



RCloud

产品用户使用手册

(RCloud 云 Exchange-Integration 服务)

北京中软国际信息技术有限公司

2010 年

目 录

第一章 概述.....	4
1.1 面向读者.....	4
1.2 用户导读.....	5
1.3 名词解释.....	6
第二章 安装部署指南	8
2.1 部署环境.....	8
2.2 产品结构.....	9
2.2.1 产品构成.....	9
2.2.2 发布包结构.....	10
2.3 使用模式.....	11
2.3.1 单总线.....	12
2.3.2 复合总线.....	12
2.4 数据库搭建.....	13
2.4.1 IBM DB2	14
2.4.2 Oracle.....	15
2.4.3 MS SQL Server.....	17
2.5 应用环境搭建.....	17
2.5.1 Websphere6.1 下环境搭建.....	17
2.5.2 WebSphere5.1 下搭建环境.....	26
2.5.3 金蝶 Apusic 5.1 下搭建环境.....	31
2.5.4 Weblogic9.2 下搭建环境.....	36
2.5.5 安装后的形态.....	37
2.6 连接器部署配置.....	39
2.6.1 DE Agent 的配置	39
2.6.2 SOAP/WS 连接器配置	41
2.6.3 HTTP 连接器配置	41

2.6.4 IBM-MQ 连接器配置.....	41
2.6.5 FTP 连接器.....	42
2.6.6 文件连接器.....	42
2.6.7 TongLINKQ 连接器配置.....	42
2.6.8 Win32Agent 连接器配置.....	43
2.6.9 数据库 Agent 触发器.....	43
2.7 RCLOUD 云 EI 服务服务器配置.....	50
2.7.1 传输.....	50
2.7.2 自动测试链路状态.....	53
2.7.3 日志.....	53
2.7.4 RODEConfig.xml 的 Security 配置.....	55
2.7.5 消息转发策略配置.....	55
2.8 RCLOUD 云 EI 服务 设计工具安装	67
第三章 集成交换设计与管理指南	68
3.1 总述.....	68
3.2 RCLOUD 云 EI 服务设计工具使用指南	71
3.2.1 运行 RCloud Studio.....	71
3.2.2 连接配置.....	72
3.2.3 消息格式定义.....	77
3.2.4 格式转换器定义.....	87
3.2.5 消息流定义.....	94
3.2.6 代理服务设计与发布.....	127
3.2.7 导入、导出设计资源.....	132
3.2.8 WSDL 定义.....	144
3.3 RCLOUD 云 EI 服务管理控制台使用指南	144
3.3.1 访问控制.....	145
3.3.2 首页概述.....	146
3.3.3 新建 RCloud 云 EI 服务总线.....	148

3.3.4 RCloud 云 EI 服务总线管理.....	198
3.3.5 多管控配置.....	237
第四章 注意事项及优化指南	242
4.1 总线部署使用时的注意事项	242
4.2 RCLOUD 云 EI 服务部署时的注意事项.....	243
4.2.1 RCloud 云 EI 服务服务注册.....	243
4.2.2 DE Agent 客户端部署注意事项.....	243
4.2.3 RCloud 云 EI 服务服务器部署注意事项.....	244
4.3 RCLOUD 云 EI 服务服务器配置建议.....	244
附录一 RCLOUD 云 EI 服务 WEB SERVICE 访问安全配置	246
5.1 概述	246
5.2 安装安全模块	246
5.3 配置服务安全策略	246
5.3.1 身份验证配置.....	246
5.3.2 消息数字签名.....	252
5.3.3 消息加密配置.....	255
5.3.4 组合使用配置.....	259
5.3.5 发送消息时动态指定安全策略.....	265
5.3.6 使用策略描述文件.....	269
附录二 通过 MQ (V5.3) 资源管理器使用.....	278
6.1. 队列管理器的创建	278
6.1.1. 队列的创建.....	280
6.1.2. 通道的创建.....	282
附录三 MQ 传输配置示例.....	286
7.1. 节点信息	287
7.2. 节点 A 的配置：	287

7.2.1.MQ	287
7.3.节点 B.A 的配置:	288
7.3.1.MQ	288
7.3.2.WebSphere	289
7.4.节点 C.A 的配置:	290
7.4.1.MQ	290
7.5.节点 D.B.A 的配置:	291
7.5.1.MQ	291
附录四 TONGLINK/Q7.1 传输配置示例	292
8.1 TONGLINK /Q7.1 使用概述	292
8.1.1 TLQ 连接模式.....	292
8.1.2 TLQ 消息传输简述.....	292
8.2 TLQ 传输配置示例	293
8.2.1 A 点 (发送端) etc 配置文件.....	293
8.2.2 B 点 (接收端) etc 配置文件.....	296
8.2.3 启动 TLQ 以及调试诊断.....	297
8.2.4 测试 TLQ 物理链路是否畅通.....	298
附录五 数据源配置	302
容器管理数据源	303
RCLOUD 云 EI 服务管理数据源（使用外部数据库）	304
RCLOUD 云 EI 服务管理数据源（使用内嵌数据库）	305

第一章 概述

1.1 面向读者

本手册详细描述了 RCloud 云 Exchange-Integration 服务（以下简称 RCloud

云 EI 服务) 的安装部署过程及各组件使用方法, 主要面向基于该产品的二次开发人员、实施部署人员以及最终用户的系统管理人员。

1.2 用户导读

第二章内容介绍了 RCloud 云 EI 服务的部署环境、产品的发布包结构和完成安装后生成的文件目录结构, 并针对多种主流企业级中间件产品(数据库、应用服务器等)分别详细描述了产品安装方法、数据库创建步骤以及资源配置等内容。可按照如下步骤安装部署 RCloud 云 EI 服务交换网络:

1、准备数据库环境(创建数据库、执行初始化脚本) 2.4

2、在应用服务器中搭建运行环境:

 1) 部署 RCloud 云 EI 服务管理控制台 2.5

 2) 部署 RCloud 云 EI 服务服务器 2.5

 3) 部署 RCloud 云 EI 服务管理代理服务器 2.5

3、部署客户端应用

安装 DEMO 应用, DEMO 应用中含有 DE Agent 客户端处理程序。

第三章内容是使用 RCloud 云 EI 服务设计分布式业务应用之间的信息交换体系, 详述此过程中的功能特性及使用方法。下面以一个在面向服务(SOA)架构环境中基于企业服务总线(ESB)搭建业务交换场景的典型操作来引导参阅手册中各部分功能描述:

1、在 RCloud 云 EI 服务设计工具中以图形化的方式设计展现业务环境中的信息交换逻辑。包括消息流设计、消息格式设计以及不同消息格式之间的映射关系设计, 设计完成后保存到 RCloud 云 EI 服务服务器上 3.2

2、根据实际业务应用的分布情况以及信息交换的具体要求, 选择适当的部署模式: 单总线模式 2.3.1 与复合总线模式 2.3.2

3、在管理控制台中创建服务总线, 注册、配置及管理各类资源, 构建整个业务交换场景。使用管理员账号登录 RCloud 云 EI 服务管理控制台 3.3.1

4、在管理控制台中构建 RCloud 云 EI 服务服务交换环境:

 1) 新建 RCloud 云 EI 服务总线 3.3.3.1(单总线) 或 3.3.3.2(复合总线)

2) 如果选择新建复合总线，则需要将当前 RCloud 云 EI 服务服务器的路由配置手动映射到所有物理服务器节点（单总线是自动映射的）。

3) 将业务环境中涉及到的应用、消息队列等资源注册为集成对象 3.3.3.3

4) 为总线中的各集成对象配置连接器，使其具备与总线的通信能力

3.3.3.4。其中 DEAgent、Excel 连接器三种连接器配置完成后，需要执行推送操作，并重新启动客户应用。

5) 选择在 RCloud 云 EI 服务设计工具中设计好的消息流，并将消息流中的各输入、输出节点与注册在 RCloud 云 EI 服务服务器上的集成对象所对应的连接器绑定。

6) 在管理控制台中启动连接器 3.3.3.5。

6、在交换系统运行过程中，用户可对整个 RCloud 云 EI 服务总线的运行状态及运行时数据进行全面的监控和管理：

1) 服务器监控。监控连接器 3.3.4.1.1、交换引擎 3.3.4.1.2 的服务状态，并可以对服务引擎、连接器进行启停操控。
2) 查看当前服务器下集成对象日志 3.3.4.1.3 以及服务器日志 3.3.4.1.4。
3) 服务器端异常消息 3.3.4.2.1、客户端异常消息 3.3.4.2.2 查看及管理。
4) 管理数据源配置给交换服务器端的定时任务和 java 扩展提供访问数据库连接。3.3.4.3

5) 查看定时服务器和定时任务的状态，并通过启停来管理定时服务器 3.3.4.4.1 和定时任务 3.3.4.4.2。

6) 统计 RCloud 云 EI 服务服务器、客户端和消息流的交换记录数和流量。
3.3.4.5

7) 消息流定时备份。3.3.4.6

1.3 名词解释

◆ 消息

是按一定规范封装的数据包，应用与 RCloud 云 EI 服务总线的数据交换是通过消息来传递的。

◆ 路由

在数据交换过程中寻找消息目的地(一般为一个 RCloud 云 EI 服务服务器或者相应连接器及服务等)的过程叫路由。

◆ 路由表

多个 RCloud 云 EI 服务服务器按照一定拓扑关系组成一棵树型结构，树中每个 RCloud 云 EI 服务服务器都有自己的唯一标识，并且与上下级交换服务器存在逻辑结构上的父子关系(但事实上相连的 RCloud 云 EI 服务服务器间关系对等，不存在上下级关系)，形成了 RCloud 云 EI 服务总线的拓扑结构；RCloud 云 EI 服务服务器间的这种关系构成了一份路由表，是消息在 RCloud 云 EI 服务服务器间传输路径的依据。一个 RCloud 云 EI 服务总线有且只有一份路由表。

◆ RCloud 管理代理

RCloud 云 EI 服务采用基于 RCloud MC 规范的分布式管理框架，要求每个被管理的 RCloud 云 EI 服务服务器所在的物理主机上必须部署相应的 RCloud 管理代理模块，提供来自管理控制台的管理请求代理功能。

◆ RCloud 云 EI 服务交换总线

提供了应用集成的连接中枢，可以消除不同应用之间的技术差异，让不同的应用协调运作，实现了不同服务之间的通信与整合。

◆ 消息流

消息流是在 RCloud 云 EI 服务设计工具中设计，以消息流 ID 号和版本号作为唯一标识，它描述了消息在 RCloud 云 EI 服务服务器中有序的、有步骤的处理过程。

◆ 消息映射

不同消息格式之间的转换规则，用户通过设计消息映射，使不同格式的消息能够互相转换。

◆ 集成对象

一个集成对象表示一个客户端应用或需要集成的遗留系统，集成对象通过连接器和 RCloud 云 EI 服务服务器连接。

◆ 连接器

连接器是集成对象接入 RCloud 云 EI 服务交换网络所必须的组件，它是集成对象与 RCloud 云 EI 服务服务器间的连接。目前 RCloud 云 EI 服务中提供的连接器类型有：DE Agent 连接器、IBM-MQ 连接器、HTTP 连接器、SOAP/WS 连接器、FTP 连接器、文件接入连接器等。

◆ 推送

推送是 RCloud 云 EI 服务管控中的一个操作，只用于 DEAgent 连接器、FTP 连接器、文件接入连接器的配置，是指将当前连接器配置信息及链路信息推送到所选择的客户端。需要注意的是，在当前连接器配置信息如连接器 ID、连接器与 RCloud 云 EI 服务服务器间传输方式等变化时，都需要再次进行“推送”操作。

◆ 消息处理器

消息处理器是客户端应用处理来自 RCloud 云 EI 服务服务器消息的组件。应用根据提供的接口实现特定功能的消息处理器，通过在管理控制台中 DEAgent 连接器维护部分的注册，与指定的连接器关联在一起。

第二章 安装部署指南

2.1 部署环境

操作系统： Microsoft Windows XP、Microsoft Windows 2003、Linux

JDK: SUN JDK 1.4 及以上、IBM JDK 1.4 及以上

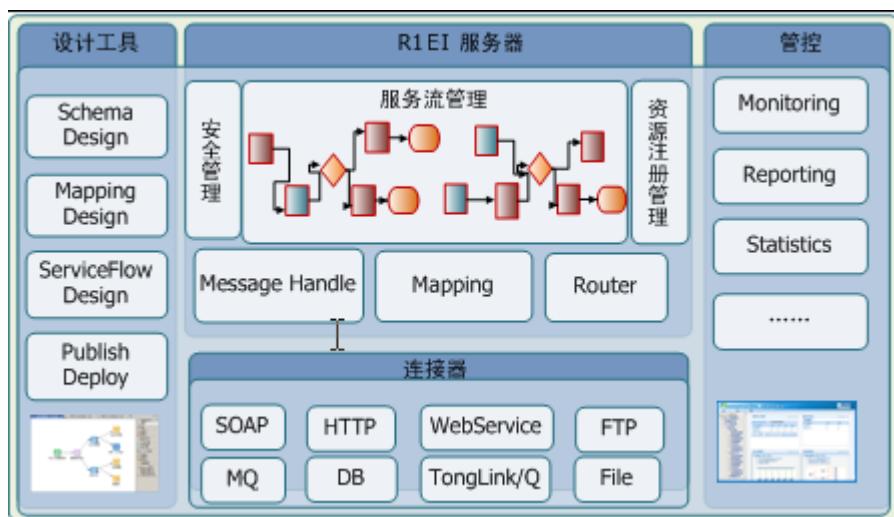
应用服务器: IBM WebSphere V5.x V6.x、Bea Weblogic V8.x V9.x V10, 金蝶 Apusic 5.1

数据库: IBM DB2 V8.x V9.x 、Oracle9i、Oracle10g、Oracle11g、SQL Server2000

2.2 产品结构

2.2.1 产品构成

RCloud 云 EI 服务是一个分布式的集成交换中间件软件平台, 是构建企业数据中枢和服务总线的支撑系统, 其总体结构如下图:



总体结构图

如总体结构图中所示, 从组件结构上分为以下四个主要部分:

1、RCloud 云 EI 服务服务器: 是服务总线运行时实体, 负责消息的映射转换、服务流的调度运行、消息路由传输、查找调用已注册服务等。

2、RCloud 云 EI 服务连接器: 负责客户应用系统接入 RCloud 云 EI 服务总线; 支持多种接入方式: DEAgent、SOAP/WS、HTTP、FTP、IBM-MQ、格式化的文件如 Excel 等。

3、RCloud 云 EI 服务设计工具: 集成在 RCloud Studio(RCloud 集成开发工具)中, 负责服务流的设计、消息格式设计、消息转换(映射)设计; 发布、部署

等。

RCloud 云 EI 服务管理监控工具：基于 RMC(RCloud Management Console) 规范实现，采用 B/S 结构，集成在 RCloud MC 中。负责企业服务总线的配置、服务流消息监控管理、总线拓扑结构配置部署、业务系统与服务流绑定、资源监控、统计。

2.2.2 发布包结构

RCloud 云 EI 服务产品的发布包是一个压缩文件，包含有各类产品文档和 RCloud 云 EI 服务的安装文件，目录结构如下(其中 4.x.x 指产品对应的版本号)：

目录结构	说明
db	下是各种数据库的建库脚本，目前包括 db2、Oracle 和 SQLServer 的建库脚本
doc	下是 RCloud 云 EI 服务产品的用户手册和部署手册
bin	下 RCloud 云 EI 服务四个组件的安装文件包
bin 文件夹下目录结构	
bin/server	
RCloudEISever 4.X.X.war	是 RCloud 云 EI 服务服务器安装包，是安装在 J2EE 应用服务器上的一个 J2EE 应用
bin/server/lib	当使用 TongLINKQ 或者 IBM-MQ 作为服务器间传输链路时，需要放到应用服务器 lib 目录下的 jar 包
bin/console	
RCloudManagementConsole-EIPlugin.zip	RCloud 云 EI 服务管理控制台插件安包，用于在已有的管理控制台(4.1 版本以上)上部署 RCloud 云 EI 服务管控
bin/client	
RCloudDEWin32Agen t4.X.X. zip	
Win32Agent 连接器压缩包	

RCloudDEAgent 4.X.X.zip	
是 RCloudDEAgent 部署压缩包，目录结构如下	
RCloudEIAgent 4.X.X.zip 压缩包目录结构	
build.number	Build 号，是标识本次打包的唯一标识
changelog.txt	SDK 的修改列表
readme.txt	说明信息
3rdlib	下是应用客户端需要的第三方 jar 包
Lib	包下是应用基于 RCloud 云 EI 服务开发的 SDK 包
classes	包含一个 rode.properties 文件，部署 DEAGENT 时，需要拷贝到 DEAGENT 应用的 classes 下
RCloudHome	<p>是应用客户端需要的配置文件，部署应用客户端后要把下面的配置文件拷贝到配置的 RCloudHome 目录下。</p> <p>(客户端 RCloudHome 目录可以在 classes/rode.properties 文件中的 resourceOneHome 属性中设定。默认根目录为中间件安装路径根目录下的 /RCloudHome 目录。例如：resourceOneHome=/RCloudHome)</p> <p>另外需要注意的是，如果 RCloud 云 EI 服务服务器和 DEAGENT 部署在同一台应用服务器的话，例如一个 IBM WebSphere 上已经部署了 RCloud 云 EI 服务服务器，那么部署 DEAGENT，只需要拷贝 DEAGENT 发布包 中 的 RCloudHome/DataExchange/RodeClientConfig.xml 文件到相应的 RCloudHome 下即可，其他文件无需拷贝</p>
wsdd	DEAgent 所在应用客户端需要的 webservice 部署描述符文件

2.3 使用模式

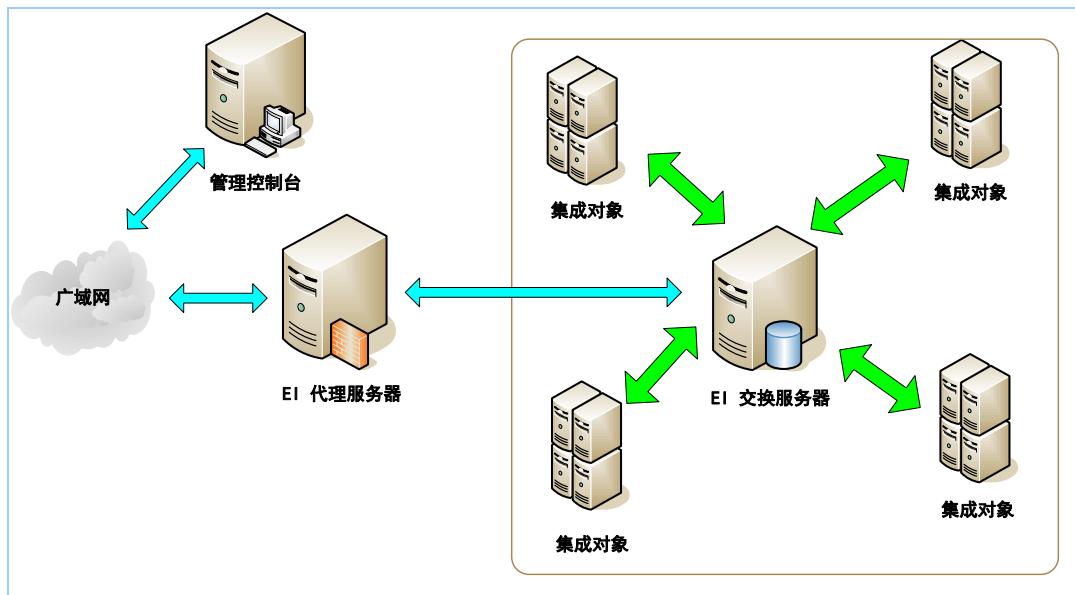
RCloud 云 EI 服务的 EI 服务器、连接器(SDK)、统一管理控制台 (RMC) 可以部署在不同的机器上，也可以部署在同一台机器上，不受物理位置限制，但是在一台物理机器上只能部署一个 RCloud 云 EI 服务服务器和一个 RCloud 管理代理服务器，而且 RCloud 云 EI 服务服务器和 RCloud 管理代理服务器必须部署在同一台机器上。RCloud 云 EI 服务可以实现单总线及分布式环境下的复

