0"></xs:element>
  </xs:choice>
  <xs:attribute name="TotalEventCount" type="xs:int"></xs:attribute>
</xs:complexType>
<xs:element name="GroupReceiveRequest"
  type="GroupReceiveRequestType">
</xs:element>
<xs:element name="ConfirmReceiveResponseResults"
  type="ConfirmReceiveResponseResultsType">
</xs:element>
<xs:complexType name="ConfirmReceiveResponseResultsType">
  <xs:sequence>
    <xs:element name="EventID" type="xs:string"></xs:element>
    <xs:element name="ToAddress" type="xs:string"></xs:element>
    <xs:element name="Result" type="xs:boolean"></xs:element>
    <xs:element name="GenericFault" type="cdt:GenericFaultType"
      maxOccurs="1" minOccurs="0">
    </xs:element>
  </xs:sequence>
</xs:complexType>
<xs:element name="ConfirmGroupReceiveRequest"
  type="ConfirmGroupReceiveRequestType">
</xs:element>
<xs:complexType name="ConfirmGroupReceiveRequestType">
  <xs:sequence>
    <xs:element name="ConfirmEventElement"
      type="ConfirmReceiveElementType" maxOccurs="unbounded"
      minOccurs="1">
    </xs:element>

```

```

    </xs:sequence>
    <xs:attribute name="ISDeleteEvent" type="xs:boolean"></xs:attribute>
  </xs:complexType>
  <xs:element name="ConfirmGroupReceiveResponse"
    type="ConfirmGroupReceiveResponseType">
  </xs:element>
  <xs:complexType name="ConfirmGroupReceiveResponseType">
    <xs:sequence>
      <xs:element name="confirmResult" type="xs:boolean"></xs:element>
      <xs:element name="GenericFault"
        type="cdt:GenericFaultType" maxOccurs="1" minOccurs="0">
      </xs:element>
      <xs:element name="confirmResults" type="ConfirmReceiveResponseResultsType"
maxOccurs="unbounded" minOccurs="0"></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="ConfirmReceiveElement"
    type="ConfirmReceiveElementType">
  </xs:element>
  <xs:complexType name="ConfirmReceiveElementType">
    <xs:attribute name="EventID" type="xs:string"></xs:attribute>
    <xs:attribute name="ISSuccessReceive" type="xs:boolean"></xs:attribute>
    <xs:attribute name="Receiver" type="xs:string"></xs:attribute>
  </xs:complexType>
</xs:schema>

```

A.5 业务代理服务XML消息定义 (ExchangeBusinessProxy-data-format. xsd)

业务代理服务接口的请求/响应消息的XML Schema定义。

```

<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="http://www.wondersgroup.com/ExchangeBusinessProxy-data-format"
  elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ebp="http://www.wondersgroup.com/ExchangeBusinessProxy-data-format"
  xmlns:ebpc="http://www.wondersgroup.com/ExchangeBusinessProxyCore-data-format.xsd"
  xmlns:cdt="http://schema.egs.org.cn/exchange/cdt">
  <import namespace="http://schema.egs.org.cn/exchange/cdt"
    schemaLocation="coreDataTypes.xsd">
  </import>
  <element name="BusinessProxyRequest" type="ebp:BusinessProxyRequestType">
  </element>
  <complexType name="BusinessProxyRequestType">
    <sequence>
      <element name="BusinessXML" type="base64Binary"

```

```

        maxOccurs="1" minOccurs="0">
        </element>
    </sequence>
    <attribute name="ProxyID" type="string"></attribute>
    <attribute name="ActionType" type="string"></attribute>
</complexType>
<element name="BusinessProxyResponse"
    type="ebp:BusinessProxyResponseType">
</element>
<complexType name="BusinessProxyResponseType">
    <choice>
        <element name="ErrorMessage" type="cdt:GenericFaultType"></element>
    </choice>
    <attribute name="Result" type="boolean"></attribute>
    <attribute name="ResultXML" type="base64Binary"></attribute>
</complexType>
</schema>

```

A. 6 交换安全服务XML消息定义 (exchange-security.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
    xmlns="http://schemas.egs.org.cn/exchange/exchangesec"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:cdt="http://schema.egs.org.cn/exchange/cdt"
    targetNamespace="http://schemas.egs.org.cn/exchange/exchangesec"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <xs:import namespace="http://schema.egs.org.cn/exchange/cdt" schemaLocation="coreDataTypes.xsd"/>
    <xs:complexType name="AuthenticateRequestType">
        <xs:attribute name="UserName" type="xs:string" use="required"/>
        <xs:attribute name="Password" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:element name="AuthenticateRequest" type="AuthenticateRequestType"/>
    <xs:complexType name="AuthenticateResponseType">
        <xs:choice>
            <xs:element ref="cdt:GenericFault"/>
        </xs:choice>
        <xs:attribute name="UserTokenID" type="xs:string"/>
    </xs:complexType>
    <xs:element name="AuthenticateResponse" type="AuthenticateResponseType"/>
    <xs:complexType name="LogoutRequestType">
        <xs:attribute name="UserName" type="xs:string" use="required"/>
        <xs:attribute name="UserTokenID" type="xs:string" use="required"/>
    </xs:complexType>
    <xs:element name="LogoutRequest" type="LogoutRequestType"/>
    <xs:complexType name="LogoutResponseType">