合总线两种部署模式。

2.3.1 单总线

单总线是指在同一网络环境中，部署一个独立的交换服务总线。应用、服务、数据等各类集成对象通过该总线进行信息交互。

通常情况下，集成对象是包含在应用中的，而应用本身可以跟 RCloud 云 EI 服务服务器部署在同一台机器上也可以在不同的机器上。

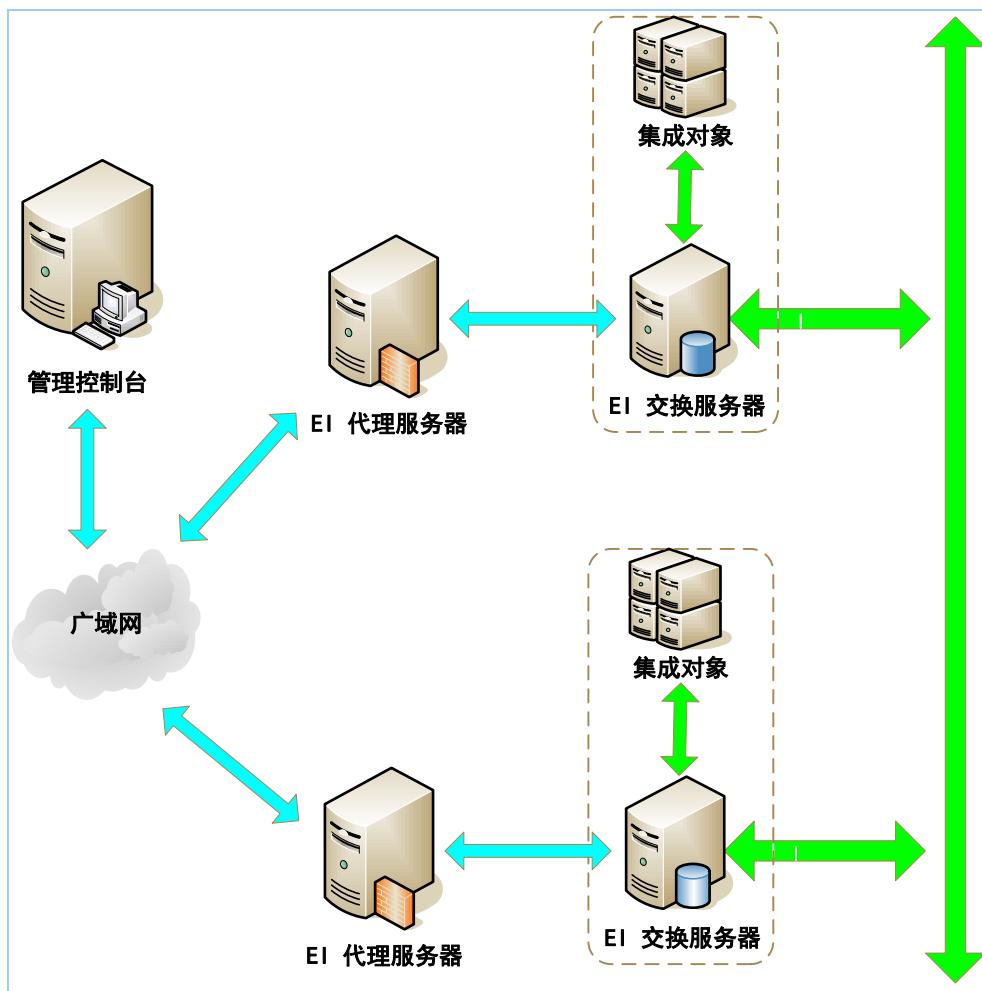


单总线部署示意图

2.3.2 复合总线

复合总线的部署模式适用于分布式环境下多个应用系统之间的信息交换，最典型的场景是依照部-省-市-县复合总线部署的应用系统，在分布式环境中的每一节点上部署一套交换服务总线，同一节点上的交换通过本节点的总线完成即可，不同节点之间的交换通过复合总线的信息转发完成。

RCloud 云 EI 服务管理控制台可以同时管理整个分布式环境中所有的 RCloud 云 EI 服务服务器及其管控的连接器等信息。



复合总线部署示意图

2.4 数据库搭建

RCloud 云 EI 服务分别对 IBM DB2、Oracle、SQLServer 等多种数据库提供了相应的建库与初始化脚本，实现对主流数据库的支持。对每一种数据库都包含如下脚本文件：

RCloudEI4.5_CreateTable.sql(创建表)

RCloudEI4.5_CreateKey.sql (创建主外键)

RCloudEI4.5_TableInitData.sql(初始化数据)

RCloudEI4.5_DropKey.sql (删除主外键)

RCloudEI4.5_DropTable.sql(删除表)

另外，针对IBM DB2 还提供了创建表空间的脚本：

RCloudEI4.5_CreateTableSpace.sql。

2.4.1 IBM DB2

首先创建一个RCloud 云EI服务的数据库

```
db2 create db [dbname]
```

其次连接该数据库，并依次执行数据库脚本：

```
db2 connect to [dbname] user [dbuser] using [password]  
//执行脚本时的执行路径需要指向脚本存放的路径下  
db2 -tvf RCloudEI4.5_CreateTableSpace.sql  
db2 -tvf RCloudEI4.5_CreateTable.sql  
db2 -tvf RCloudEI4.5_CreateKey.sql  
db2 -tvf RCloudEI4.5_TableInitData.sql
```

经过以上的步骤RCloud 云EI服务的数据库就创建完毕了.

注意： RCloudEI4.5_CreateTableSpace.sql脚本文件中BUFFERPOOL的名字、
TABLESPACE的名字和路径可以根据实际情况进行修改：

```
CREATE TABLESPACE USERSPACE2          --名字可修改  
  PAGESIZE 32K  
  MANAGED BY SYSTEM  
  USING ('d:\DB2\NODE0000\rodetbs')   --路径可修改  
  EXTENTSIZE 64  
  PREFETCHSIZE 32  
  BUFFERPOOL RODE ;                 --名字可修改  
  
CREATE system TEMPORARY TABLESPACE tmptableSPACE4 --名字可修改  
  PAGESIZE 32K  
  MANAGED BY SYSTEM  
  USING ('d:\DB2\NODE0000\systmptbs')--路径可修改
```

```
EXTENTSIZE 64  
PREFETCHSIZE 32  
BUFFERPOOL RODE ; --名字可修改
```

2.4.2 Oracle

Oracle 数据库与 DB2 数据库有所不同，Oracle 中新建数据库的系统开销比较大，这不是使用 Oracle 数据库时推荐的做法，一般情况下可以通过在一个数据库中建立多个不同的用户，并为每个用户分配表空间，为不同用户创建一套表。一般推荐使用 oracle 客户端，可以方便的创建表空间和用户，但在本手册里介绍另一种方法，无需 oracle 客户端的情况下创建 RCloud 云 EI 服务数据库。

在oracle上搭建RCloud 云EI服务数据库步骤如下：

1、在安装oracle的计算机上，运行cmd进入命令行，使用sqlplus / as sysdba 进入sqlplus。

```
C:\WINDOWS\system32\cmd.exe - sqlplus / as sysdba
Microsoft Windows [版本 5.2.3790]
(C) 版权所有 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>sqlplus / as sysdba
SQL*Plus: Release 10.2.0.1.0 - Production on 星期日 8月 17 14:06:07 2008
Copyright (c) 1982, 2005, Oracle. All rights reserved.

连接到:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> -
```

2、创建表空间，表空间文件的保存位置为oracle安装目录中数据库实例所在文件夹，表空间名称可自定义，比如如下命令：

```
create tablespace stirbp_rode datafile
'D:\oracle\product\10.2.0\oradata\orcl\sitrbp_rode.dbf' size 50M autoextend
on next 1M;
```

```
SQL*Plus: Release 10.2.0.1.0 - Production on 星期日 8月 17 14:06:07 2008
Copyright (c) 1982, 2005, Oracle. All rights reserved.

连接到:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> create tablespace stirbp_rode datafile 'D:\oracle\product\10.2.0\oradata\orcl\stirbp_rode.dbf'
2
```

3、创建用户，并且分配表空间，临时表空间取默认即可，比如如下命令：

```
create user stirbp_rode identified by stirbp_rode default tablespace stirbp_rode ;
```

4、授权（分配角色），比如如下命令：

```
grant dba to stirbp_rode ;
```

5、退出操作，比如如下命令：

```
quit;
```

6、再次登录 sqlplus，并使用刚才创建的用户登录：

```
SQL> quit;
从 Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options 断开

C:\Documents and Settings\Administrator>sqlplus
'sqlplus' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

C:\Documents and Settings\Administrator>sqlplus

SQL*Plus: Release 10.2.0.1.0 - Production on 星期日 8月 17 14:19:07 2008
Copyright (c) 1982, 2005, Oracle. All rights reserved.

请输入用户名: stirbp_rode
请输入口令:
ERROR-
```

7、按顺序执行数据库脚本：

```
@ /path/RCloudEI4.5_CreateTable.sql
@ /path/RCloudEI4.5_CreateKey.sql
@ /path/RCloudEI4.5_TableInitData.sql
Commit;
```

```
连接到:
Oracle Database 10g Enterprise Edition Release 10.2.0.1.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> @ d:\xxxxxx.sql;
```

```
WITH THE PARTITIONING, CLUSTER AND DATA MINING OPTIONS  
SQL> COMMIT;
```

经过以上的步骤RCloud 云EI服务的数据库就创建完毕了。

2.4.3 MS SQL Server

登陆 MS SQL Server 控制台，依次执行 db/ MsSqlServer2000 目录下 SQL:

RCloudEI4.5_CreateTable.sql

RCloudEI4.5_CreateKey.sql

RCloudEI4.5_TableInitData.sql

2.5 应用环境搭建

介绍在各应用服务器下 RCloud 云 EI 服务环境的搭建。

2.5.1 Websphere6.1 下环境搭建

2.5.1.1 部署管理控制台

交换服务器和连接器的部署要依赖管理控制台，所以部署 RCloud 云 EI 服务服务器时要首先部署管理控制台。部署管理控制台步骤如下：

1、在应用服务器上安装管理控制台的 war 包(推荐上下文根： /rmc；也可自定义)，其中，管控的 war 包需要从 RCloud 管理控制台的发布包中获取，需要注意的是，RCloud 管控发布包的版本需要在 4.1 以上，4.1 以下版本不支持安装管控插件，具体管控的结构以及使用，请参考 RCloud 管理控制台使用手册。

2、启动应用程序服务

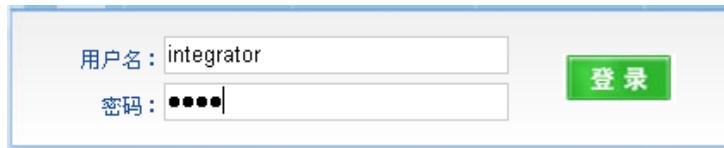


部署界面

3、安装管控插件

在部署好并且启动状态下的管控应用上，即可安装 RCloud 云 EI 服务管控插件：

(1) 使用 integrator/pass 登录管控



使用超级用户登录管控

(2) 选择左侧导航树种的“插件管理”，点击右侧的按钮“安装插件”，安装 RCloud 云 EI 服务管控插件，其中 RCloud 云 EI 服务管控插件位于 RCloud 云 EI 服务发布包中的 bin/console 中。



安装 RCloud 云 EI 服务管控插件

(3) 如果安装成功，则列表中会出现插件，同时，使用 admin/pass 登录管

控，也会出现 RCloud 云 EI 服务导航树以及相关的操作界面。



安装 RCloud 云 EI 服务管控插件后的列表以及界面

2.5.1.2 部署 RCloud 云 EI 服务服务器

在一个 RCloud 云 EI 服务总线中有至少一台 RCloud 云 EI 服务服务器，这些服务器要有逻辑上和物理上的连接才能构成一个 RCloud 云 EI 服务总线，逻辑上的连接就是路由表，物理上的连接是传输链路。在部署 RCloud 云 EI 服务服务器前首先要将 RCloud 云 EI 服务总线的路由表规划好，并在管理控制台的“RCloud 云 EI 服务总线工作区”中配置路由表。RCloud 云 EI 服务服务器部署过程如下：

一、执行产品包 db 下相应数据库的脚本创建交换服务器需要的库表。应用服务器上创建数据源，对应的数据库为执行了建库脚本的数据库。

RCloud 云 EI 服务采用了 J2EE 标准中应用服务器所提供的 JDBC 数据源获得访问数据库的能力。如果要修改数据源的相关参数，以 IBM Websphere 应用服务器 V6.1 为例讲述数据源配置过程：

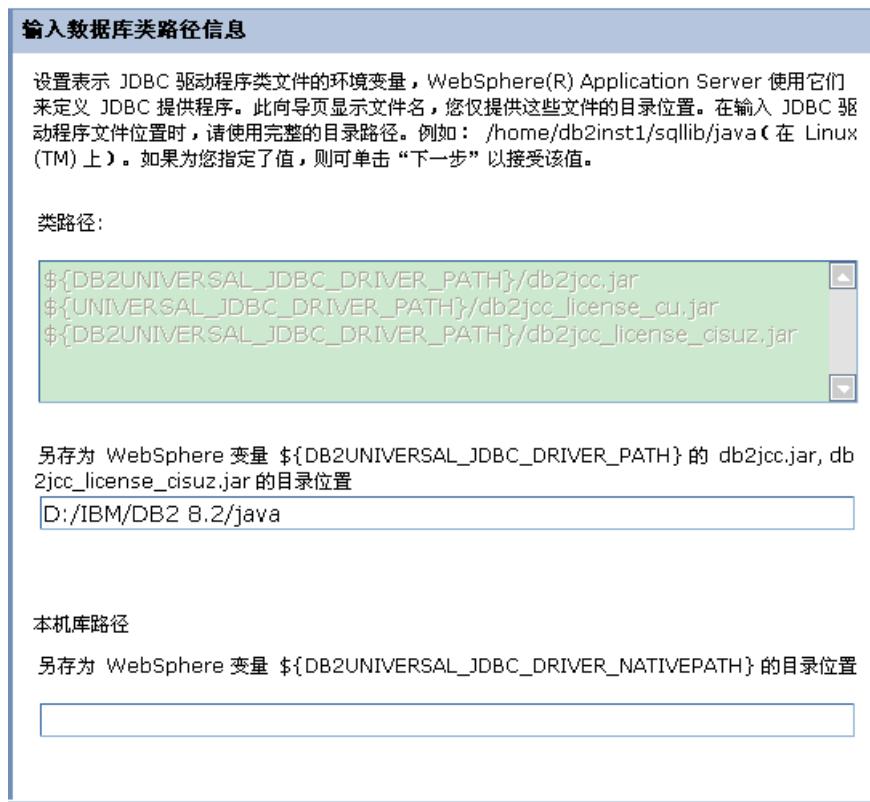
1、登录 Websphere 管理控制台在资源中选择 JDBC 提供程序。进行配置 jdbc 的 provider，如下图：



选择数据库驱动程序图

2、修改提供程序中 JDBC Driver 的路径:

其中，变量{DB2UNIVERSAL_JDBC_DRIVER_NATIVEPATH}的目录位置要填成db2安装目录下的lib目录，也可以不填。



修改 JDBC Driver 路径

3、配置和修改 J2C

数据源使用 J2C 认证的验证信息进行数据库的验证，所以如果需要修改访问数据库的相关用户和密码修改，需要修改数据源内相关的 J2C 认证参数：

[安全管理、应用程序和基础结构](#) > [JAAS – J2C 认证数据](#) > icss-9d80605a42Node01/dei115

指定供 Java(TM) 2 连接器安全性使用的一组用户标识和密码。

配置

常规属性

* 别名

icss-9d80605a42Node01/dei115

* 用户标识

db2admin

* 密码

描述

应用

确定

复位

取消

添加 J2C 认证

4、新建数据源

在配置好的 JDBC 提供程序下注册数据源(Datasource)，这里选择数据源如下图：注册结束后保存配置内容，就完成了对 JDBC 数据源的配置。

作用域

cells:icss-9d80605a42Node01Cell:nodes:icss-9d80605a42Node01:servers:server1

* 数据源名

rode

* JNDI 名称

RODE/DataSource

组件管理的认证别名 和 XA 恢复认证别名

选择组件管理的认证别名。如果 JDBC 提供程序支持 XA，则所选认证别名也将设置为 XA 恢复认证别名。如果选择 [创建新的 J2C 认证别名](#)，则此向导将被取消。

icss-9d80605a42Node01/dei115

创建 jdbc 数据源

选择 JDBC 提供程序

指定 JDBC 提供程序以支持此数据源。

创建新的 JDBC 提供程序

选择现有的 JDBC 提供程序

DB2 Universal JDBC Driver Provider



输入数据库名称

二、在应用服务器上安装 RCloud 云 EI 服务服务器 war 包(推荐上下文根: /rode; 也可自定义), 部署时需要注意:

(1)、安装的过程中要将引用资源 RODE/DataSource 映射到第一步中创建的数据源的 JNDI 名称 (即安装 RCloud 云 EI 服务服务器需要指定数据源)。并且还应该修改配置文件, 配置为使用外部数据源, 详见章节 4.1。

(2)、部署后, 如果使用 TongLINKQ 作为 EI 服务器间传输链路, 则把发布包 bin/server/lib 目录下的有关 TongLinkQ 的 jar 包 (三个: tlclient7.jar, tlqadapter.jar, tlclient.jar) 以及 connector.jar 剪贴到 websphere 安装目录下的 \AppServer\lib\下。

(3)、安装完成后将类加载顺序改为“类已装入并且是先使用应用程序类装入器”, 需要修改两处, 如下:

第一处: 点击“类装入和更新检测”, 选择下一页中的“类已装入并且是先使用应用程序类装入器”, 然后保存更改。如下图:

企业应用程序

企业应用程序 > R1DE-Iserver4_0_5-B20090211_war

使用此页面来配置企业应用程序。单击链接以打开各个页面，以便进一步配置该应用程序或它的模块。

配置

常规属性

* 名称
R1DE-Iserver4_0_5-B20090211_war

应用程序引用验证
发出警告

详细信息属性

- [特定于目标的应用程序状态](#)
- [启动行为](#)
- [应用程序二进制文件](#)
- [类装入和更新检测](#)
- [远程请求分派器属性](#)
- [安全角色到用户 / 组映射](#)
- [查看部署描述符](#)
- [最后参与者支持扩展](#)

引用

- [资源引用](#)
- [共享库引用](#)

模块

- [管理模块](#)

Web 模块属性

- [会话管理](#)
- [Web 模块的上下文根](#)
- [servlet 的初始化参数](#)
- [Web 模块的 JSP 重新装入选项](#)
- [虚拟主机](#)

按钮

应用 确定 复位 取消

企业应用程序

企业应用程序 > R1DE-Iserver4_0_8-B20090616_war > 类装入器

使用此页面来配置应用程序文件更新时重新装入类的方式。

配置

常规属性

更新应用程序文件时重新装入类

更新的文件的轮询时间间隔
1 秒

类装入器顺序

类已装入并且是先使用父类装入器

类已装入并且是先使用应用程序类装入器

WAR 类装入器策略

应用程序中每个 WAR 文件的类装入器

应用程序的单个类装入器

按钮

应用 确定 复位 取消

第二处：点击“管理模块”，然后点击模块名称链接，选择下一页中的“类已装入并且是先使用应用程序类装入器”，最后保存更改。如下图：

企业应用程序

企业应用程序 > R1DE-Iserver4_0_5-B20090211_war

使用此页面来配置企业应用程序。单击链接以打开各个页面，以便进一步配置该应用程序或它的模块。

配置

常规属性

* 名称
R1DE-Iserver4_0_5-B20090211_war

应用程序引用验证
发出警告

详细信息属性

- 特定于目标的应用程序状态
- 启动行为
- 应用程序二进制文件
- 类装入和更新检测
- 远程请求分派器属性
- 安全角色到用户 / 组映射
- 查看部署描述符
- 最后参与者支持扩展

引用

- 资源引用
- 共享库引用

应用 确定 复位 取消

企业应用程序

企业应用程序 > R1DE-Iserver4_0_5-B20090211_war > 管理模块

管理模块

指定应用程序中所包含的模块的安装目标，例如，应用程序服务器或应用程序服务器集群。可将模块安装在同一台应用程序服务器上或分散安装在几台应用程序服务器上。也可以指定某些 Web 服务器为目标，这些 Web 服务器将作为路由器来路由对此应用程序的请求。每个 Web 服务器的插件配置文件（plugin-cfg.xml）是基于通过路由的应用程序生成的。

集群和服务器：
WebSphere:cell=rdm-zanglNode01Cell,node=rdm-zanglNode01,server=server1 应用

除去 更新 除去文件 导出文件

选择	模块	URI	模块类型	服务器
<input type="checkbox"/>	RodeServer	R1DE-Iserver4_0_5-B20090211.war,WEB-INF/web.xml	Web 模块	WebSphere:cell=rdm-zanglNode01Cell,node=rdm-zanglNode01,server=server1

确定 取消



注：安装 RCloud 云 EI 服务服务器的 war 包后，第一次启动会在日志中报本地路由没有配置的错误，是正常的。用户需要在管理控制台中进行映射操作即可。

错误日志如下：

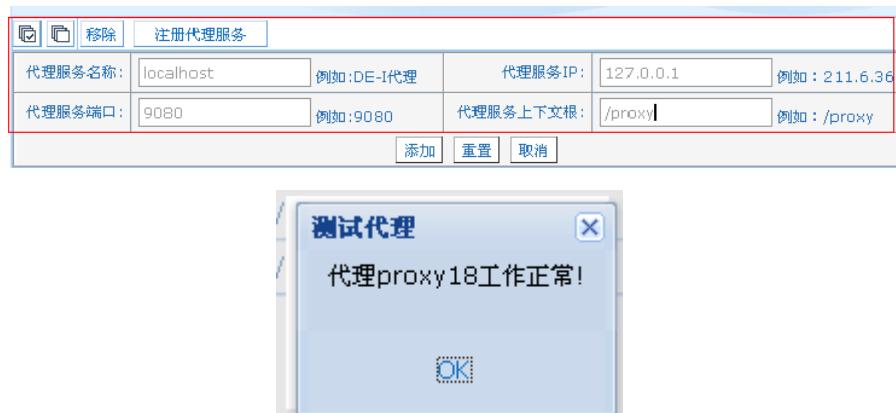
2009-12-23	10:07:45,283	ERROR	-
[com.icss.ro.de.server.service.servlet.ServiceStartupUtil]本地路由名称错误,请到管理控制台重新映射。			
2009-12-23	10:07:45,283	WARN	-
[com.icss.ro.de.server.service.servlet.ServiceStartupServlet]本地路由名称错误,请到管理控制台重新映射。			

2.5.1.3 部署 RCloud 云 EI 服务管理代理服务器

在应用服务器上安装 RCloud 管理代理服务器 war 包（推荐上下文根：/proxy；也可自定义），war 包位于 RCloud 管理控制台发布包中 bin/proxy 下，具体使用方法可以参考 RCloud 管理控制台使用手册。部署好 RCloud 管理代理，还需要在管控中，对管理代理进行注册

1) 注册代理服务

使用 octopus/pass 登录管控，在导航树种，选择“代理务管理”，即可进入到注册代理服务界面，在页面中选择“注册代理服务”，使用部署时候的受管模块信息，进行注册，注册后测试管理代理是否工作正常



注册代理服务以及测试

2) 添加受管模块

在注册后的代理服务上，选择“管理”，进入添加受管模块的页面，在页面中添加受管模块相关信息，即之前部署好的 RCloud 云 EI 服务服务器 IP、端口、上下文根信息，以及受管模块名称和 ID，受管模块的名称和 ID 可根据业务自定义。

The screenshot shows the 'Add Managed Module' interface with fields for 'Managed Module Name' (dei), 'Managed Module ID' (dei), 'Managed Module Type' (WEB service), 'IP Address' (127.0.0.1), 'Port Number' (9080), and 'Root Path' (/rode). At the bottom are 'Add', 'Reset', and 'Cancel' buttons.

添加受管模块

2.5.2 WebSphere5.1 下搭建环境

2.5.1 小节讲述的是在 IBM WebSphere 6.1 下的应用环境的搭建，本小节主要

讲述在 IBM WebSphere 5.1 下搭建环境须要注意的问题。

2.5.2.1 部署前准备

- 1、将 IBM WebSphere 5.1 目录 WebSphere\AppServer\lib 下的 qname.jar 中 javax\xml\namespace 下的 QName.class 替换成最新的(即替换成应用程序安装包中 WEB-INF\lib 下的 jaxrpc.jar 包中的\javax\xml\namespace 下的 QName.class)。
- 2、安装客户端应用程序之前需要确认客户端应用的\WEB-INF\lib 中是否存在 axiom-api-1.2.4.jar，如果客户端需要访问 web 服务，还需有 wstx-asl-3.2.1.jar。

2.5.2.2 部署管理控制台

部署管理控制台步骤如下：

- 1、在应用服务器上安装管理控制台的 war 包(推荐上下文根： /rmc；也可自定义)，完成管理控制台的部署。步骤和 WebSphere6.1 中的步骤相似。部署完后在已安装的应用程序列表中点击管理控制台应用的超链接，进入如下的界面：

常规属性		
名称	* R1ManagementConsole4_0_5-B20081217_war	① 指定应用程序的逻辑名。一个单元内的应用程序名称必须是唯一的。
启动权值	* 1	② 指定启动应用程序的顺序。低启动顺序的应用程序应该在高启动顺序的应用程序前启动。
应用程序二进制文件	* \$(APP_INSTALL_ROOT)/rdm-test18	③ 指定企业应用程序二进制文件的全路径名。路径名可以是绝对路径或可以包含路径映射变量，例如，APP_INSTALL_ROOT。
使用二进制文件中的元数据	<input type="checkbox"/>	④ 指定应用程序服务器是否将使用绑定、扩展和位于应用程序部署文档的部署描述符（deployment.xml 文件（缺省）或者是在位于应用程序的 ear 文件中那些）。
启用分布	<input checked="" type="checkbox"/>	⑤ 指定应用程序是否将自动分布到单元上的其它节点。缺省为自动分布。
类载入器方式	* PARENT_FIRST	⑥ 指定是否在装入父类载入器前，类通过父类载入器装入。
WAR 类载入器策略	* 模块	⑦ 定义应用程序中所有 WAR 共有一个单个类载入器还是每个 WAR 都有一个类载入器。
为资源创建 MBean	<input checked="" type="checkbox"/>	⑧ 为资源创建 MBean

注：其中的“类载入器方式”选择“PARENT_FIRST”。

在同一页面中，点击“相关项”中的“Web 模块”，如下图：

为资源创建 MBean	<input checked="" type="checkbox"/>	类载入器。 ① 为资源创建 MBean
启用重新装入	<input type="checkbox"/>	① 指定更新应用程序文件时，是否为它们启用类重新装入。
重新装入间隔	3	① 为更新文件扫描应用程序文件系统的时限（以秒计）。
<input type="button" value="应用"/> <input type="button" value="确定"/> <input type="button" value="复位"/> <input type="button" value="取消"/>		

其它属性

目标映射	将已部署对象（应用程序或模块）映射到目标环境（服务器、群集、群集成员）
库	指定使用全局库引用的列表。
会话管理	特定于此应用程序的会话管理器属性
提供 HTTP 端点 URL 信息	为 SOAP/HTTP 绑定提供 Web Service 端点 URL 信息
提供 JMS 和 EJB 端点 URL 信息	为 SOAP/JMS 和 EJB 绑定提供 Web Service 端点 URL 信息
发布 WSDL 文件	为启用 Web Service 的模块发布 WSDL 文件
查看部署描述符	查看部署描述符
映射 Web 模块的虚拟主机	映射 Web 模块的虚拟主机
映射模块到应用程序服务器	映射模块到应用程序服务器

相关项

Web 模块	为此应用程序定义的 Web 模块
EJB 模块	为此应用程序定义的 EJB 模块
连接器模块	为此应用程序定义的连接器模块

进入如下图中的页面，点击应用程序的超链接：

企业应用程序 > [R1ManagementConsole4_0_5-B20081217.war](#) >

Web 模块

为应用程序中每个 Web 模块创建一个 WebModuleDeployment 的实例。该实例包含部署特定信息，用于包含会话管理设置的 Web 模块。①

总计: 1

过滤器
 首选项

Uri ▾

[R1ManagementConsole4_0_5-B20081217.war](#)

进入配置界面，如下图：

企业应用程序 > [R1ManagementConsole4_0_5-B20081217.war](#) > [Web 模块](#) >

R1ManagementConsole4.0.5-B20081217.war

为应用程序中每个 Web 模块创建一个 WebModuleDeployment 的实例。该实例包含部署特定信息，用于包含会话管理设置的 Web 模块。①

配置

常规属性

Uri	* <input type="text" value="R1ManagementConsole4_0_5-B2008"/>	① 当相对于应用程序 URL 解析时，URI 指定文件系统上模块归档内容的位置。如果模块封装为部署应用程序（EAR）的一部分，则 URI 必须匹配应用程序部署描述符中 ModuleRef URI 的 URI。
备用 DD		① 给定模块的 altDD URI。
启动权值	* <input type="text" value="10000"/>	① 当企业应用程序包含多个模块时，服务器启动期间，此处指定的启动权值可用于给予此模块高于其它模块的启动优先级。具有低启动顺序的模块将首先启动。
类载入器方式	* <input type="button" value="PARENT_FIRST"/>	① 指定是否在装入父类载入器前，类通过父类载入器装入。
<input type="button" value="应用"/> <input type="button" value="确定"/> <input type="button" value="复位"/> <input type="button" value="取消"/>		

其它属性

注：其中的“类载入器方式”选择为“PARENT_FIRST”。

然后，启动应用程序，安装 RCloud 云 EI 服务管控插件，步骤和 IBM

WebSphere 6 下安装方法一致。

2.5.2.3 部署 RCloud 云 EI 服务服务器

步骤和 WebSphere6.1 中部署 RCloud 云 EI 服务的步骤相似。部署完后，将上一小节中的两个地方的类载入器方式改为“PARENT_LAST”。

部署后，如果使用 TongLINKQ 作为 EI 服务器间传输链路，则把发布包 bin/server/lib 目录下的有关 TongLinkQ 的 jar 包（三个：tlclient7.jar, tlqadapter.jar, tlclient.jar）以及 connector.jar 剪贴到 websphere 安装目录下的\AppServer\lib\下。

2.5.2.4 部署 RCloud 云 EI 服务管理代理服务器

同样地，步骤和 WebSphere6.1 中部署管理代理的步骤相似。部署完后，也要将上一小节中的两个地方的类载入器方式改为“PARENT_FIRST”。

2.5.2.5 可能出现的问题及解决方法

部署完成后启动应用程序并访问，可能会遇到的问题及解决方法如下：

1、应用启动错误，错误信息：

```
SRVE0100E: 未识别出由 servlet AxisServlet 抛出的 init() 异常：  
javax.servlet.ServletException: javax.xml.namespace.QName: method  
<init>(Ljava/lang/String;Ljava/lang/String;Ljava/lang/String;)V not found  
at  
com.ibm.ws.webcontainer.servlet.StrictServletInstance.doInit(StrictServletInstance.java:9  
9)
```

原因：IBM WebSphere 5.1 目录 WebSphere\AppServer\lib 下的 qname.jar 中 javax\xml\namespace 下的 QName 比 RCloud 云 EI 服务应用中的旧，导致 RCloud 云 EI 服务应用启动时出错。

解决方法：将应用程序安装包中 WEB-INF\lib 下的 jaxrpc.jar 包中的
\javax\xml\namespace 下的 QName.class 替换 WAS 下旧的类文件。

2、应用启动时， systemErr.log 中报错如下：

```
SystemErr R java.lang.ClassCastException:  
org.apache.xerces.jaxp.SAXParserFactoryImpl  
[08-12-5 21:17:30:219 CST] 75f4975b SystemErr R at  
javax.xml.parsers.SAXParserFactory.newInstance(Unknown Source)  
[08-12-5 21:17:30:219 CST] 75f4975b SystemErr R at  
com.ibm.ws.webcontainer.srt.TldListenerParser.<init>(TldListenerParser.java:45)
```

原因：在 RCloud 云 EI 服务应用目录中有多个 XML 解析实现类，导致加载的时候类型错误。

解决：将 RCloud 云 EI 服务 服务器安装包中的\WEB-INF\lib 中的
xmlParserAPIs-2_2_1.jar, xml-apis-1.3.03.jar, xercesImpl-2.8.1.jar 三个 jar 包删除。

3、客户端附件传输时，传输失败，错误如下：

```
org/apache/axiom/attachments/ByteArrayDataSource  
java.lang.NoClassDefFoundError:  
org/apache/axiom/attachments/ByteArrayDataSource  
at  
com.icss.ro.de.common.message.util.MessageUtil.createMessage (Unkn  
own Source)
```

原因：客户端的\WEB-INF\lib 中缺少 axiom-api-1.2.4.jar 包， axiom-api-1.2.4.jar

包可以在 RCloud 云 EI 服务服务器的安装程序 war 包中的 WEB-INF\lib 下找到。

2.5.3 金碟 Apusic 5.1 下搭建环境

本小节讲述在 Apusic 5.1 下的应用环境的搭建以及须要注意的问题。

2.5.3.1 部署前准备工作

1. Apusic 需要 JDK5 以上版本的支持，因此修改其配置文件的 java_home，可以通过 2 种方法指定，分别如下：
 - 1) 通过手动修改系统的环境变量中的 JAVA_HOME 指定
 - 2) 通过手动修改 APUSIC 的配置文件：

比如 apusic-5.1\bin\setenv.cmd 中修改：

```
rem set JAVA_HOME variable here.  
  
set JAVA_HOME=c:\Java\jdk1.5.0_16
```

应用系统使用 log4j 做日志处理，由于在同一台机器上安装有 Apusic 和 Tonglink/q，Tonglink/q 安装过程修改了系统环境变量，并且其内部包之间针对 log4j 的使用产生冲突，致使 log4j 无法创建日志文件，修改：

apusic-5.1\domains\mydomain\bin\setenv.cmd 文件，把其中的：

```
set CLASSPATH=%CP% %DOMAIN_LIB_CP% %CLASSPATH%
```

改为：

```
set CLASSPATH=%CP% %DOMAIN_LIB_CP%
```

即不加载系统的 CLASSPATH

2. 指定应用的数据源以及 WEB 管理上下文根，建议使用第 3 种方法

Apusic 部署应用方式有 2 种，一种通过 WAR 包形式部署，另一种通过文件夹形式部署

- 1) 默认上下文根是部署应用时，填写的应用名称，不用修改任何配置
- 2) 修改 apusic-5.1\domains\mydomain\config\server.xml 文件，加上 context-root 属性

```
<application name="rode"  
base="D:\deapps_for_oracle\RCloudDE-IServer4.0.5-B20090205"  
context-root="rode" start="auto"/>
```

- 3) 通过把 application.xml 以及 apusic-application.xml 拷贝到应用的 META-INF 文件夹下

application.xml:

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<!DOCTYPE application PUBLIC '-//Sun Microsystems, Inc.//DTD  
J2EE Application  
1.3//EN' 'http://java.sun.com/dtd/application_1_3.dtd'?>  
  
<application>  
  
    <display-name>default</display-name>  
  
    <description>Default Application</description>  
  
    <module>  
  
        <web>  
  
            <web-uri></web-uri>  
  
        <!--此处填写的为 WEB 管理上下文根-->  
  
            <context-root>/rode</context-root>  
  
        </web>  
  
    </module>  
  
</application>
```

apusic-application.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE apusic-application PUBLIC '-//Apusic//DTD Apusic
Application
3.0//EN'      'http://www.apusic.com/dtds/apusic-application_3_
0.dtd'>

<apusic-application>

<module uri="">

<web>

    <context-root>/rode</context-root>

<resource-ref>

<res-ref-name>rode</res-ref-name>

<!--此处填写的名字为数据源 JNDI 名字-->

<jndi-name>RODE/DataSource</jndi-name>

</resource-ref>

</web>

</module>

</apusic-application>
```

如果使用 IBM-MQ 作为 RCloud 云 EI 服务服务器间传输链路，把发布包 bin/server/lib 目录下的有关 TongLinkQ 的 jar 包(三个: tlclient7.jar, tlqadapter.jar, tlclient.jar) 复制到 Apusic 的 lib 目录中。

2.5.3.2 部署 RCloud 云 EI 服务服务器

1. 配置应用服务器数据源

(1) 启动服务器

apusic-5.1\domains\mydomain\bin\startapusic.cmd

(2) 配置数据源

访问 <http://localhost:6888/admin>

用户名/密码: admin/admin

Apusic 管理控制台界登录如下：



建立数据源，注意数据源 JNDI 一定要是 RODE/DataSource



注意，配置完数据源后需要重新启动数据源才能生效

2. 部署 RCloud 云 EI 服务服务器

在 APUSIC 管理控制台上选择“J2EE 应用”->“部署”，选择 WAR 或文件夹所在路径即可，如下图：



在部署 RCloud 云 EI 服务服务器时需要注意：

1、如果使用 IBM-MQ 作为 RCloud 云 EI 服务服务器间传输链路，把发布包 bin/server/lib 目录下的 connector.jar 复制到 RCloud 云 EI 服务服务器部署目录的 WEB-INF/lib 目录下

2、部署后，把发布包 bin/server/lib 目录下的有关 TongLinkQ 的 jar 包（三个：tlclient7.jar, tlqadapter.jar, tlclient.jar）剪贴到 Apusic 安装目录下的\lib\ext\下。

2.5.3.3 部署管理代理以及管理控制台

根据需求，在 APUSIC 管理控制台上，部署管理代理应用以及管理控制台即可，不需要指定数据源，部署后安装 RCloud 云 EI 服务管控插件以及配置代理服务方法和 IBM WebSphere 6 下方法相同。

2.5.3.4 安装部署注意事项

1. 应用间存在依赖关系的配置

在某个域中部署多个应用，如应用 A、B、C，A、B、C 之间相互有依赖关系，要某个应用先启动或者按照顺序启动；

配置方法：

- 1) 如果通过 web 管理控制台进行部署，可根据应用间的依赖顺序，如要求 A 先启动，则先部署 A。即按照系统间的依赖顺序进行部署；
- 2) 如果是采用文件方式如 war 包等直接拷贝到 domain_home\application 目录下进行部署，则需要自己调整应用的启动顺序，调整方式修改 domain_home\config\server.xml 文件样例如下：

```
<application name="default"
    base="applications/default" start="auto"/>
<application name="C" base="applications/C"
    start="auto"/>
<application name="B" base="applications/B"
    start="auto" />
<application name="A" base="applications/A"
    start="auto" />
```

根据需求如启动顺序要求的是 ABC，则调整上面的文件为样例如下

```
<application name="default"  
base="applications/default" start="auto"/>  
  
<application name="A" base="applications/A"  
start="auto" />  
  
<application name="B" base="applications/B"  
start="auto" />  
  
<application name="C" base="applications/C"  
start="auto"/>
```

2. 安装 JDK 配置

客户 jdk 安装路径为带有空格的路径下，如 c:\program files\java\jdk1.5.0，此种方式路径会影响 config.cmd 命令的使用

配置方法：

建议客户安装 JDK 为非空格路径，如 c:\java\jdk15

2.5.4 Weblogic9.2 下搭建环境

参考 Weblogic 相关应用部署文档，将 RCloud 云 EI 服务服务器、RCloud 管理控制台、RCloud 管理代理部署到 weblogic9.2 服务器上，本小节主要介绍需要注意的地方。

注意事项：

- 1) 数据源的 JNDI 名字用 RODE.DataSource。
- 2) 将 WAR 包解压后再部署，应用的上下文根在应用的 WEB-INF/weblogic.xml 中指定。
- 3) 部署后，把发布包 bin/server/lib 目录下的有关 TongLinkQ 的 jar 包（三个：tlclient7.jar, tlqadapter.jar, tlclient.jar）剪贴到到部署 DEI 服务器所在域的 lib 中（如：安装目录\user_projects\domains\[DEI 服务器所在域]\lib\）。

- 4) 如果使用 IBM-MQ 作为 EI 服务器间传输链路，把发布包 bin/server/lib 目录下的 connector.jar 复制到 EI 服务器部署目录的 WEB-INF/lib 目录下

2.5.5 安装后的形态

将 RCloud 云 EI 服务包含的各应用形态组件安装至应用服务器（例如 IBM WebSphere6.1）之后，在应用服务器的应用安装目录下会生成多个相应的文件夹，文件夹中包含了安装后各组件应用程序的所有文件。

第一次启动 RCloud 云 EI 服务应用程序和管理控制台应用程序时，在应用服务器的安装所在盘根目录自动创建名为 RCloudHome 的工作目录。 RCloudHome 是 RCloud 产品(包括 RCloud 云 EI 服务)集中存放配置文件和日志信息的目录， RCloud 云 EI 服务产品的应用程序启动时会生成相应目录：DataExchange， 用户配置信息将会保存到相应文件中。删除或随意修改将导致 RCloud 云 EI 服务产品不可使用，请慎重操作。该目录结构如下：（管控、管理代理和 RCloud 云 EI 服务服务器使用相同的 RCloudHomg 目录名称。其中有些文件夹是在这些应用启动后的使用过程中生成的，如 logs、temp）

```
└── RCloudHome
    ├── config
    │   ├── Log.xml                                //系统日志配置文件
    │   └── SystemConfig.xml                      //系统的各种配置信息
    ├── DataExchange
    │   ├── RodeClientConfig.xml      //配置在本机器上的所有连接器的信息
    │   └── forwardpolicy
    │       ├── Forward_Policy.xml.sample        //转发策略示例文件
    │       └── msg
    │           ├── 53                         // 路由名称或连接器 ID [53 仅为示例]
    │           │   ├── rec_5304//从 5304 接收到的消息的缓冲池[5304 为示例]
    │           │   ├── to_5304                  //发往 5304 的消息的发送缓冲池
    │           │   ├── err_rec_5304            //从 5304 接收消息的处理容错池
    │           │   └── err_to_5304             //发往 5304 时发送失败时的容错池
    │           └── ServiceCompoment          //底层服务组件注册配置 EI 扩展使用
    └── temp //存放消息交换过程中的临时文件，谨慎删除其中的文件
        ├── Deskey 、 Privatekey、 Publickey      //安全电子证书
        ├── RODEConfig.xml                        // RCloud 云 EI 服务交换服务器配置文件
        └── DENodeRoute.xml                      //全局路由配置文件
```

```
| -InfobusConfig.xml //兼容 RCloudInfous 方式运行时需要的配置文件
|-Licence
| |-Rone.lic //用户使用权限许可证书
|-logs
| |-DE-I //EI 默认的日志存放位置
|-ManagementConsole //管理控制台框架显示配置文件（不可修改）
| |-DataExchange //管控中针对 EI 管理的配置文件位置
| | |-ServerAliaseLogHome //服务器别名配置广播日志
| | |-ClientTypeConfig.xml //禁止更改
| | |-DENodeRoute.xml //全局路由配置，禁止手动更改
|-temp
|-upload
|-Repository //EI 管控插件包存放的物理位置，禁止更改
```

其中，RCloud 云 EI 服务的配置信息和日志信息主要放在 RCloudHome 目录下的 DataExchange、logs 和 ManagementConsole 下面的 DataExchange 文件夹下。

(1) DataExchange 文件夹中存放的是有关 RCloud 云 EI 服务的全部配置信息，其中：

msg 文件夹下存放的是配置在本 server 上的所有连接器及当前服务器的消息池；

DENodeRoute.xml 文件中保存的全局路由配置信息；

RodeClientConfig.xml 文件中保存在本台机器上配置的连接器信息；

InfobusConfig.xml 文件是在兼容 RCloudInfobus 运行时的配置信息，主要是适配器信息。

ServiceProvider.xml 文件保存配置在本 server 上所有的集成对象信息。

(2) Logs 文件夹中存放的是所有的日志信息，其中：

logs\DE-I\deserver 下存放的是 EI 服务器产生的日志；

logs\MC 下存放的是管控产生的日志；

在本 server 中配置的连接器所产生的日志被放在 logs\DE-I 下。

(3) ManagementConsole\DataExchange 文件夹下是 RCloud 云 EI 服务在管控中的配置信息，其中：

ClientTypeConfig.xml 文件中试所有的连接器类型，禁止更改；

DENodeRoute.xml 管控中配置生成的全局路由拓扑配置文件；

2.6 连接器部署配置

连接器是数据交换节点消息接入器。集成对象通过连接器与 RCloud 云 EI 服务服务器交换消息。目前，有以下几种连接器：DE Agent 连接器、IBM-MQ 连接器、SOAP/WS 连接器、HTTP 连接器、FTP 连接器、文件连接器、TongLinkQ 连接器。所有的连接器需要在管理控制台注册，并可以在管理控制台中进行配置信息管理和状态监控管理。

2.6.1 DE Agent 的配置

DEAgent 连接器部署在客户应用中，是客户应用与 RCloud 云 EI 服务总线交互的客户端代理，通过 RCloud 云 EI 服务提供的 SDK 实现与 DE-I 总线的服务/消息交互，因此在使用 DEAgent 连接器时，首先需要将 DEAgent 部署到客户端应用中。