```

```

    <xs:choice>
      <xs:element name="LogoutResult" type="xs:boolean"/>
      <xs:element ref="cdt:GenericFault"/>
    </xs:choice>
  </xs:complexType>
  <xs:element name="LogoutResponse" type="LogoutResponseType"/>
</xs:schema>

```

A.7 基本数据类型XML消息定义 (coreDataTypes.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:cdt="http://schema.egs.org.cn/exchange/cdt"
targetNamespace="http://schema.egs.org.cn/exchange/cdt" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:simpleType name="QueryLanguageType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="HQL"/>
      <xs:enumeration value="SQL92"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="DataSetFormat">
    <xs:restriction base="xs:string">
      <xs:enumeration value="BerkelyDataFile"/>
      <xs:enumeration value="DerbyDataFile"/>
      <xs:enumeration value="csv"/>
      <xs:enumeration value="businessdata"/>
      <xs:enumeration value="simplifiedata"/>
      <xs:enumeration value="InlineXML"/>
      <xs:enumeration value="Customized"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="QueryParameterType">
    <xs:attribute name="parameterName" type="xs:string"/>
    <xs:attribute name="parameterValue" type="xs:string"/>
  </xs:complexType>
  <xs:element name="QueryParameter" type="cdt:QueryParameterType"/>
  <xs:simpleType name="ChangeDataType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Insert"/>
      <xs:enumeration value="Update"/>
      <xs:enumeration value="Delete"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="GenericFaultType">

```

```

    <xs:sequence>
      <xs:element name="Code" type="xs:string"/>
      <xs:element name="ErrorMessage" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="GenericFault" type="cdt:GenericFaultType"/>
</xs:schema>

```

A.8 数据格式XML消息定义 (exchange-data-format.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:egsbd="http://schemas.egs.org.cn/businessdata"
  xmlns="http://schema.egs.org.cn/exchange/edf"
  targetNamespace="http://schema.egs.org.cn/exchange/edf"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:import
    namespace="http://schemas.egs.org.cn/businessdata"
    schemaLocation="egs-org-cn-businessdata.xsd"/>
  <xs:complexType name="SimpleDataType">
    <xs:attribute name="FieldName" type="xs:string" use="required"/>
    <xs:attribute name="FieldValue" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:element name="SimpleData" type="SimpleDataType"/>
  <xs:complexType name="DataFileType">
    <xs:choice>
      <xs:element name="Base64EncodedData" type="xs:base64Binary"/>
      <xs:element name="AttachmentData" type="xs:base64Binary"/>
    </xs:choice>
    <xs:attribute name="dataFileID" type="xs:NCName" use="required"/>
    <xs:attribute name="fileName" type="xs:string" use="required"/>
    <xs:attribute name="dataFileFormat" type="xs:string" use="required"/>
  </xs:complexType>
  <xs:complexType name="ExchangeDataPackageUnitType">
    <xs:sequence>
      <xs:element name="ExtendedElement" type="xs:anyType" minOccurs="0"/>
      <xs:choice>
        <xs:element name="DataFile" type="DataFileType"/>
        <xs:element name="BusinessData" type="egsbd:BusinessDataType"/>
        <xs:element ref="SimpleData" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="groupID" type="xs:string" use="required"/>
    <xs:attribute name="groupSize" type="xs:int" use="required"/>
    <xs:attribute name="sequenceInGroup" type="xs:int" use="required"/>
    <xs:attribute name="unitID" type="xs:string" use="required"/>