2.6.1.1 客户端应用配置

在客户端应用中部署 DEAgent 步骤如下：

1. 客户端应用开发环境配置与开发

客户应用与 RCloud 云 EI 服务总线交互，是通过 RCloud 云 EI 服务提供的 SDK 实现与 DE-I 总线的服务/消息交互，因此，在开发客户端应用时，需要准备相应的开发环境，用于在应用中调用 RCloud 云 EI 服务提供的 SDK 发送和接收消息的处理。开发环境准备参考《RCloud4.5Exchange-Integration 产品用户开发手册》中 2.1.3.3 章节说明。

当开发环境配置完成后，调用 RCloud 云 EI 服务提供的 SDK 处理相应的消息发送和接收逻辑。参考《RCloud4.5Exchange-Integration 产品用户开发手册》中 2.1.3.1 和 2.1.3.2 章节说明

2. 客户端 RCloudHome 目录的创建

将发布包中客户端部署压缩包中 RCloudHome 文件夹下的内容全部拷贝到应用服务器所在盘(Windows 系统)或 root 根目录(UNIX 系统)下的 RCloudHome 目录下。将客户端部署压缩包中 RCloudHome/DataExchange 下的 rone.lic 文件，拷到 RCloudHome/license 下。

注意：如果您想指定 RCloudHome 的路径到别的位置，而不使用默认目录，那么就需要进行如下步骤：

1) 打开客户端应用/WEB-INF/classes 目录下的 rode.properties，修改其中的 RCloudHome 配置参数，来指定 RCloudHome 目录路径。
(/RCloudHome 表示应用服务器所在盘的根目录下的 RCloudHome 目录)

2) 注意：rode.properties 中的 clientID 不能手动修改，这个参数是给部署的应用客户端定义“身份”。

3. 在管理控制台“总线工作区”->“配置集成对象”中添加该集成对象（参考 3.3.3.3）。

4. 为该集成对象配置 DEAgent 连接器：

管理控制台“总线工作区”->“配置连接器”，填写连接器配置的基本信息并添加连接器到 RCloud 云 EI 服务服务器和 RCloud 云 EI 服务服务器到连接器的传输方式（参考 [3.3.3.3](#)）。

5. 回到管理控制台“总线工作区”->“配置连接器”，双击刚注册的集成对象，右侧出现刚添加的连接器，选择连接器点击“推送”。（参考 3.3.3.5）

6. 在应用服务器中重新启动连接器所在的 WEB 应用。

7. 管理控制台“总线工作区”->“配置连接器”，启动该连接器。（参考 [3.3.3.4](#)）

注意：

1、当连接器的信息和所在 RCloud 云 EI 服务服务器的路由信息发生变化时，要重新做推送操作，并且重启 RCloud 云 EI 服务服务器和连接器所在的客户端应用。

2、连接器所在的应用重启后，如果应用停止前连接器的状态是停止的，管控中显示的连接器状态为停止，要到管理控制台中启动应用客户端连接器才能

使连接器正常工作。如果应用停止前连接器状态是运行的，那么应用重启时，连接器会随应用一起启动。

3、如果在客户端应用未将 rode.properties 文件打包到客户端应用 WAR 包中的/WEB-INF/classes 目录下，rode.properties 文件需要手工复制到客户端应用的/WEB-INF/classes 目录下。

2.6.1.2 消息处理器配置

部署消息处理器步骤如下：

1、开发消息处理器并和应用一起打包。

消息处理器的开发，请参考《RCloud4.5Exchange-Integration 产品用户开发手册》中相关部分的说明。

2、管理控制台中相应的连接器下注册该消息处理器。

2.6.2 SOAP/WS 连接器配置

SOAP/WS 连接器不需要部署，直接在管控中配置。详细说明参见第三章中 RCloud 云 EI 服务连接器配置。

SOAP/WS 接入开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.3 HTTP 连接器配置

HTTP 连接器不需要部署，直接在管控中配置。详细说明参见第三章中 RCloud 云 EI 服务连接器配置。

HTTP 接入开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.4 IBM-MQ 连接器配置

IBM-MQ 连接器使用的前提是当前应用是基于 IBM MQ 的应用，并已经在
中软国际有限公司

当前应用环境中部署了 IBM Websphere MQ。IBM Websphere MQ 的部署与使用请参考 IBM Websphere MQ 相关文档。

IBM-MQ 连接器本身不需要部署，直接在管控中配置。详细说明参见第三章中 RCloud 云 EI 服务连接器配置。

IBM-MQ 接入开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.5 FTP 连接器

配置详细说明参见第三章中 FTP 连接器配置。

FTP 接入开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.6 文件连接器

文件连接器为一种特殊的 DE Agent 连接器，部署请参考 DE Agent 连接器部署说明，配置请参照第三章 文件连接器配置说明。

文件接入开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.7 TongLINKQ 连接器配置

TongLINKQ 连接器使用的前提是当前应用是基于 TongLINKQ 的应用，并已经在当前应用环境中部署了 TongLINKQ。TongLINKQ 的部署与使用请参考 TongLINKQ 的相关文档。

TongLINKQ 连接器本身不需要部署，直接在管控中配置。详细说明参见第三章中 RCloud 云 EI 服务连接器配置。

TongLINKQ 接入开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.8 Win32Agent 连接器配置

Win32Agent 连接器在管控中的配置详细说明请参见第三章 RCloud 云 EI 服务连接器配置。

Win32Agent 连接器的部署和开发开发请参考《RCloud4.5Exchange-Integration 产品用户开发手册》(开发指南)。

2.6.9 数据库 Agent 触发器

数据库 Agent 触发器是 RCloud 云 EI 服务提供的一个附加功能，用于帮助开发商将 DB2 数据库表中的数据通过触发器的机制即时生成 RCloud 云 EI 服务的消息，发送至相应 RCloud 云 EI 服务服务器完成数据交换。

注意：RCloud 云 EI 服务关注应用中提供的数据或服务间的交互与处理，不关注数据的提取，从应用中或数据源提取数据请考虑使用 ETL 工具或数据服务实现，也可以通过应用开发来实现。

2.6.9.1 DB2 数据库下 Agent 触发器的手动部署

一、环境变量设置

首先，操作系统的 Java 环境变量：JAVA_HOME、PATH 必须设置正确。当在 linux 下部署时，要注意一定安装使用 IBM Developer Kit(IBM 的 JDK)，并进行以下操作：

在 /usr/lib 中创建符号链接以指向 Java 共享库。根据要使用的 JDK 版本，将有指向不同共享库的链接：

1、对于 IBM Developer Kit 1.3 创建指向 libjava.so、libjvm.so 和 libhpi.so 的符号链接。可通过作为 root 用户运行下列命令来创建符号链接：

```
cd /usr/lib
ln -fs JAVAHOME/jre/bin/libjava.so .
ln -fs JAVAHOME/jre/bin/classic/libjvm.so .
```

```
ln -fs JAVAHOME/jre/bin/libhpi.so .
```

其中 JAVAHOME 是 IBM Developer Kit 的基本目录。如果 DB2 UDB 找不到这些库，在尝试运行 Java 例程时将产生 -4301 错误，并且管理通知日志中将出现有关找不到库的消息。

2、对于 IBM Developer Kit 1.4.1 创建指向 libjava.so、libjvm.so、libhpi.so 和 libjsig.so 的符号链接。可通过作为 root 用户运行下列命令来创建符号链接：

```
cd /usr/lib

ln -fs JAVAHOME/jre/bin/libjava.so

ln -fs JAVAHOME/jre/bin/classic/libjvm.so

ln -fs JAVAHOME/jre/bin/libhpi.so

ln -fs JAVAHOME/jre/bin/libjsig.so
```

其中 JAVAHOME 是 IBM Developer Kit 的基本目录。如果 DB2 UDB 找不到这些库，在尝试运行 Java 例程时将产生 -4301 错误，并且管理通知日志中将出现有关找不到库的消息。

3、对于 IBM Developer Kit 1.4.2 创建指向 libjava.so、libjvm.so、libhpi.so、libjsig.so、libjitc.so、libxhpi.so 和 libdbgmalloc.so 的符号链接。可通过作为 root 用户运行下列命令来创建符号链接：

```
cd /usr/lib

ln -fs JAVAHOME/jre/bin/libjava.so

ln -fs JAVAHOME/jre/bin/classic/libjvm.so

ln -fs JAVAHOME/jre/bin/libhpi.so

ln -fs JAVAHOME/jre/bin/libjsig.so

ln -fs JAVAHOME/jre/bin/libjitc.so

ln -fs JAVAHOME/jre/bin/libxhpi.so

ln -fs JAVAHOME/jre/bin/libdbgmalloc.so
```

其中 JAVAHOME 是 IBM Developer Kit 的基本目录。如果 DB2 UDB 找不到这些库，在尝试运行 Java 例程时将产生 -4301 错误，并且管理通知日志中将出现有关找不到库的消息。

二、数据环境变量

1、Windows

1) 命令行键入： db2cmd。

2) 在 db2 clp 内键入： db2 update dbm cfg using JAVA_HEAP_SZ 8192，
8192 是建议大小，如果数据量大，可将 JAVA_HEAP_SZ 设的更些。

3) 将应用客户端部署压缩包中 lib 和 3rdlib 下的所有 jar 文件解压到硬盘上，打开应用客户端部署压缩包中 classes 目录下的 rode.properties，修改其中的 2 个配置参数：

a) resourceOneHome: RCloudHome 所在目录全路径，默认为 /RCloudHome。

b) clientID: 应用客户端的 ID，就是在管理控制台中注册的应用客户端信息中的 clientID，通过这个参数定义客户端“身份”。

4) 用应用客户端部署压缩包中 RCloudHome 覆盖上一步配置的 RCloudHome 目录，并修改配置文件要修改的参数。另外，与 WEB 客户端不同的是：管理控制台不能向客户端触发器推送信息，只能手动修改文件：

RCloudHome/ DataExchange/RodeClientConfig.xml。

管理控制台“连接器维护”选择应用客户端，点击“目录信息”会在下面文本域里生成该客户端的目录信息，将文本域中的内容拷贝到 RodeClientConfig.xml 中<GlobalRodeClient> </GlobalRodeClient>之间。

5) 在 db2clp 内键入：

```
call  
sqlj.install_jar('file:///JARFORDER/activation.jar',  
'activation'),
```

将 activation.jar 导入到 DB2 环境中。JARFORDER 是 lib 和 3rdlib 下的所有 jar 文件解压到硬盘上全路径名称。

6) 重复持续 call sqlj.install_jar，将 activation.jar 换成其他 jar 文件， activation 也换成相应 jar 的名称，直到将 lib 和 3rdlib 里所有的 jar 文件都导入到 DB2 环境

2、Linux

1) 以 db2 数据库实例拥有者登录。

2) 键入:

```
db2 update dbm cfg using JAVA_HEAP_SZ 8192
```

8192 是建议大小, 如果数据量大, 可将 JAVA_HEAP_SZ 设的更大些。

3) 将应用客户端部署压缩包中 lib 和 3rdlib 下的所有 jar 文件解压到硬盘上, 打开应用客户端部署压缩包中 classes 目录下的 rode.properties, 修改其中的 2 个配置参数:

a) resourceOneHome: RCloudHome 所在目录全路径, 默认为 /RCloudHome/AgentHome。

b) clientID: 连接器的 ID, 就是在管理控制台中注册的连接器信息中的 clientID, 通过这个参数给部署的连接器定义“身份”。

4) 用应用客户端部署压缩包中 RCloudHome 覆盖上一步配置的 RCloudHome 目录, 并修改配置文件要修改的参数。另外, 与 WEB 客户端不同的是: 管理控制台不能向客户端触发器推送信息, 只能手动修改文件: RCloudHome/ DataExchange/RodeClientConfig.xml。管理控制台“连接器维护”选择连接器, 点击“目录信息”会在下面文本域里生成该连接器的目录信息, 将文本域中的内容拷贝到 RodeClientConfig.xml 中<GlobalRodeClient></GlobalRodeClient>之间。

5) 命令行键入:

```
call sqlj.install_jar('file:/JARFORDER/activation.jar', 'activation')
```

将 activation.jar 导入到 DB2 环境中。JARFORDER 是 lib 和 3rdlib 下的所有 jar 文件解压到硬盘上全路径名称。

6) 重复持续 `call sqlj.install_jar`, 将 activation.jar 换成其他 jar 文件, activation 也换成相应 jar 的名称, 直到将 lib 和 3rdlib 里所有的 jar 文件都导入到 DB2 环境中。

三、创建 Java UDF

1、Windows

命令行键入：db2cmd。在 db2 clp 内键入：

```
create function sendToRCloudDE(clientShortName varchar(150) , key  
varchar(50) , data varchar(1000) ) returns int fenced variant no  
sql external action language java parameter style java external name  
'com.icss.ro.de.agent.trigger.RCloudDEDBTrigger!send'
```

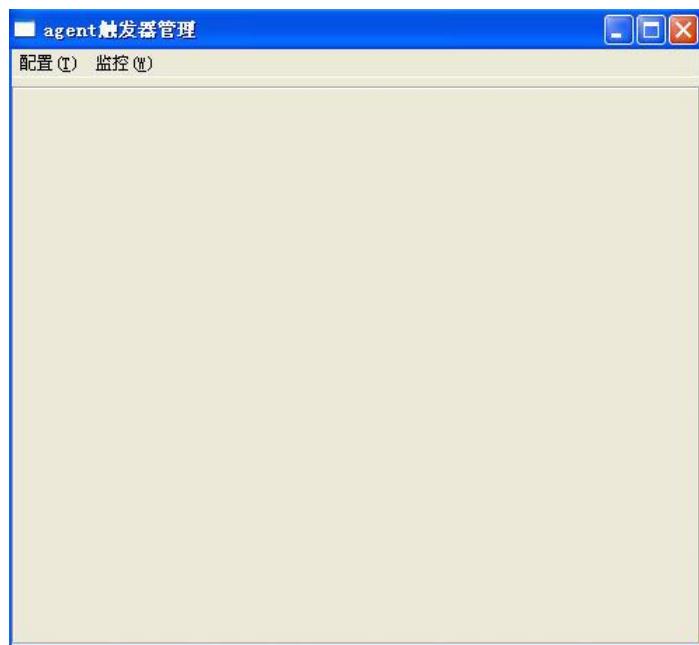
2、Linux

以 db2 数据库实例拥有者登录。键入：

```
create function sendToRCloudDE(clientShortName varchar(150) , key  
varchar(50) , data varchar(1000) ) returns int fenced variant no  
sql external action language java parameter style java external name  
'com.icss.ro.de.agent.trigger.RCloudDEDBTrigger!send'
```

2.6.9.2 DB2 下自动部署连接器服务

触发器的部署可以使用部署工具通过界面完成，并可以通过此工具来查看和管理触发消息内容。在发布包中找到 ConfigTrigger 压缩包并解压，由于需要 RCloudHome 下配置文件支持，所以默认情况下是/RCloudHome/AgentHome(如果运行程序 run.bat 与 RCloudHome 不在相同盘符下，打开 ConfigTrigger.jar 下的 rode.properties 可以修改此配置)。运行文件 run.bat 就可以看到如下界面，按着如下步骤对触发器进行配置：



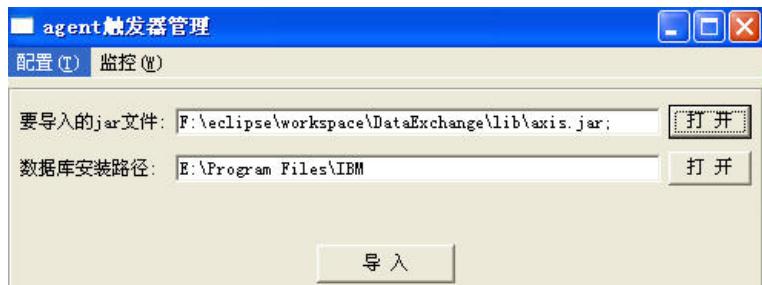
Agent 触发器管理界面

1、要对数据库进行配置，选择菜单配置→数据库连接配置。当前触发器只能对 DB2 数据库支持。填写驱动、连接名、用户名和密码。单击**测试**按钮，测试数据库配置是否成功，单击**保存**可以把此配置保存起来。如果数据库连接有更改，要在此重新保存。



创建 agent 触发器界面

2、要对触发器函数(处理消息)进行库文件导入，选择菜单配置→jar 文件导入到...进到文件导入界面。此步骤目的是把处理触发器消息的相关的 jar 包配置到正确的路径下。在“要导入的 jar 文件”处点击**打开**，选中发布包中的 lib 文件夹下所有 jar 文件，在“数据库安装路径”处点击**打开**，选择 db2 的安装路径 IBM 文件目录，然后点击**导入**按钮即可。



对触发器函数进行库文件导入界面

3、注册 java UDF，此功能是根据你配置的数据库来对触发器进行注册。选择菜单配置—> UDF 注册到...进到注册界面。确定数据库连接正确配置的情况下。点击注册，对话框提示注册成功即可。



注册 javaUDF 界面

4、触发器的创建，此功能是通过文件来创建触发器语句。选择菜单配置—> 触发器创建到...进到触发器界面。此触发器语句放在。sql 文件中，可以支持多条触发器语句共同创建(每个触发器语句之间以";"分号隔开)。点击触发器语句后 打开按钮，选择存放的触发器语句文件，在确定数据库连接正确配置的情况下。点击创建，对话框提示创建成功即可。



创建触发器界面

对数据库 agent 触发器的连接器管理和监控。此管理方式是通过 socket 通信来实现的，以对触发器消息进行管理和监控(管理控制台也可以对此进行监控管理)。此时服务器状态有三种形式，未知状态、运行状态和停止状态。所谓的运行和停止逻辑是指触发器是否能够发送消息。点击启动和停止按钮可以对触发器进行启、停。

2.7 RCloud 云 EI 服务服务器配置

RCloud 云 EI 服务的所有配置都在

RCloudHome/DataExchange/RODEConfig.xml 中。可以直接修改 RODEConfig.xml 中的配置参数，修改后要重新启动相应的 RCloud 云 EI 服务服务器，使配置生效。也可以通过管理控制台配置，修改之后即时生效，不用重启相应的服务器。RODEConfig.xml 中的配置项有：传输、日志和 Security 配置。

2.7.1 传输

```
<DETtransport>

    <MsgPoolPath>/RCloudHome/DataExchange/msg</MsgPoolPath>

    <UseMsgPool>on</UseMsgPool><!--发送消息是否使用消息池，默认使用-->

    <MsgPoolListenerInterval>30</MsgPoolListenerInterval>

    <MQListenerInterval>30</MQListenerInterval>

    <FTPLListenerInterval>30</FTPLListenerInterval><!-- ftp 文件侦听间隔 -->

    <MQTransMaxSize>4096</MQTransMaxSize><!--mq 传输允许的消息最大值(单位: K)-->
```

```
<UploadFileTemp>/RCloudHome/DataExchange/upload</Upload
FileTemp><!--消息带附件传递时使用-->

<HTTPConnectionTimeout>120</HTTPConnectionTimeout><!--
消息带附件传递时使用-->

<AttachmentMaxSize>4096</AttachmentMaxSize><!--消息
带附件传递时使用(单位: K)-->

<UseFileSeg>1</UseFileSeg> <!--"1":允许对大消息进行切片
(默认值) ; "0":不允许-->

<FileSegSize>512</FileSegSize> <!--单位(K), 消息体超过
512K 对消息进行切片, 每片大小为 512K。-->

<MQ-SVRCONN>RONEDE.SVRCONN</MQ-SVRCONN><!--EI 连接 MQ 服务器使用
的服务器连接通道名称-->

<NeedErrorNotify>false</NeedErrorNotify><!--匹配转发策略成功
后发送的消息是否需要异常反馈。true||false-->

<DefaultMaxConnectionsPerHost>256</DefaultMaxConnectionsPe
rHost><!--指定 URL 最大连接数-->

</DETransport>
```

■ MsgPoolPath

消息池的全路径名称。

■ MsgPoolListenerInterval

消息池的侦听器侦听间隔, 单位: 秒。值不能大于 2147483。

■ UseMsgPool

消息池开关“on”表示使用消息池, “off”表示不用消息池。一般情况下建
议设为“on”, 保证消息不丢失。如果该项没有设值, 默认为“on”。

■ MQListenerInterval

RCloudDE-I 侦听器侦听 IBM Websphere MQ 和 TongLINKQ 队列时间间

隔。单位：秒，当有 IBM MQ/TongLINKQ 连接器或 RCloud 云 EI 服务服务器间采用 IBM MQ/TongLINKQ 传输方式时，需要设置这个参数。队列中消息频繁，间隔设置小一些，反之设置要大一些。

■ FTPListenerInterval

RCloudDE-I 倾听器倾听 FTP 目录的时间间隔。单位：秒，若客户端和服务器使用 FTP 传输方式时，需要设置这个参数。FTP 目录中消息频繁，间隔设置小一些，反之设置要大一些。

■ MQTransMaxSize

IBM Websphere MQ 传输队列允许传输的最大消息（单位：K），默认为 4096K。

■ UploadFileTemp

附件临时保存目录

■ AttachmentMaxSize

允许传递的附件最大值（单位：K），默认为 4096K

■ UseFileSeg

是否开启消息切片功能。1：开启；0：关闭

■ FileSegSize

切片消息大小（单位：K），当消息大于该值时，如果走 MQ 通道，并且切片功能开启时，系统会将消息切片，每个切片后的消息大小等于该值，默认值为 512K。

■ MQ-SVRCONN

RCloud 云 EI 服务连接 IBM MQ 服务器时使用的服务器连接通道名称

■ NeedErrorNotify

转发消息异常是否需要通知发送端。

TRUE：异常反馈

FALSE：异常不反馈（默认值）

■ DefaultMaxConnectionsPerHost

HTTP 客户端请求主机的最大连接数，该项用于消息流中有服务调用节点时使用，如果频繁的调用外部服务，则此项设置的需要大一些

2.7.2 自动测试链路状态

```
<AutoTestRoute>

    <AutoTestRouteEnable>on</AutoTestRouteEnable>

    <!--自动测试路由开关 -->

    <AutoTestRouteInterval>10</AutoTestRouteInterval>

    <!--单位：分钟；发送消息线程轮循间隔时间 默认为 10 分钟-->

</AutoTestRoute>
```

■ AutoTestRouteEnable

自动测通路由开关“on”表示开启自动测通功能，“off”表示不开启。如果该项没有设值，默认为“on”。

■ AutoTestRouteInterval

发送消息线程轮循时间间隔。单位：分钟，自动测通路由线程会根据配置的时间间隔，向与它相邻并且配置了链路的服务器发送测试消息，来测试链路是否畅通。默认时间间隔为 10 分钟。

2.7.3 日志

```
<DELogConfig>

    <Console>

        <Level>info</Level>

        <ConversionPattern>

            %d%-5p%m%n

        </ConversionPattern>

    </Console>

    <File>

        <Level>debug</Level>

        <LogFolder>

            /RCloudHome/logs/DE-I

        </LogFolder>

    </File>

</DELogConfig>
```

```
</LogFolder>

    <MaxLogSize>1024</MaxLogSize>

    <MaxSizeRollBackups>10</MaxSizeRollBackups>

        <ConversionPattern>

            %d%-5p %m%n

        </ConversionPattern>

    </File>

</DELogConfig>
```

RCloud 云 EI 服务的日志会输出到 2 个地方：JVM 控制台和日志文件中。

控制台日志在<Console></Console>之间配置，文件日志在<File></File>之间配置。配置项含义相同。

1、Level

Level 是日志记录的优先级，分为 FATAL、ERROR、WARN、INFO、DEBUG、TRACE。

FATAL: 指出现了非常严重的错误事件，这些错误可能会导致应用程序异常中止；

ERROR: 指虽有错误，但仍允许应用程序继续运行；

WARN: 指运行环境潜藏着危害；

INFO: 指报告信息，这些信息在粗粒度级别上突出显示应用程序的进程；建议在实时系统中设置该级别；

DEBUG: 指细粒度信息事件，细粒度信息事件对于应用程序的调试是最有用的。

TRACE: 比 DEBUG 级别更加细粒度信息事件，该级别打印只限于调试期间，生产环境中禁止开启，否则性能会有损耗

2、ConversionPattern

日志布局方案，设置同 log4j，修改日志布局可参考：

<http://logging.apache.org/log4j/docs/api/org/apache/log4j/PatternLayout.html>

3、LogFolder

日志文件的存放路径。

4、MaxLogSize

日志文件最大值，单位：KB。

5、MaxSizeRollBackups

当日志文件大小超过 MaxLogSize 的值时会创建新的日志文件，新创建文件大小超过 MaxLogSize 时会再创建新的日志文件，如此反复。

MaxSizeRollBackups 是设置日志文件的最大数目，当文件数达到 MaxSizeRollBackups 时，会将时间最早的日志文件删除。

2.7.4 RODEConfig.xml 的 Security 配置

该 Security 配置包括了 RCloud 云 EI 服务中消息加密/解密的实现、RCloud 云 EI 服务管理与接入的认证实现，该实现可以通过配置使用第三方提供的安全实现。

在 RODEConfig.xml 中的<Security>标签中，可以配置 key 的路径和文件。

在发布包中，默认在 RCloudHome/DataExchange 目录下。程序会首先查找 RODEConfig.xml 是否配置了 key 的路径和文件，如果没有配置，就在 RCloudHome/DataExchange 中间查找，所以，一般不用在 RODEConfig.xml 中重新配置 key 的路径和文件。

2.7.5 消息转发策略配置

消息转发策略的配置并不是必须的，当 EI 服务器接收到的消息需要进行转发的时候，才进行转发策略的配置，否则，可以不配置消息转发策略。

转发策略的配置主要是在一个 EI 服务器中的 \RCloudHome\DataExchange\forwardpolicy 中配置 Forward_Policy.xml 文件，用来在消息到达此 EI 服务器上时匹配转发到其他的 EI 服务器或连接器。对消息转发策略的配置，在管控中提供了可视化的 UI（见 [3.3.4.9.4](#)）。

2.7.5.1 策略文件示例

```
<?xml version="1.0" encoding="UTF-8"?>

<fw:ForwardPolicy

    xmlns:fw="forwardpolicy.common.de.ro.icss.com"

    xmlns:tns="destination.message.common.de.ro.icss.com"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="forwardpolicy.common.de.ro.icss.com

        Forward_Policy.xsd

        destination.message.common.de.ro.icss.com

        DestinationSet.xsd ">

    <fw:MatchConditions

        Expression="n1&amp;#38;n2&amp;#38;n3&amp;#38;n4">

        <fw:Condition Name="n1" Type="MsgBZType">

            <fw:MsgBZType Value="a"/>

        </fw:Condition>

        <fw:Condition Name="n2" Type="Content">

            <fw:Content ReadFrom="/root/t01" Value="a"/>

        </fw:Condition>

        <fw:Condition Name="n3" Type="Source">

            <fw:DestinationSet>

                <tns:Include>

                    <tns:Address>

                        <tns:Server Exact="true" Type="RouteName"

Value="s1"/>

                        <tns:Client Exact="true" Type="ClientID"

Value="c1"/>

                    <tns:Processor Exact="true" Type="ID"
```

```
Value="" />

    </tns:Address>

    </tns:Include>

    <tns:Exclude>

        .....

    </tns:Exclude>

    </fw:DestinationSet>

    </fw:Condition>

    <fw:Condition Name="n4" Type="DestinationSet">

        <fw:DestinationSet>

            .....

        </fw:DestinationSet>

    </fw:Condition>

    </fw:MatchConditions>

    <fw:ForwardAddresses>

        <tns:Include>

            <tns:Address>

                <tns:Server Exact="true" Type="RouteName"
Value="test" />

                <tns:Client Exact="true" Type="ClientID" Value="" />

                <tns:Processor Exact="true" Type="ID" Value="" />

            </tns:Address>

        </tns:Include>

        <tns:Exclude>

            <tns:Address>

                <tns:Server Exact="true" Type="RouteName" Value="" />

                <tns:Client Exact="true" Type="ClientID" Value="" />

                <tns:Processor Exact="true" Type="ID" Value="" />

            </tns:Address>

        </tns:Exclude>

    </fw:ForwardAddresses>

</fw:Condition>
```

```
</tns:Address>  
</tns:Exclude>  
</fw:ForwardAddresses>  
</fw:ForwardPolicy>
```

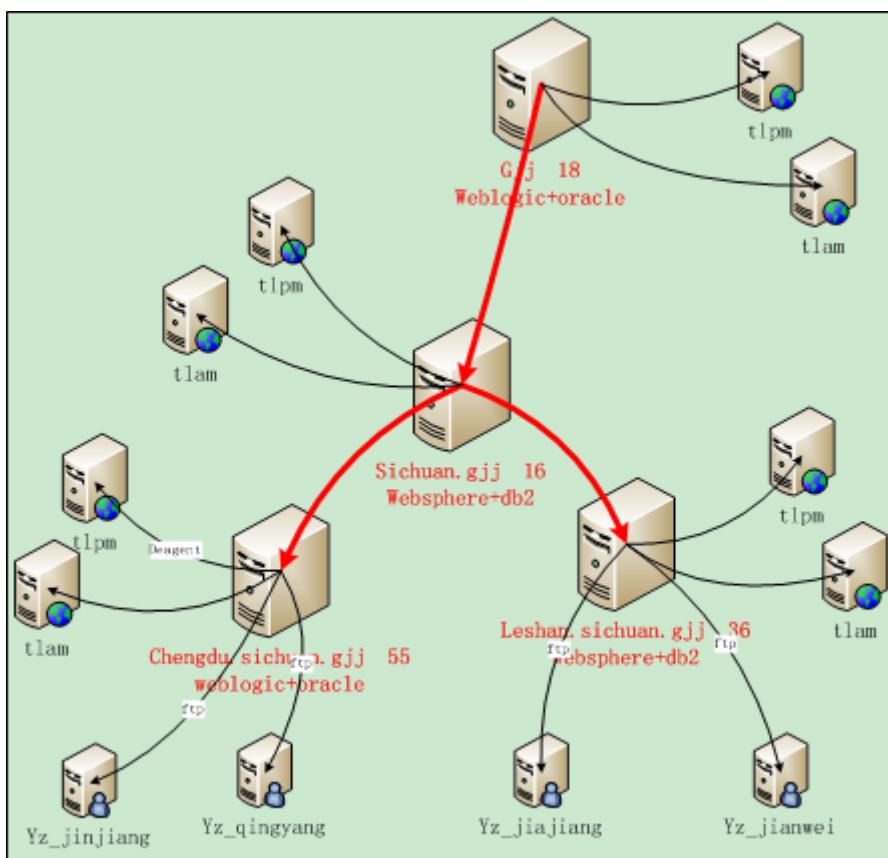
2.7.5.2 策略文件元素说明

元素名	描述	属性	描述
ForwardPolicy	转发策略文件根元素，包含两个子元素 MatchConditions, ForwardAddresses		
MatchConditions	匹配条件集合，包含子元素 Condition	Expression	条件表达式，当元素 Condition 中条件都满足时，返回 TRUE，表达式中使用 Condition 元素中 name 属性的值
Condition	匹配条件，子元素为 【MsgBZType, Content, DestinationSet】中一个，根据属性 Type 选择	Name	条件的名称
		Type	条将类型，有四种类型 【MsgBZType, Content, Source, DestinationSet】
MsgBZType	表示根据业务类型匹配，如果消息头中指定的业务类型和 value 中的相同，返回 true	Value	匹配目标值
Content	表示根据内容匹配，如果消息中指定位置的值和 value 中的相同，返回 true	ReadFrom	匹配源在消息中的 XPATH
		Value	匹配目标值
DestinationSet	地址（源地址、目标地址）Type 类型如果是 【Source, DestinationSet】时，会将消息头中对应的源地址或目标地址和这里的匹配，如果包含，返回 TRUE。		

	它包含两个子元素 【Include, Exclude】		
Include	地址集合，子元素 【Address】描述地址的详细信息		
Address	描述一个地址的信息，包括【Server, Client, Processor】		
Server	服务器地址描述，可以使用模糊匹配方式表示多个服务器。	Exact	TRUE:地址是明确的服务器地址 FALSE: 模糊指定
		Type	值有两种： RouteName: Value 为路由名称 Alias: Value 为服务器别名
		Value	根据 Exact、Type 的值设置对应的值，可以是服务器路由名称、别名、模糊表达式。
Client	连接器地址描述，可以使用模糊匹配方式表示多个连接器地址。	Exact	TRUE:地址是明确的 FALSE: 模糊指定
		Type	值有两种： ClientID: Value 为连接器 ID Alias: Value 为别名
		Value	根据 Exact、Type 的值设置对应的值。
Processor	连接器对应的消息处理器，可以使用模糊匹配方式表示多个。策略文件中该元素不需要填写	Exact	TRUE:地址是明确的 FALSE: 模糊指定
		Type	值有两种： ID: 连接器 ID ClassName: 消息处理器实现类全路径
		Value	根据 Exact、Type 的值设置对应的值。
Exclude	和元素 Include 相对应，表示在 Include 中的地址集合中需要排除的地址集合。格式规则和上面的【DestinationSet】一样		
ForwardAddresses	根元素 ForwardPolicy 下		

	的子元素，表示当 MatchConditions 中的条件表达式成立时，需要转发的地址集合描述。格式规则和上面的【DestinationSet】一样		
--	--	--	--

2.7.5.3 转发策略配置示例



如上图：

1、业务类型匹配

如果接收到的消息中，业务类型设置为“a”，那么会将该消息转发到”Gjj”下的”tlpm”

```
.....  
<fw:MatchConditions Expression="n1 ">  
<fw:Condition Name="n1" Type="MsgBZType">
```

```
<fw:MsgBZType Value="a"/>

</fw:Condition>

</fw:MatchConditions>

<fw:ForwardAddresses>

<tns:Include>

<tns:Address>

<tns:Server Exact="true" Type="RouteName" Value="Gjj"/>

<tns:Client Exact="true" Type="ClientID" Value="tlpm"/>

</tns:Address>

</tns:Include>

</fw:ForwardAddresses>

</fw:ForwardPolicy>

.....
```

2、消息内容匹配

如果接收到的消息中，指定位置 XPath ”/root/t01” 的值为”a”，那么会将该消息转发到”Gjj”下的” tlpm”

```
.....

<fw:MatchConditions Expression="n1 ">

<fw:Condition Name="n2" Type="Content">

<fw:Content ReadFrom="/root/t01" Value="a"/>

</fw:Condition>

</fw:MatchConditions>

<fw:ForwardAddresses>

<tns:Include>

<tns:Address>

<tns:Server Exact="true" Type="RouteName" Value="Gjj"/>

<tns:Client Exact="true" Type="ClientID" Value="tlpm"/>

</tns:Address>
```

```
</tns:Include>  
  
</fw:ForwardAddresses>  
  
</fw:ForwardPolicy>  
  
.....
```

3、消息源匹配

1)、地址明确匹配

当消息的源地址是” Leshan.Sichuan.gjj” 并且 源连接器 ID 是” tlpm”时，转发到” Gjj”下的” tlpm”

```
<fw:MatchConditions Expression="n1 ">  
  
    <fw:Condition Name="n3" Type="Source">  
  
        <fw:DestinationSet>  
  
            <tns:Include>  
  
                <tns:Address>  
  
                    <tns:Server Exact="true" Type="RouteName"  
Value="Leshan.Sichuan.gjj"/>  
  
                    <tns:Client Exact="true" Type="ClientID"  
Value="tlpm"/>  
  
                </tns:Address>  
  
            </tns:Include>  
  
        </fw:DestinationSet>  
  
    </fw:MatchConditions>  
  
    <fw:ForwardAddresses>  
  
        <tns:Include>  
  
            <tns:Address>  
  
                <tns:Server Exact="true" Type="RouteName"  
Value="Gjj"/>  
  
                <tns:Client Exact="true" Type="ClientID"
```

```
Value="tlpm"/>

    </tns:Address>

    </tns:Include>

</fw:ForwardAddresses>

</fw:ForwardPolicy>
```

2)、地址模糊匹配

如果消息来源是 GJJ 节点下面两层，但排除” Chengdu.Sichuan.gjj”，并且是从连接器别名是以”t”开头的客户端(排除” tlam”)来的，消息将转发给” Gjj” 下的”tlam”。

注意： 如果 Client 使用模糊表达式， Server 中指定的地址集合必需包含策略文件所在的服务器。

```
.......
```

```
<fw:MatchConditions Expression="n1 ">

    <fw:Condition Name="n3" Type="Source">

        <fw:DestinationSet>

            <tns:Include>

                <tns:Address>

                    <tns:Server Exact="false" Type="RouteName"
Value="2[gjj]"/>

                    <tns:Client Exact="false" Type=" Alias "
Value="t[a-z]+"/>

                </tns:Address>

            </tns:Include>

            <tns:Exclude>

                <tns:Address>

                    <tns:Server Exact="true" Type="RouteName"
Value="Chengdu.Sichuan.gjj"/>

                    <tns:Client Exact="true" Type=" Alias "
```

```
Value="tlam"/>

    </tns:Address>

    </tns:Exclude>

    </fw:DestinationSet>

</fw:MatchConditions>

<fw:ForwardAddresses>

    <tns:Include>

        <tns:Address>

            <tns:Server Exact="true" Type="RouteName" Value="Gjj"/>

            <tns:Client Exact="true" Type="ClientID" Value="tlpm"/>

        </tns:Address>

    </tns:Include>

</fw:ForwardAddresses>

</fw:ForwardPolicy>

.....
```

4、消息目的地匹配

请参考 3 中消息源匹配

5、组合条件匹配

策略文件中支持同时几个匹配条件组合使用

如下，当策略文件中指定的 4 个条件都满足时，消息才转发

```
<?xml version="1.0" encoding="UTF-8"?>

<fw:ForwardPolicy

    xmlns:fw="forwardpolicy.common.de.ro.icss.com"

    xmlns:tns="destination.message.common.de.ro.icss.com"

    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    xsi:schemaLocation="forwardpolicy.common.de.ro.icss.com

    Forward_Policy.xsd destination.message.common.de.ro.icss.com
```

```
DestinationSet.xsd ">

<fw:MatchConditions

Expression="n1&amp;#38;n2&amp;#38;n3&amp;#38;n4">

<fw:Condition Name="n1" Type="MsgBZType">

<fw:MsgBZType Value="a"/>

</fw:Condition>

<fw:Condition Name="n2" Type="Content">

<fw:Content ReadFrom="/root/t01" Value="a"/>

</fw:Condition>

<fw:Condition Name="n3" Type="Source">

<fw:DestinationSet>

<tns:Include>

<tns:Address>

<tns:Server Exact="true" Type="RouteName"

Value="Chengdu.Sichuan.gjj"/>

<tns:Client Exact="true" Type="ClientID"

Value="tlam"/>

</tns:Address>

</tns:Include>

</fw:DestinationSet>

</fw:Condition>

<fw:Condition Name="n4" Type="DestinationSet">

<fw:DestinationSet>

<tns:Include>

<tns:Address>

<tns:Server Exact="true" Type="RouteName"

Value="Sichuan.gjj"/>

<tns:Client Exact="true" Type="ClientID"
```

```
Value="tlam"/>

    </tns:Address>

    </tns:Include>

</fw:DestinationSet>

</fw:Condition>

</fw:MatchConditions>

<fw:ForwardAddresses>

    <tns:Include>

        <tns:Address>

            <tns:Server Exact="true" Type="RouteName" Value="

Sichuan.gjj"/>

            <tns:Client Exact="false" Type="ClientID" Value="

t[a-z]"/>

        </tns:Address>

    </tns:Include>

    <tns:Exclude>

        <tns:Address>

            <tns:Server Exact="true" Type="RouteName"

Value="Sichuan.gj"/>

            <tns:Client Exact="true" Type="ClientID" Value="tlam"/>

        </tns:Address>

    </tns:Exclude>

</fw:ForwardAddresses>

</fw:ForwardPolicy>
```

2.7.5.4 转发策略配置注意事项

1. 需要有响应的请求不支持多目的地转发(多个目的地在同一个 server 节点下)
2. 消息切片和消息转发策略避免同时使用。
3. 客户端如果使用模糊匹配，那么期望的客户端 ID 必需要在转发策略所在 server 节点上能够找到 (客户端 ID 相同)。

如果转发策略中客户端使用别名，那么，期望客户端必需和转发策略在同一个 server 节点下。

2.8 RCloud 云 EI 服务 设计工具安装

RCloud 云 EI 服务 设计工具集成在 RCloud Studio 中，使用 RCloud 云 EI 服务设计工具须首先安装 RCloud Studio，因此本章节内容请参考《RCloud4.0Studio 产品安装使用手册》。

第三章 集成交换设计与管理指南

3.1 总述

在集成交换场景中使用 RCloud 云 EI 服务，一般有如下主要步骤：

序号	主要步骤	步骤内容描述
	集成交换场景与业务分析	确定集成交换场景的涉及的应用及应用的分布，进而确定 RCloud 云 EI 服务中总线的使用模式与 RCloud 云 EI 服务服务器间的拓扑结构、传输方式、应用与 RCloud 云 EI 服务服务器间的连接方式等
		确定涉及到的业务消息格式及业务消息格式间的关系
	集成交换对象设计与部署	业务消息格式定义
		消息格式间的映射定义
		消息流(消息处理)定义
		将设计部署到相应 RCloud 云 EI 服务服务器
	总线创建与配置	使用 RCloud 云 EI 服务的管理控制台创建配置 RCloud 云 EI 服务的总线拓扑结构与 RCloud 云 EI 服务服务器间传输配置
		注册相关集成对象(需要集成的应用)
		为集成对象创建相应连接器完成到服务器的连接，若有必要并注册相应消息处理器
		通过[绑定]操作，完成输入消息格式、输入连接器到设计时定制的消息流输入节点的绑定关系；完成消息流输出节点到相应连接器及消息处理器的绑定关系；
	测试并交付运行时	在管控中启动各个连接器
		应用发送测试消息并使用消息跟踪检查消息路由与处理情况
	监控、维护与管理	消息跟踪功能可监控跟踪消息

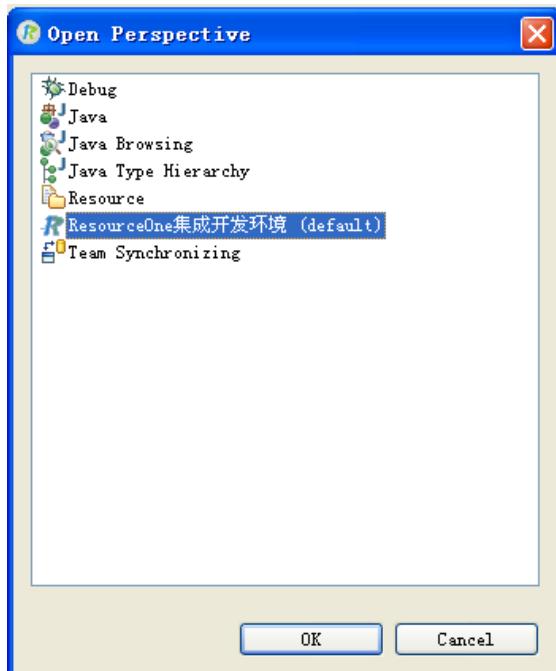
		日志查看
		异常消息管理
		信息统计

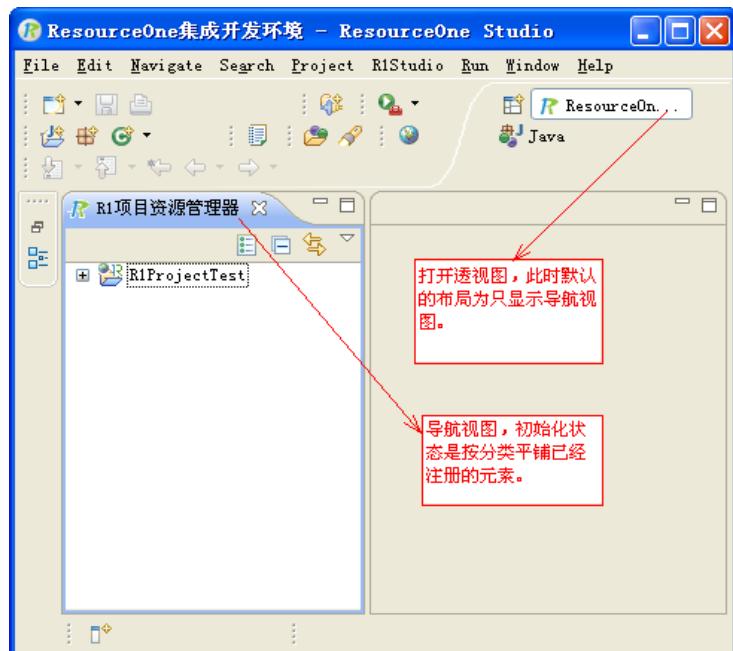
[注]: 表中步骤前提是已经在集成场景与业务分析的基础定义了应用的接入方式并完成了应用相关开发。因此应用相关开发未在主要步骤中列出。