```

```

    <xs:attribute name="source" type="xs:string" use="required"/>
    <xs:attribute name="entityIDName" type="xs:anyURI" use="required"/>
    <xs:attribute name="entityDispName" type="xs:string"/>
    <xs:attribute name="createDate" type="xs:dateTime"/>
    <xs:attribute name="expireDate" type="xs:dateTime"/>
    <xs:anyAttribute/>
  </xs:complexType>
  <xs:complexType name="ExchangeDataPackageType">
    <xs:sequence>
      <xs:element ref="ExchangeDataPackageUnit" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="packageID" type="xs:string" use="required"/>
    <xs:attribute name="transactionID" type="xs:string" use="required"/>
    <xs:attribute name="createTime" type="xs:dateTime" use="required"/>
    <xs:attribute name="expireTime" type="xs:dateTime"/>
    <xs:attribute name="title" type="xs:string" use="required"/>
    <xs:anyAttribute/>
  </xs:complexType>
  <xs:element name="ExchangeDataPackageUnit" type="ExchangeDataPackageUnitType"/>
  <xs:element name="ExchangeDataPackage" type="ExchangeDataPackageType"/>
</xs:schema>

```

A.9 业务数据表示格式XML消息定义 (egs-org-cn-businessdata.xsd)

```

<?xml version="1.0" encoding="GB18030"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.egs.org.cn/businessdata"
  targetNamespace="http://schemas.egs.org.cn/businessdata"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="BusinessData" type="BusinessDataType"/>
  <xs:complexType name="BusinessDataType">
    <xs:sequence>
      <xs:element ref="DataStructure" minOccurs="0"/>
      <xs:element ref="DataSet" minOccurs="0"/>
      <xs:element ref="Attachments" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="DataStructure" type="DataStructureType"/>
  <xs:complexType name="DataStructureType">
    <xs:sequence>
      <xs:element name="Identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
      <xs:element name="ExplanatoryComment" type="xs:string" minOccurs="0"/>
      <xs:element ref="DataUnit" maxOccurs="unbounded"/>
      <xs:element ref="ExtendAttribute" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

```

```

    </xs:sequence>
</xs:complexType>
<xs:element name="DataUnit" type="DataUnitType"/>
<xs:complexType name="DataUnitType">
  <xs:sequence>
    <xs:element name="IDName" type="xs:string"/>
    <xs:element name="DisplayName" type="xs:string"/>
    <xs:element name="Definition" type="xs:anyType" minOccurs="0"/>
    <xs:element name="Comments" type="xs:string" minOccurs="0"/>
    <xs:element name="Datatype" type="xs:string"/>
    <xs:element name="MaximumSize" type="xs:int" minOccurs="0"/>
    <xs:element name="MinimumSize" type="xs:int" minOccurs="0"/>
    <xs:element name="Scale" type="xs:int" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="ExtendAttribute" type="ExtendAttributeType"/>
<xs:complexType name="ExtendAttributeType">
  <xs:sequence>
    <xs:element name="AttributeName" type="xs:string"/>
    <xs:element name="AttributeValue" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="UnitDataType">
  <xs:sequence>
    <xs:element name="UnitIDName" type="xs:string"/>
    <xs:element name="UnitDisplayName" type="xs:string"/>
    <xs:element name="UnitValue" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RecordDataType">
  <xs:sequence>
    <xs:element name="UnitData" type="UnitDataType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DataSetType">
  <xs:sequence>
    <xs:element name="RecordData" type="RecordDataType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="DataSet" type="DataSetType"/>
<xs:complexType name="AttachmentType">
  <xs:sequence>
    <xs:element name="IDName" type="xs:string"/>

```

```

    <xs:element name="DisplayName" type="xs:string" minOccurs="0"/>
    <xs:element name="Description" type="xs:string" minOccurs="0"/>
    <xs:element name="TypeDescription" type="xs:string" minOccurs="0"/>
    <xs:element name="Content" type="xs:anyType"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="AttachmentsType">
  <xs:sequence>
    <xs:element name="Attachment" type="AttachmentType" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="Attachments" type="AttachmentsType"/>
</xs:schema>

```

A. 10 FTP协议数据格式XML消息定义 (FtpExchangeTransport.xsd)

```

<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://schema.egs.org.cn/exchange/ftpexchangetransport"
  elementFormDefault="qualified" xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schema.egs.org.cn/exchange/ftpexchangetransport">
  <element name="FtpSend" type="tns:FtpSendType"></element>
  <complexType name="FtpSendType">
    <sequence>
      <element name="ToAddress" type="string"
        maxOccurs="unbounded" minOccurs="1">
      </element>
      <element name="ExchangeEvent"
        type="tns:ExchangeEventType" maxOccurs="unbounded" minOccurs="1">
      </element>
    </sequence>
  </complexType>
  <complexType name="ExchangeEventType">
    <sequence>
      <element name="ExchangeDataPackage" type="tns:ExchangeDataPackageType"></element>
      <element name="ExtentAttribute" type="anyType" maxOccurs="1"
minOccurs="0"></element>
    </sequence>
    <attribute name="EventID" type="string"></attribute>
    <attribute name="ActionType" type="string"></attribute>
  </complexType>
  <complexType name="ExchangeDataPackageType">