下面说明使用 RCloud 云 EI 服务设计工具与管理控制台的第一步操作:

1、在 RCloud Studio 中使用 RCloud 云 EI 服务 设计工具设计交换服务流程

选择 RCloud 集成开发环境透视图: 点击菜单栏 Window --> Open Perspective --> Other 则出现 Open Perspective 页面, 选择“RCloud 集成开发环境”, 如下图所示:





选择透视图

2、在 RCloud 云 EI 服务管理控制台中构建服务总线

在浏览器通过地址: <http://host:port/context> 访问管理控制台, 将会出现下面的登录页面: 默认的用户: admin 密码: pass



RCloud 管理控制台登录界面

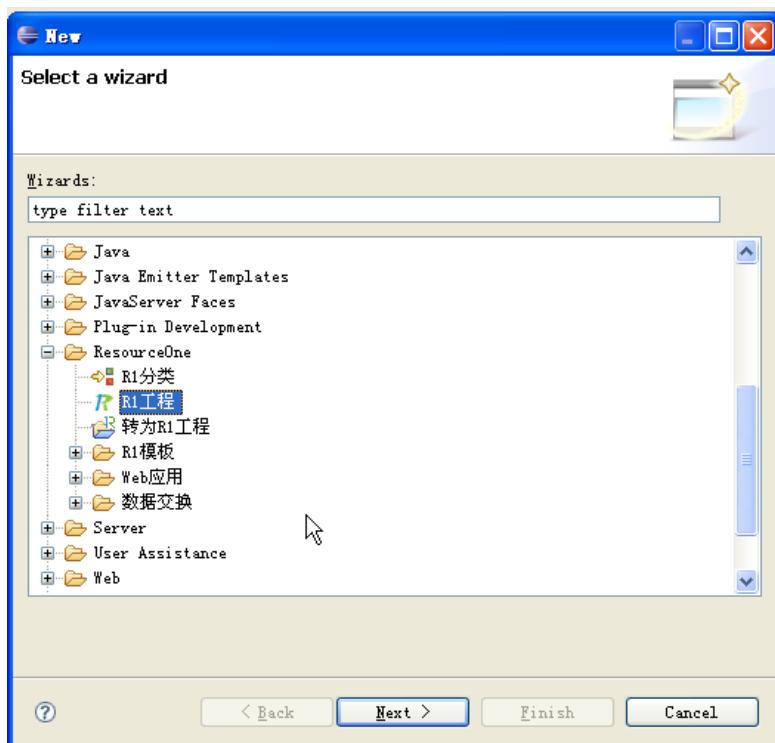
3.2 RCloud 云 EI 服务设计工具使用指南

在创建服务总线等具体的构建工作之前，首先应根据业务环境中各应用系统之间具体的转换需求与交换流程，使用 RCloud Studio 中的 RCloud 云 EI 服务设计工具以图形化的方式设计并展现应用系统之间的这些信息交换逻辑流程。这些设计包括交换服务流程设计、消息格式设计以及不同消息格式之间的映射关系设计，设计完成后保存到 RCloud 云 EI 服务服务器中，供运行时在 RCloud 云 EI 服务服务器中实例化运行。

3.2.1 运行 RCloud Studio

RCloud 云 EI 服务设计工具已经集成在 RCloudStudio 中，直接进入 RCloudStudio 工作区。然后在“RCloud 集成开发环境”透视图下进行相关设计。

首先，需要新建一个 RCloud 工程。打开菜单的 FILE>>NEW>>Other，在弹出的对话框中选择 RCloud 工程。



创建 RCloud 工程

RCloud 工程的目录结构如下：



资源管理器

3.2.2 连接配置

连接配置包括: RCloudStudio 中的数据库连接配置、RCloud Studio 到 RCloud 云 EI 服务服务器配置、RCloudStudio 到 RCloud 统一资源库配置。

其中数据库连接配置主要用于 RCloud Studio 工具中对于数据库连接的引用, 如创建 DAO/VO 时使用的数据库连接、创建到 RCloud 云 EI 服务服务器的 jdbc 时引用到的数据库连接等等。

RCloud Studio 到 RCloud 云 EI 服务服务器的连接, 主要用于将设计结果通过该连接保存到 RCloud 云 EI 服务服务器; 连接有两种类型, 一是 JDBC 连接, 一是通过 Web 服务连接。

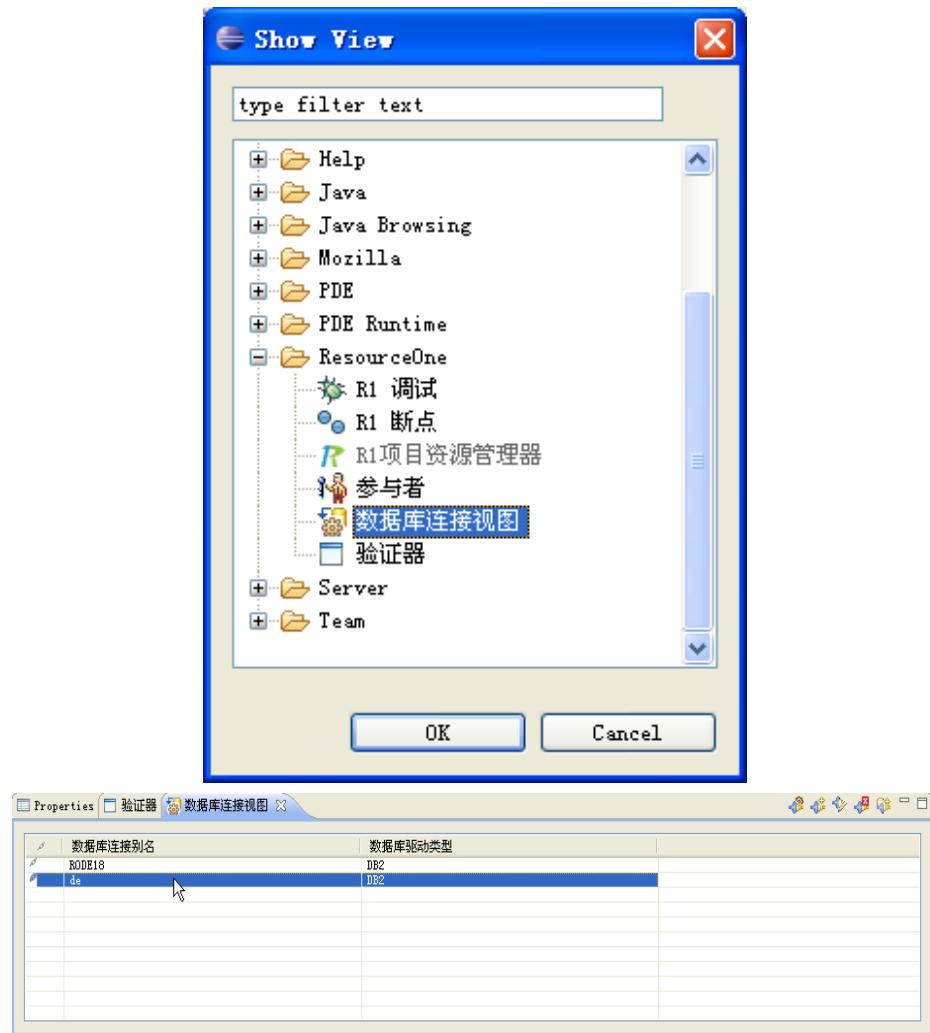
RCloud Studio 到 RCloud 统一资源库的连接, 主要用于将设计结果通过该 Web 服务连接发布到 RCloud 统一资源库。

以上三种连接具体配置操作参见下文描述。

3.2.2.1 数据库连接配置

配置数据库连接, 是通过 RCloud Studio 的 Connection(连接)视图创建数据库连接。在 RCloud 云 EI 服务设计工具中需要操作数据库使用的所有的连接都要在该视图中创建。

通过 RCloudStudio 菜单: Window → Show View → Other, 在弹出的对话框中选择“数据库连接视图”出现数据库连接编辑界面。



数据库连接视图

数据库连接编辑界面点击新建，在弹出的对话框填入相应信息。点击“测试”通过，这样一个数据库连接就完成了。



创建数据库连接

在数据库连接视图，建立所需要的数据库连接。

建立 MS SqlServer 类型的数据库连接时，连接字符串需要写为以下形式：

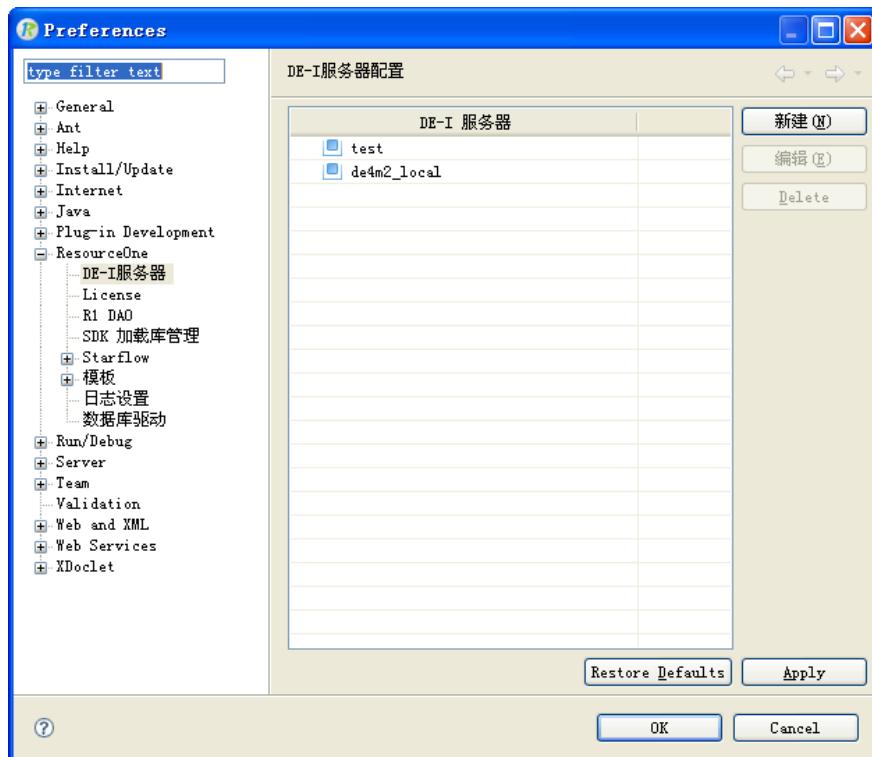
```
jdbc:microsoft:sqlserver://192.9.100.33:1433;DatabaseName=rode;SelectMeth
```

```
od=cursor
```

ip 地址和数据库名字可以根据实际情况修改，其它的需要带着。

3.2.2.2 RCloud 云 EI 服务服务器连接配置

通过 RCloudStudio 菜单：Window → Preferences → RCloud → EI 服务器，进入如下页面。



RCloud 云 EI 服务连接配置

3.2.2.2.1 JDBC 方式连接

点击“新建”按钮，在弹出的“新建服务器”页中，执行如下操作：

- 1) 填写服务器连接显示名。
- 2) 连接方式选择 JDBC 连接。
- 3) 点击“从[数据……JDBC 连接”按钮，在弹出的连接列表中选择一个数据库连接。如果连接列表中没有需要的连接，需要新建一个数据库连接，然

后再选择。(参考数据库连接的配置)

- 4) 点击“OK”完成。

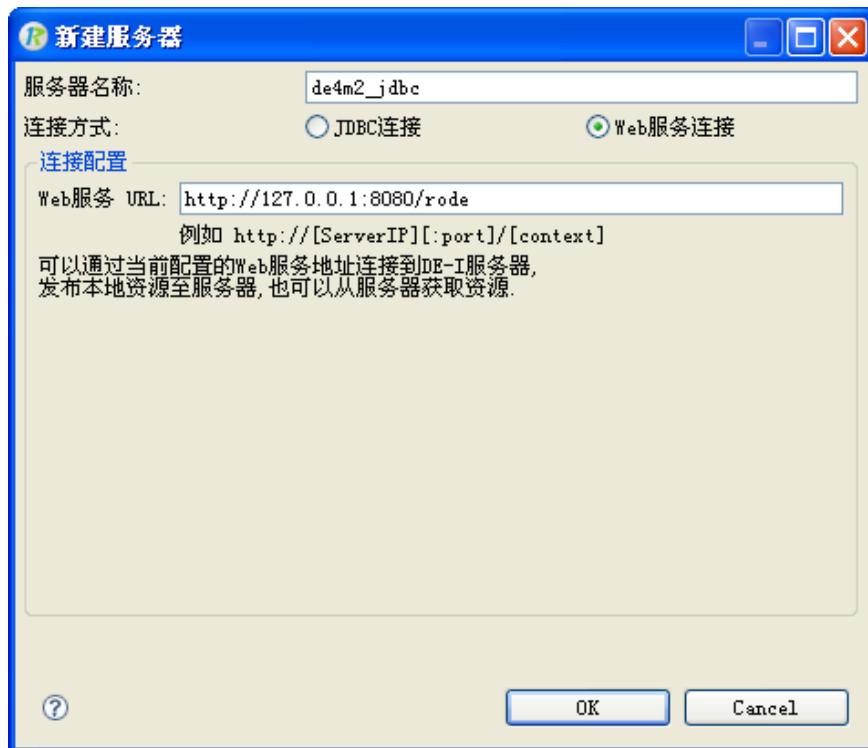


新建 RCloud 云 EI 服务服务器 JDBC 连接

3.2.2.2 WEB 服务连接方式

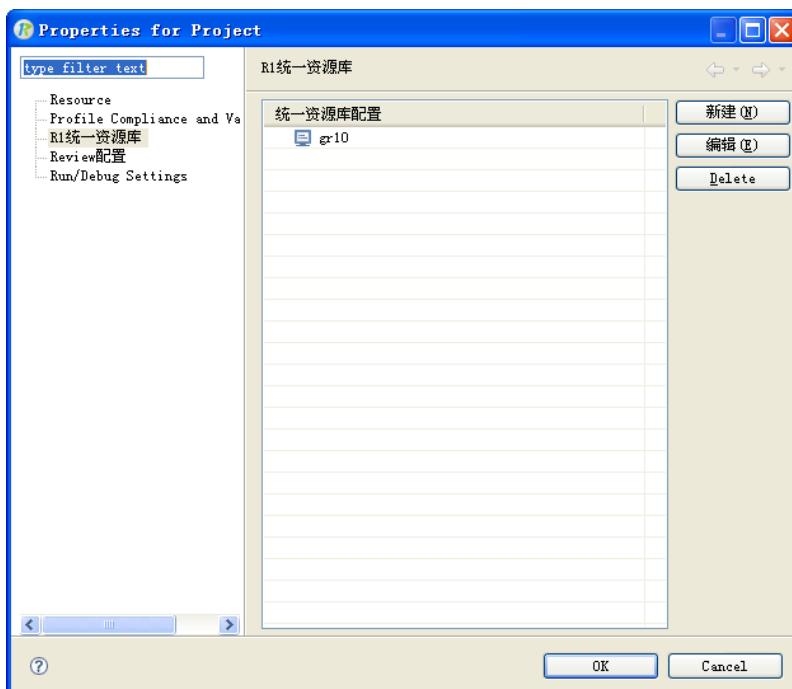
点击“新建”按钮，在弹出的“新建服务器”页中，执行如下操作：

- 1) 填写服务器连接显示名。
- 2) 连接方式选择“WEB 服务连接”。
- 3) 填写 RCloud 云 EI 服务服务器地址，格式为 http://[ip]:[port]/[context]
- 4) 点击“OK”完成。



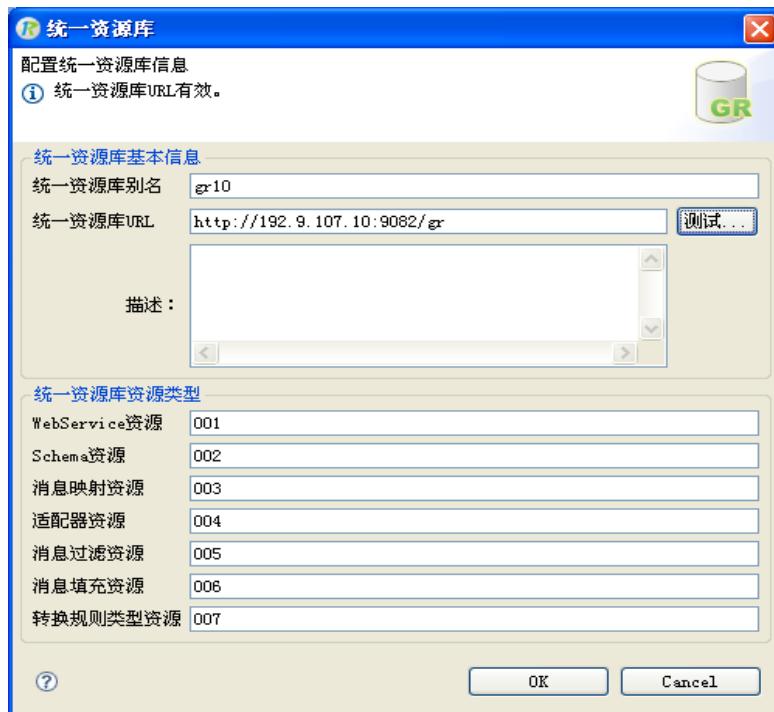
3.2.2.3 统一资源库连接配置

选择当前工程，点击鼠标右键菜单的“Properties”，在弹出的窗口中选择左侧树的“RCloud 统一资源库”。如下图：



点击“新建”，在弹出的窗体中填写资源库显示名称和访问地址，测试通过

后，点击“OK”，完成配置。

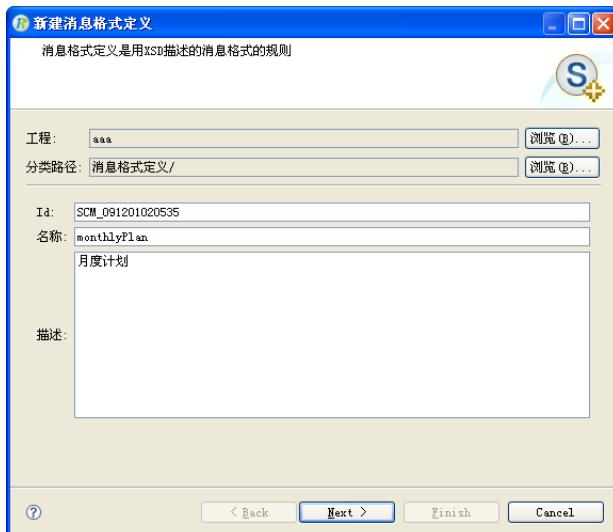


3.2.3 消息格式定义

消息格式定义 即 XML Schema，是对 XML 消息格式的约束描述。

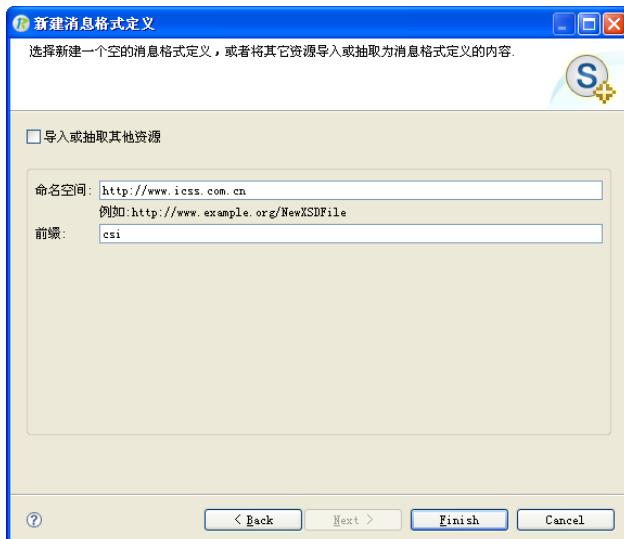
3.2.3.1 创建消息格式定义

在 RCloud 项目管理器的 RCloud 工程的交换对象组件中，右击消息格式定义-->New-->消息格式定义，在弹出对话框中输入 ID、名称以及描述后点击'Next'。



新建消息格式定义首页

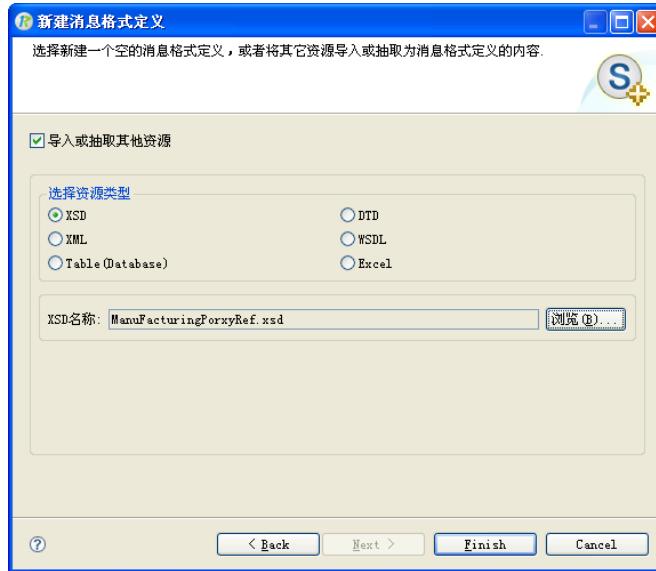
在新的页面里输入命名空间与前缀，点击'Finish'将产生一个默认的包含有一个元素和复杂类型的消息格式。



添加消息格式定义属性

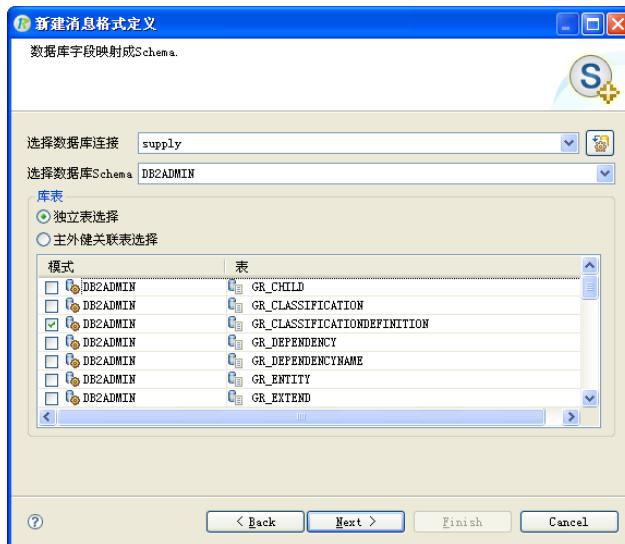
在第二页若选择'导入或抽取其他资源'时，选择资源类型，将已经存在的资源内容抽取或导入到将要产生的消息格式里，根据选择不同的资源类型，存在如下 6 种操作：

- 1) 选择 XSD 时，点击'浏览'选择一个已存在的 XSD 文件。



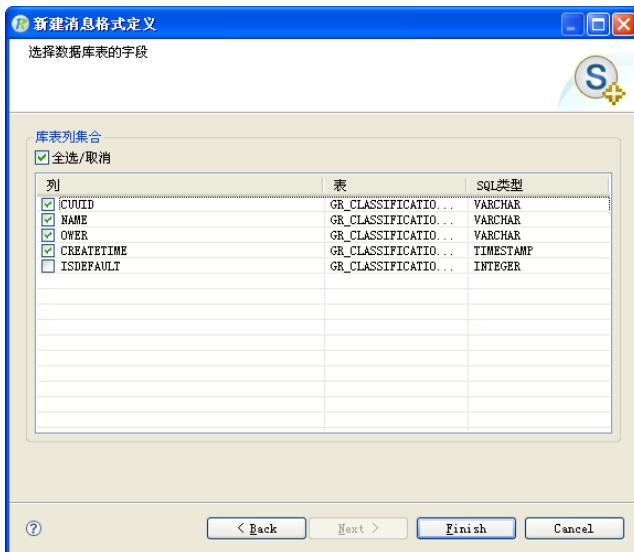
导入 XSD

- 2) 选择'DTD'时，输入命名空间及前缀，然后点击'浏览'选择一个已存在的 DTD 文件。
- 3) 选择'XML'时，输入命名空间及前缀，然后点击'浏览'选择一个已存在的 XML 文件。
- 4) 选择'WSDL'时，点击'浏览'选择一个已存在的 WSDL 文件,然后选择一个 Schema。(注：目前版本不支持对含有多个 Schema 的 WSDL 的抽取)
- 5) 选择'Table (Database)'时，输入命名空间及前缀，点击'Next'，通过下图所示页面进行数据库及表的选择，再点击'Next'，



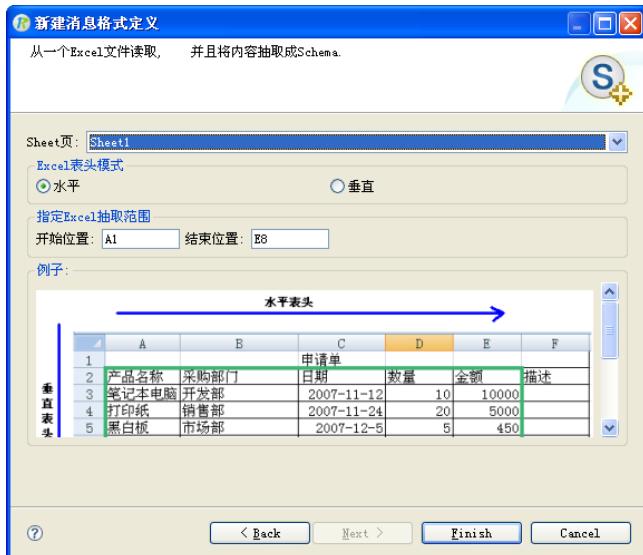
选择数据库与表

在新的页面里勾选需要的字段，如下图所示：



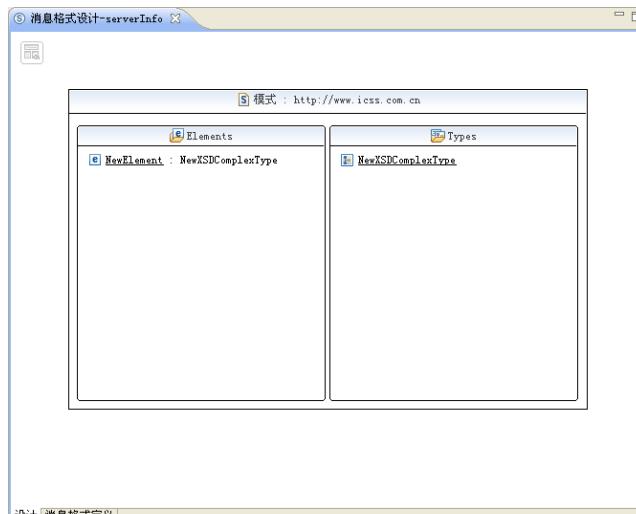
选择表字段

- 6) 选择'Excel'时，输入命名空间及前缀，然后点击'浏览'选择一个已存在的 Excel 文件，点击'Next'，得到下图所示的内容



抽取 Excel

创建后的消息格式以 XSD 作为存储方式，其编辑界面如下图所示：(此版本之前产生的消息格式以 DTD 为存储方式)。

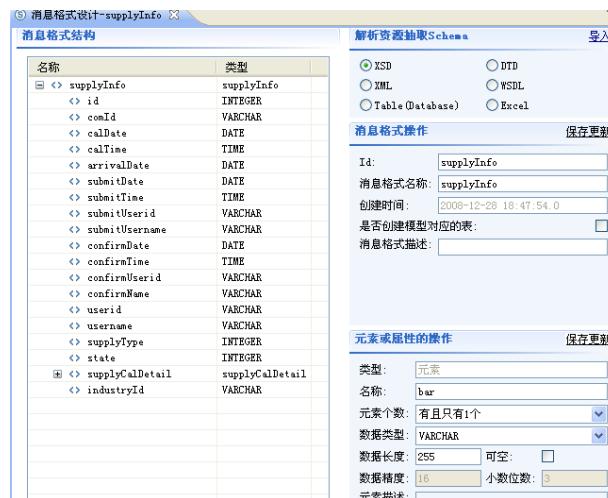


消息格式(XSD)编辑器

3.2.3.2 设计消息格式定义

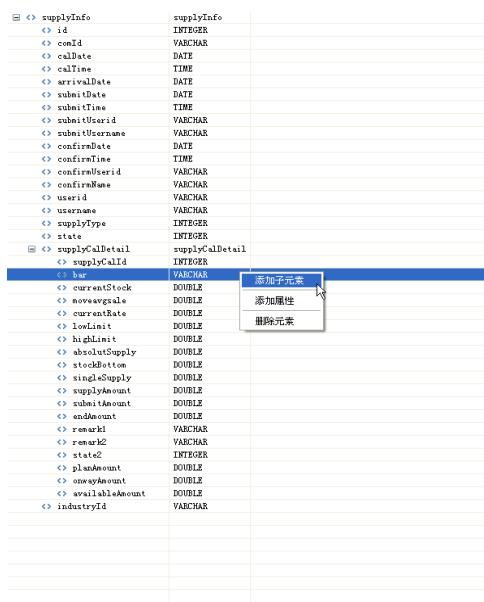
Studio 支持以 XSD 作为存储方式的消息格式(下简称消息格式 XSD))的设计，同时也支持之前版本所产生的以 DTD 为存储方式(下简称消息格式(DTD))的消息格式的编辑。

➤ 消息格式(DTD)编辑界面如下：



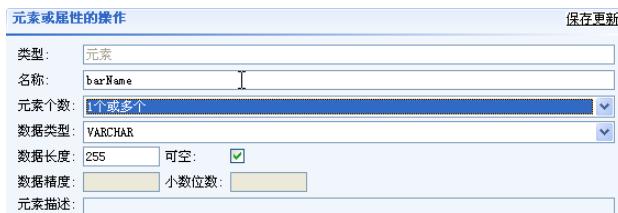
消息格式(DTD)编辑器

在‘消息格式结构’区域内的结构树的某一元素上右击→添加子元素，如下页面所示：



消息格式(DTD)编辑器

元素可通过如下如下的编辑器的右下角区域进行编辑，在此可以编辑其名称、元素个数、数据类型等属性，在修改属性后，需要点击右上角的‘保存更新’方可将更新进行保存。



元素编辑区域

跟添加子元素类似，通过在某一元素上右击→添加属性来添加元素的属性，同样在编辑器右下角可以编辑属性的名称和类型，其操作与元素编辑类似。

在结构树上右击元素或属性再点击→删除元素或者删除属性即可删除元素或属性。

编辑器右上角的“解析资源抽取 Schema”用来导入或抽取已经存在资源的内容，抽取其中部分或全部内容作为当前消息格式的 DTD 文件的内容，其操作方式与新建消息格式时的操作类似。

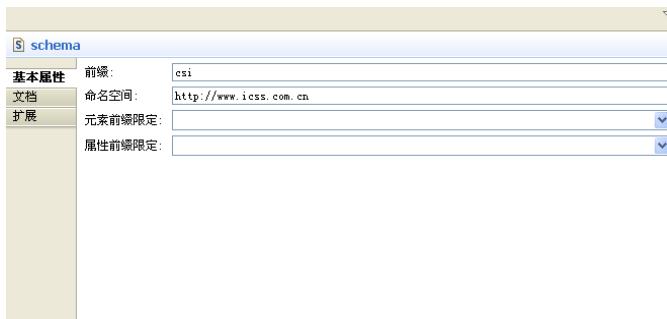
➤ 消息格式定义(XSD)编辑器如下图：



消息格式定义(XSD)编辑器

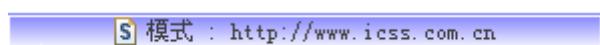
消息格式(XSD)设计包括 XSD 设计以及消息格式定义设计两部分：

1. '设计'页面用于编辑消息格式的 XSD，并利用如下所示属性视图对命名空间、前缀、元素及类型进行更名等具体的、细节的编辑操作。



XSD 的属性视图

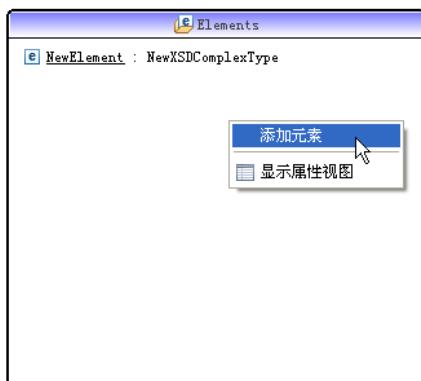
消息格式定义(XSD)编辑器的上部区域



命名空间显示区

为当前 XSD 的属性显示区，此处显示 XSD 的命名空间，鼠标单击此区域，在属性视图上进行编辑此 XSD 的命名空间、前缀等属性。

编辑器左下区域为元素编辑区，如下图所示，在其空白处右击鼠标再选中‘添加元素’可以添加一个新的元素，元素的基本属性通过属性视图进行重命名、类型设置等编辑操作。



元素编辑区

元素的类型也可通过编辑器设置，其操作步骤为：

1. 右击将要设置类型的元素，此处为'NewElement1'，点击‘设置类型’。



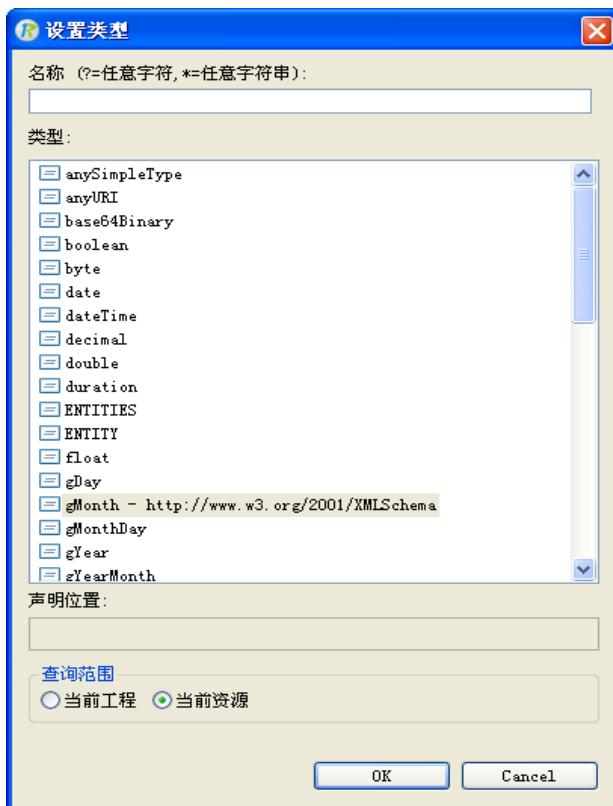
设置类型

2. 可以新建一个类型也可以使用已存在的类型作为当前元素的类型，点击‘新建...’后，在弹出对话框中根据提示可以选择复杂类型、简单类型以及匿名的本地类型，选择任意一种情况后点击‘OK’即产生此元素的类型。



新建类型

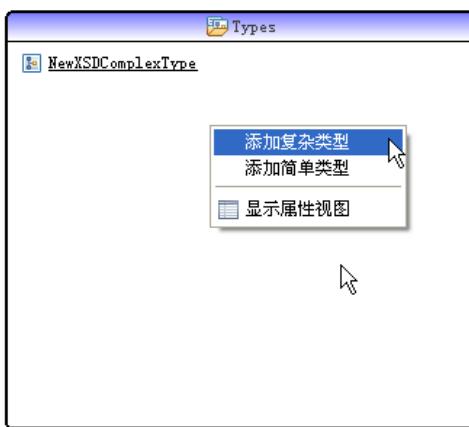
3. 若选择浏览已存在元素时，点击‘浏览...’，将弹出如下对话框，在对话框中找出需要的类型，对话框的视图默认显示为当前资源里的所有类型，可以通过改变查询范围将范围改为当前工程。



元素类型选择器

注：XSD 的最外层的元素不能进行多重性的设置，因此在此处设置元素多重性将不能实现。

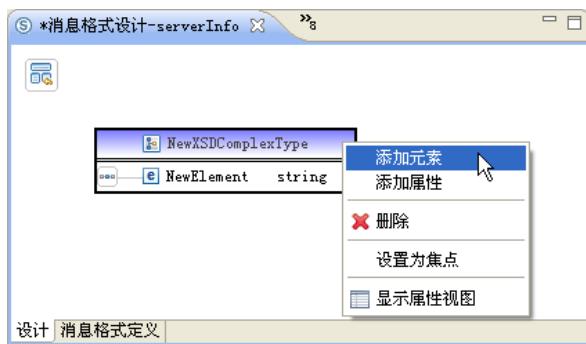
编辑器的右下角为 XSD 类型设计区，如下图所示，在空白区域内右击鼠标添加复杂类型或简单类型：



类型编辑区

添加一个简单类型后，单击此类型，在其属性视图上对其进行编辑。

如果添加了一个复杂类型，通过双击或者右击→设置为焦点进入复杂类型编辑页面：



复杂类型编辑区

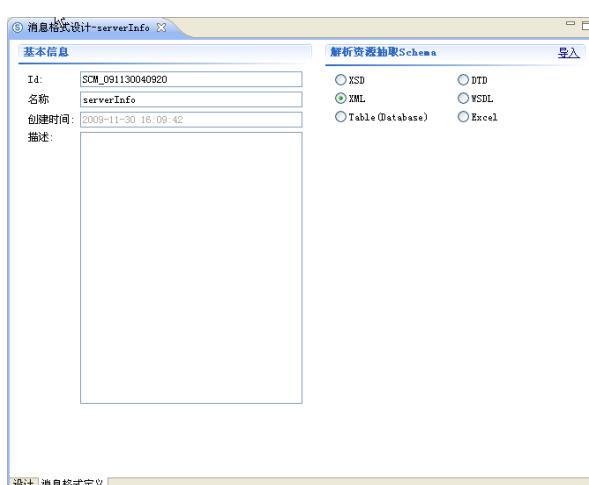
在上图所示页面里通过右击鼠标可以进行添加元素和属性的操作，添加后的元素和属性的基本属性在其对应属性视图里进行编辑，并且此处的元素可以通过右击鼠标→设置多重性进行多重性的设置。



元素多重性设置

设计好复杂类型之后，通过点击编辑器左上角的 返回到编辑器。

2. '消息格式定义'页用于编辑消息格式定义描述信息以及导入抽取外部资源：



消息格式定义

上图所示页面左边“**基本信息**”区域可进行消息格式名称、描述信息的修改。右边“解析资源抽取 Schema”用于导入或抽取外部资源作为消息格式的 XSD 的内容，其操作方式与新建消息格式时的方式类似。

3.2.4 格式转换器定义

3.2.4.1 消息映射定义

消息映射可以可视化定义。源和目标来源消息格式中定义，也可以导入文件。

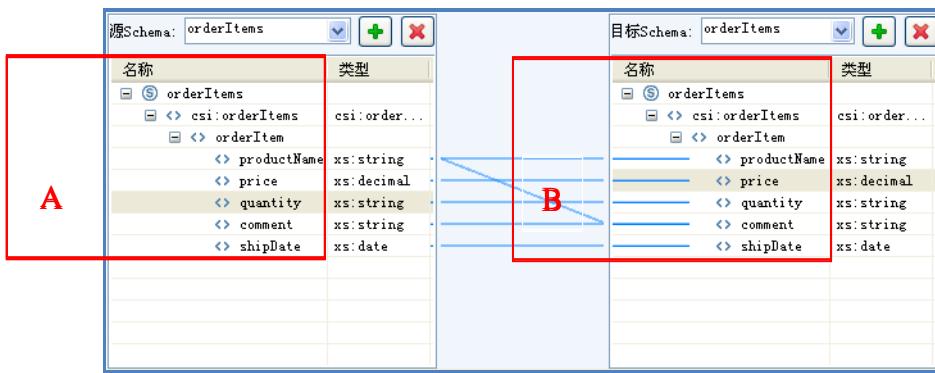


消息映射导入

一、新建消息映射

鼠标移到“消息映射”目录，点击右键，在出现的右键菜单中选择 NEW → 消息映射，填写 Schema 映射文件名，点击“Finish”创建成功一个新的空消息映射文件。

需要对消息映射编辑，鼠标双击刚创建的 Schema 映射文件，进入消息映射设计界面。



消息映射设计

1、介绍界面：

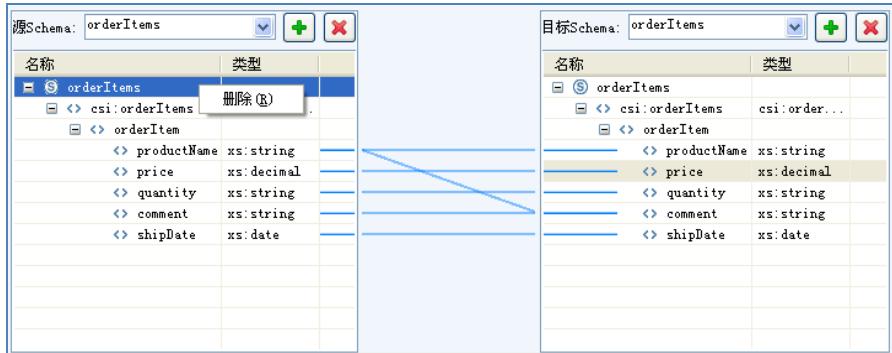
图中左、右两处分别为源 Schema 和目标 Schema 编辑区域，可以通过按钮添加删除。

添加：Schema 下拉列表中选择源 Schema，点击 添加。

删除：Schema 下拉列表中选择源 Schema，点击 删除。

注意：目标 Schema 只允许添加一个，源 Schema 可以有多个。

对 Schema 删除操作也可以选择 Schema 节点右键菜单并选择“删除”。如下图：



介绍界面

注意：

- 1) 当删除目标 Schema 则将删除之前所有的映射；
- 2) 当源 Schema 子节点中创建了映射时，会删除之前所有的映射，但如果源 Schema 中没有任何节点存在映射，则不会。
- 3) 源树和目标树显示和映射操作(拖拽)区域，可以拖拽左侧树(源)节点到右侧树(目标)节点上，并添加一连接线。用户可以在映射列表中选择一映射条目，自动关联映射连接。