```

```

    <sequence>
      <element name="ExchangeDataPackageUnit" type="tns:ExchangeDataPackageUnitType"
maxOccurs="unbounded" minOccurs="1"></element>
    </sequence>
    <attribute name="packageID" type="string"></attribute>
    <attribute name="transactionID" type="string"></attribute>
    <attribute name="createTime" type="dateTime"></attribute>
    <attribute name="expireTime" type="dateTime"></attribute>
    <attribute name="title" type="string"></attribute>
    <anyAttribute></anyAttribute>
  </complexType>
  <complexType name="ExchangeDataPackageUnitType">
    <sequence>
      <element name="ExtendedElement" type="anyType" maxOccurs="1"
minOccurs="0"></element>
      <choice>
        <element name="DataFile" type="tns:DataFileType"></element>
        <element name="SimpleData" type="tns:SimpleDataType" maxOccurs="unbounded"
minOccurs="1"></element>
      </choice>
    </sequence>
    <attribute name="groupID" type="string" use="required" />
    <attribute name="groupSize" type="int" use="required" />
    <attribute name="sequenceInGroup" type="int" use="required" />
    <attribute name="unitID" type="string" use="required" />
    <attribute name="source" type="string" use="required" />
    <attribute name="entityIDName" type="anyURI" use="required" />
    <attribute name="entityDispName" type="string" />
    <attribute name="createDate" type="dateTime" />
    <attribute name="expireDate" type="dateTime" />
    <anyAttribute />
  </complexType>
  <complexType name="DataFileType">
    <choice>
      <element name="FtpContentData" type="string"></element>
      <element name="AttachmentData" type="string"></element>
    </choice>
    <attribute name="dataFileID" type="string"></attribute>
    <attribute name="fileName" type="string"></attribute>
    <attribute name="dataFileFormat" type="string"></attribute>
  </complexType>
  <complexType name="SimpleDataType">
    <attribute name="FieldName" type="string"></attribute>

```

```

    <attribute name="FieldValue" type="string"></attribute>
</complexType>
<element name="FtpReceive" type="tns:FtpReceiveType"></element>
<complexType name="FtpReceiveType">
    <sequence>
        <element name="ReceiveExchangeEvent"
            type="tns:ReceiveExchangeEventType" maxOccurs="unbounded"
            minOccurs="1">
        </element>
    </sequence>
    <attribute name="TotalEventCount" type="int"></attribute>
</complexType>
<complexType name="ReceiveExchangeEventType">
    <sequence>
        <element name="ReceivedEvent"
            type="tns:ExchangeEventType" maxOccurs="unbounded" minOccurs="1">
        </element>
    </sequence>
    <attribute name="ReceivedTime" type="dateTime"></attribute>
    <attribute name="EventSender" type="string"></attribute>
    <attribute name="EventReceiver" type="string"></attribute>
</complexType>
</schema>

```

附录B
(资料性附录)
API接口说明

API是交通运输物流公共信息平台提供给物流通用软件和重要物流及相关信息系统的开发接口，对数据交换服务接口进行封装，包含多个调用方法。软件开发商可利用API进行开发，提高开发效率。

B.1 Cuteinfo客户端 API 接口说明(.Net版)

见《Cuteinfo(.Net版)客户端API接口说明书》

B.2 Cuteinfo客户端API接口说明(Java版)

见《Cuteinfo(Java版)客户端API接口说明书》

附录C
 (资料性附录)
 交换\应用服务器列表

序号	分类	服务器	名称地址	说明
1	正式地址	一号交换中心	http://exa.logink.org:81/cuteinfo	
2		二号交换中心	http://exb.logink.org/cuteinfo	
3		三号交换中心	http://exc.logink.org/cuteinfo	
4		交换管理中心	http://eac.logink.org/eac	
5	测试地址	测试交换中心	http://test.logink.org:81/cuteinfo	
6		测试管理中心	http://test.logink.org/eac	
7		测试货物跟踪	http://test.logink.org/cts	
8		测试通用网站	http://test.logink.org/zjhy	