2、创建从源到目标的映射

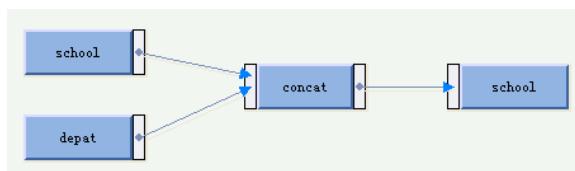
用户可以从源树节点到目标树节点拖拽一映射连接，并在映射列表中生成一映射条目。

1) 从非含有子元素的元素节点做映射的时候(包括简单元素节点和属性节点，映射表达式规则为赋值。



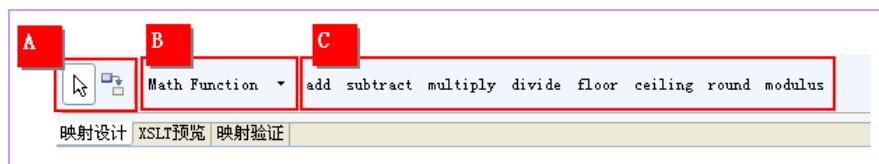
从源到目标的映射 1

2) 当多个源节点映射到同一个目标节点，映射表达式规则为 concat。



从源到目标的映射 2

用户可以自定义复杂的映射表达式规则，映射设计工作区底部(见下图)为常用表达式函数，用户通过简单的拖拽就可以生成复杂的映射表达式规则。



映射规则函数

上图区域分布介绍：

A 区域：鼠标工作状态选择， 为恢复， 为创建连接，就是在两节点间画线。

B 区域：函数类型选择区，现在提供五种类型的函数。

Math Function： 数学计算的常用函数。

Conversion Function： 类型转换的常用函数。

Logical Function： 逻辑判断函数。

String Function： 字符串处理函数。

Constants： 常量处理函数。

C 区域：根据 B 区域类型的选择，列出当前支持的函数。

3、创建主题映射

后续版本实现。

二、新建消息映射(导入):

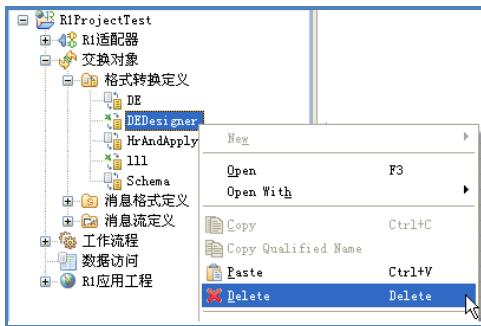


消息映射导入

三、更新消息映射:

直接在 Schema 映射设计界面修改，保存。

四、删除消息映射:



删除消息映射

五、保存消息映射:

自动保存。

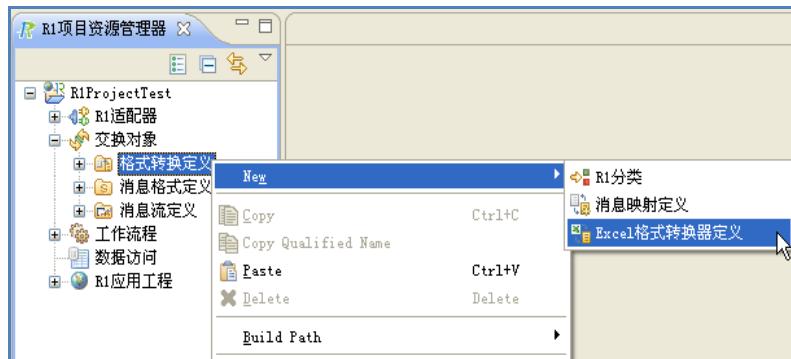
六、导出消息映射

操作和导出 Dex 资源文件基本相同。详细参照导出 Dex 资源文件。

3.2.4.2 新建文件格式转换器

文件转换规则现在提供 Excel 文件抽取规则；新建规则步骤如下：

鼠标右键点击“格式转换规则”->new->Excel 文件格式转换器定义：



Excel 格式转换器定义

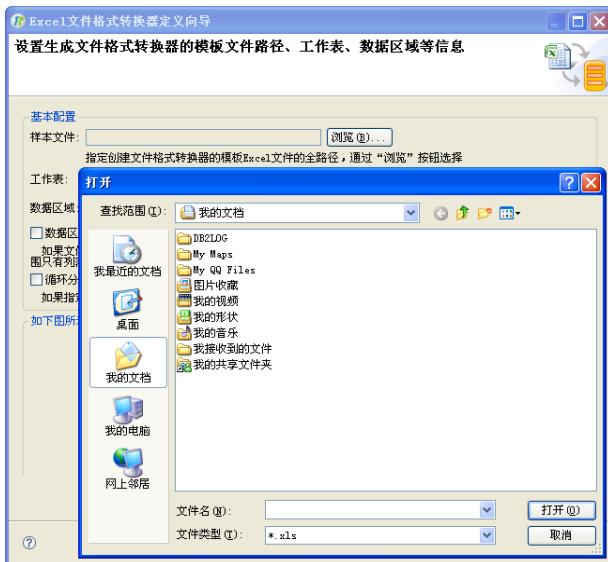
出现新建转换规则界面，输入转换规则 ID 和转换规则名称，如“testId”，“testName”，如下图：



文件转换向导

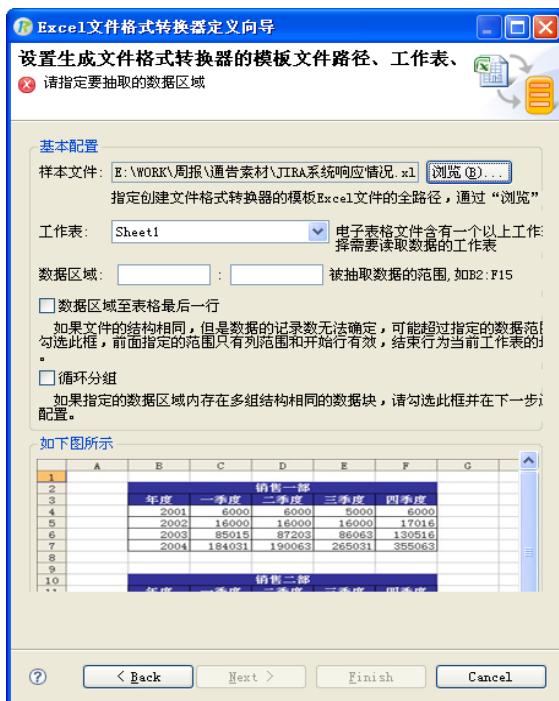
点击“next”到下一步：

点击在“样本文件”后的“浏览”按钮，选择样本文件，在引入样本文件后，选择抽取的 Excel 文件中的那个工作表单(Sheet)，填写数据区域：如 A1，C4



浏览样本文件

在这里如果数据需要循环分组抽取，则需要选中“数据有循环分组”出现如下界面：



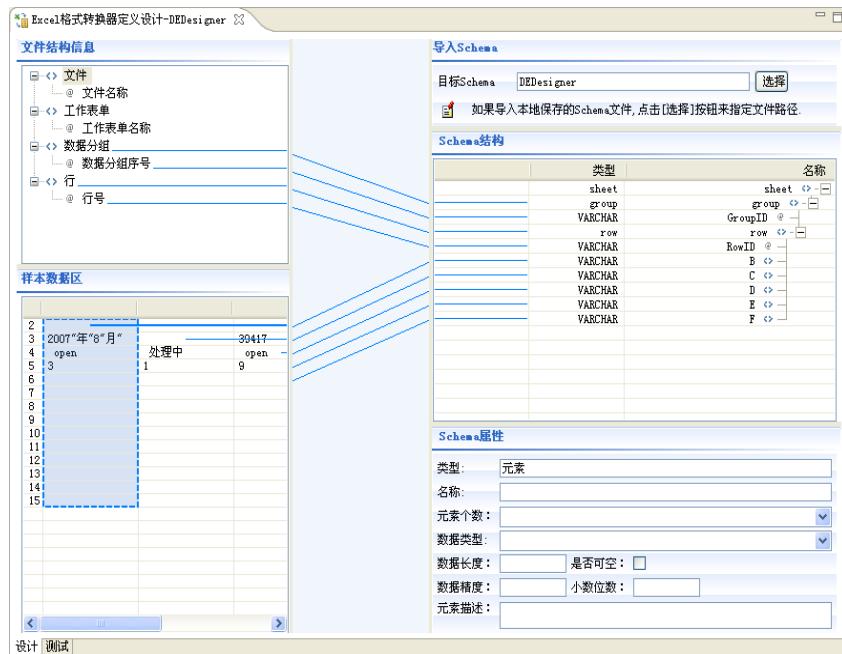
数据行循环分组配置

如果不是循环分组抽取数据则不选“数据有循环分组”选项。如下图所示：



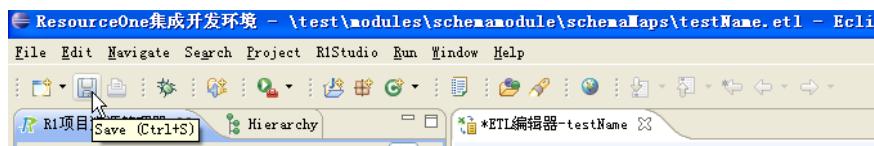
提示界面

填写好表头行数和表头所在的行数，点击“Finish”完成新建并打开转换规则编辑器，如下图所示：



转换器规则编辑器

规则填写和新建一个消息 Shema 一样。作完映射后，点击“保存”按钮，保存转换规则：



保存规则

保存完新设计的文件格式转换器，需要再把转换规则发布到统一资源库中：转换规则的发布和消息流的发布一样。

3.2.5 消息流定义

3.2.5.1 新建消息流文件

消息流文件扩展名为 dex。首先新建工程。



新建工程

RCloud 工程 → 交换对象 → 消息流定义，点击右键菜单的 NEW → 消息流定义文件，可以新建消息流定义文件。如下图所示：



新建交换文件

“新建消息流定义文件”的向导框，开发商标识是流程开发商的“身份证”。需要注意的是，该标识只能在新建消息流定义文件时输入，并且新建文件以后，这个标识就不能更改了。该标识会被加密存储到消息流定义文件中。所以一定要记住在这里输入的“开发商标识”。在设计好消息流后，最后需要把消息流定义文件导入到数据库中，在导入数据库时，会要求输入开发商标识，如果在导入数据库时输入的开发商标识和新建时输入的开发商标识不符，就会提示：“保存到数据库时身份认证失败！”，流程定义文件就无法导入到数据库中。开发商名称填写后，可以通过消息流包编辑器界面更改。

3.2.5.2 新建消息流

一、创建消息流

要创建和修改消息流，在当前项目工程下，单击选择该包名，在树节点上右击选中 New → 消息流定义，在左边编辑框中创建新的消息流，并输入消息流名称和组，点击“Finish”按钮。

注：组是用来分类流程，如可以按组划分“产品部”以及“销售部”流程类别



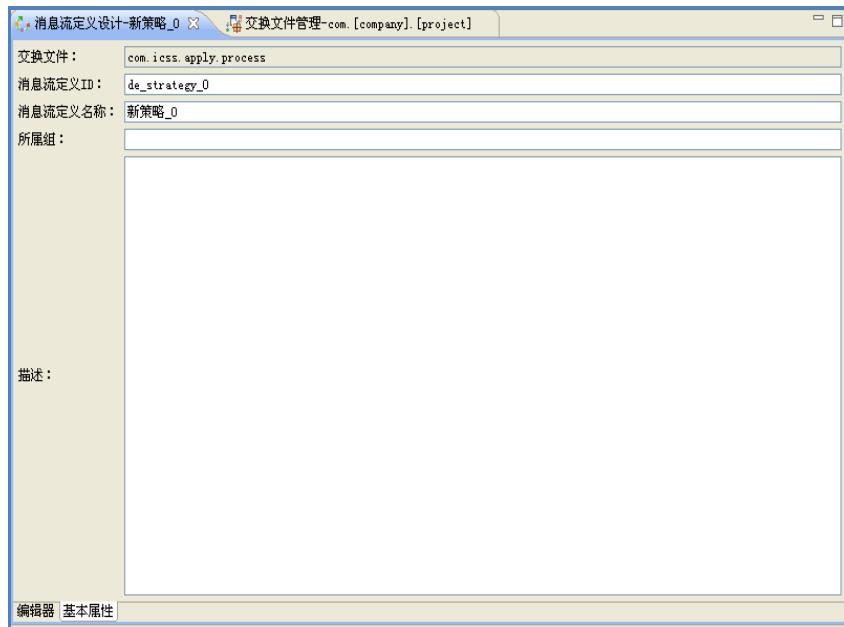
新建流程

如果创建的是代理服务流程，请将上图中“对服务进行封装”选中，可以直接点击“Finish”完成；也可以点击“next”（如下图），根据已经设计好的代理服务的 WSDL 指定服务接口，或者是自定义服务接口，然后再点击“Finish”完成。



二、属性对话框

在消息流定义设计界面，点击底部的『基本属性』，如下图所示：

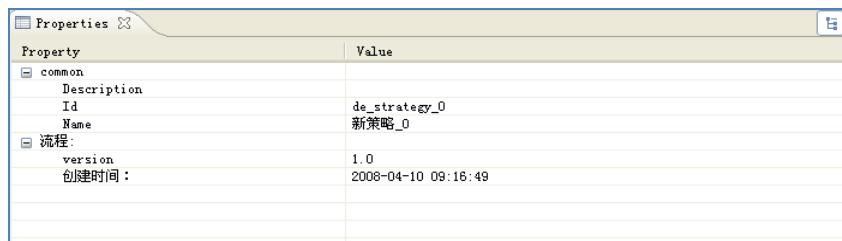


基本属性

可以打开消息流编辑框，填写流程各属性信息。其中流程 ID、流程名称是必填项，所属组和描述是可选项。也可对已有的消息流进行修改。这里需要注意的是，流程 ID 是必须是唯一的，并且建议 ID 可以代表一定的流程含义。

三、属性栏

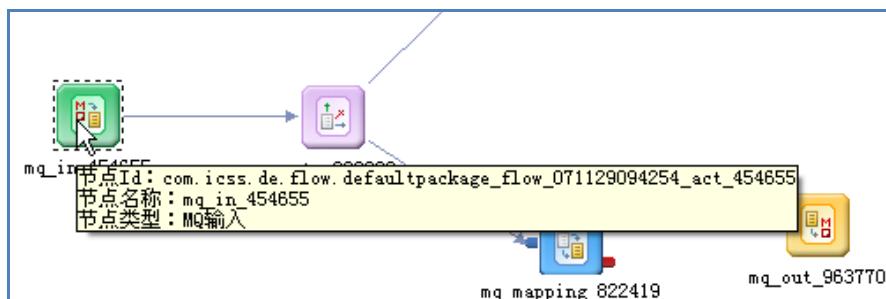
选择不同流程或包节点在属性栏中都会显示相关信息



详细信息显示

四、Tooltip 显示

鼠标 Over 流程节点或流程编辑器出现信息提示。如下图：



信息提示

3.2.5.3 消息流节点

此功能是提供消息流节点的创建和属性修改，右键消息流节点，点击『打开流程』，可以进入消息流的设计建模过程，下面将按照右侧 Palette 中各种消息流节点排列顺序依次说明。



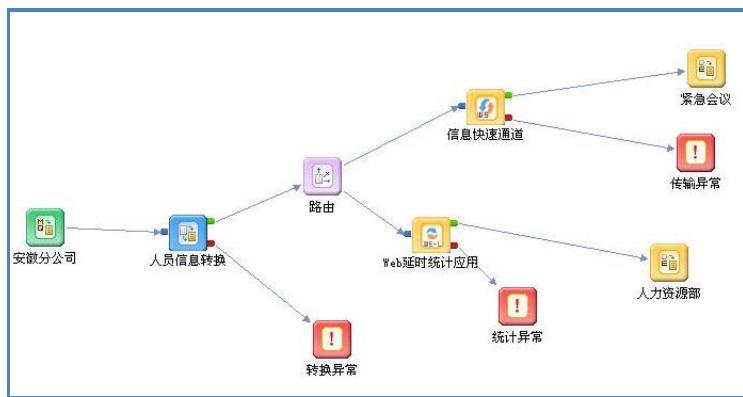
输入节点

输出节点

代理服务封装节点

普通节点

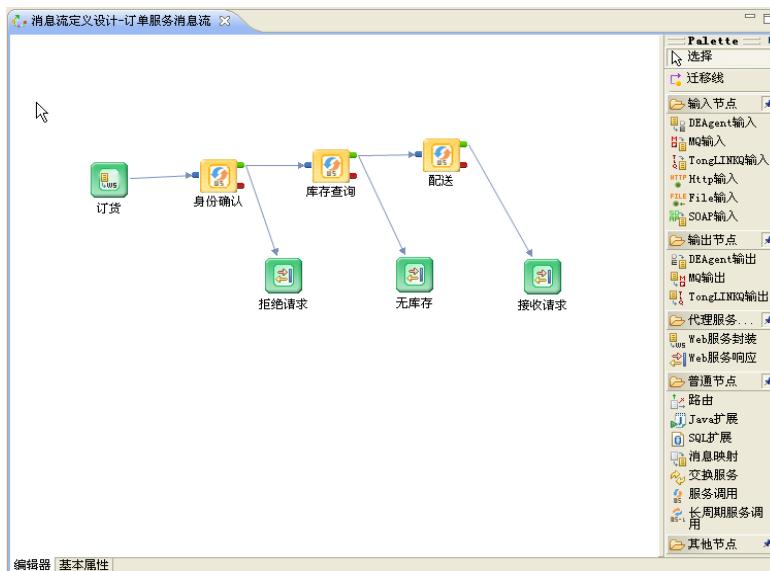
其他节点



活动节点预览

一、节点操作概览

点击 Palette 上的『DE Agent 输入』、『路由』、『Java 扩展』等按钮，可在编辑器内创建相应的消息节点。



编辑消息节点

1、属性对话框

右键节点，点击『属性』，可以打开该节点的属性页面，以路由节点为例：

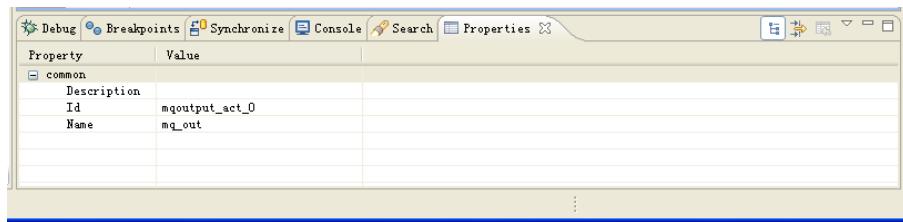
如下图所示，『ID』是指这个节点的 ID，RCloud 云 EI 服务设计工具会自动生成一个保证唯一的值，没有特殊的需求，不需要修改，『名称』是指这个节点的名称，为了保证流程图的可阅读性，建议填写较为明确的名称，例如：CRM 客户端应用输入、某个人工作台等等。『描述』属性可以填写本节点的描述性内容。



路由节点

2、属性栏显示

活动节点属性也可以通过属性栏(Window -> showView -> other -> General -> Property 视图)显示。



活动节点属性

3、Tooltip 显示:

鼠标 Over 节点时候出现 Tooltip 显示当前节点相关信息。



相关信息提示

二、输入节点

输入节点是指一个流程的消息输入节点，是整个流程启动的入口。对应 RCloud 云 EI 服务的概念是连接器，一个流程必须存在输入节点。输入节点有：DEAgent 输入、MQ 输入、SOAP/WS 输入、Http 输入、FTP 输入、File 输入

入



输入节点

1、DEAgent

当接入方式为 DEAgent 客户端这种连接器时，也就是使用 RCloud 云 EI 服务的客户端 API 来发送消息，那么这样的客户端就对应 DEAgent 输入节点。双击 DEAgent 输入节点图标，出现 DEAgent 输入节点的属性页面，如下图所示：



DEAgent 输入



DE 输入属性对话框

1) 响应方式

设定该节点的响应方式，默认无响应。

2) 多输入关联键值(输入时必须设置)

通过设置多输入的 XPath 来定义关联。如:存在两个输入应用 CRM, STORE。其消息格式(或 XML)定义中规定:/customer/customer_id(CRM)和 /store/customer_id (STORE)关联, 即分别为两个输入的 XPath 值, 如果不是多输入 XPath 可不填。

3) 消息格式

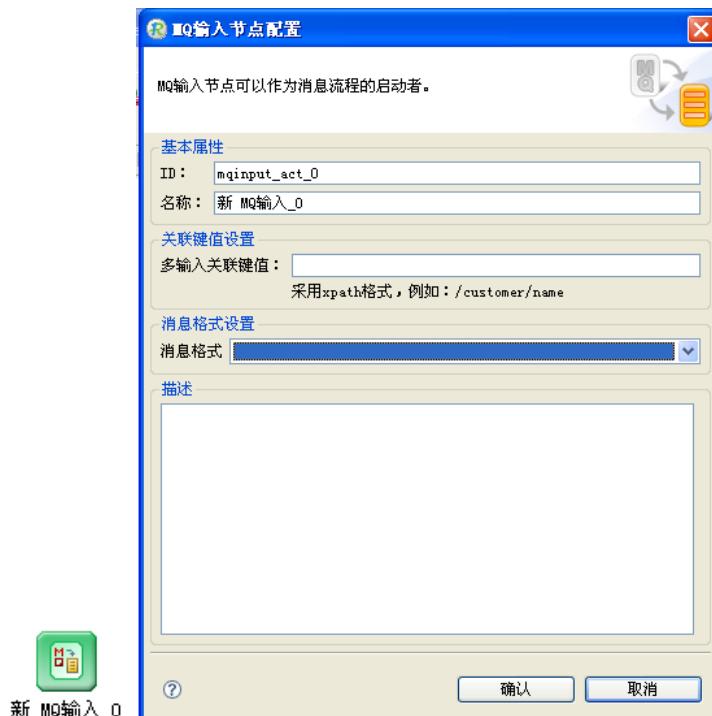
在下拉框中选择该节点的消息格式文件。

2、MQ 输入节点

接入方式为 MQ 方式时, 即对应 RCloud 云 EI 服务的 MQ 连接器。

注意: 在 RCloud 云 EI 服务设计工具中创建 MQ 节点后, 要在对应的 MQ 服务器上创建名为“RONEDE. SVRCONN”的服务器连接通道。

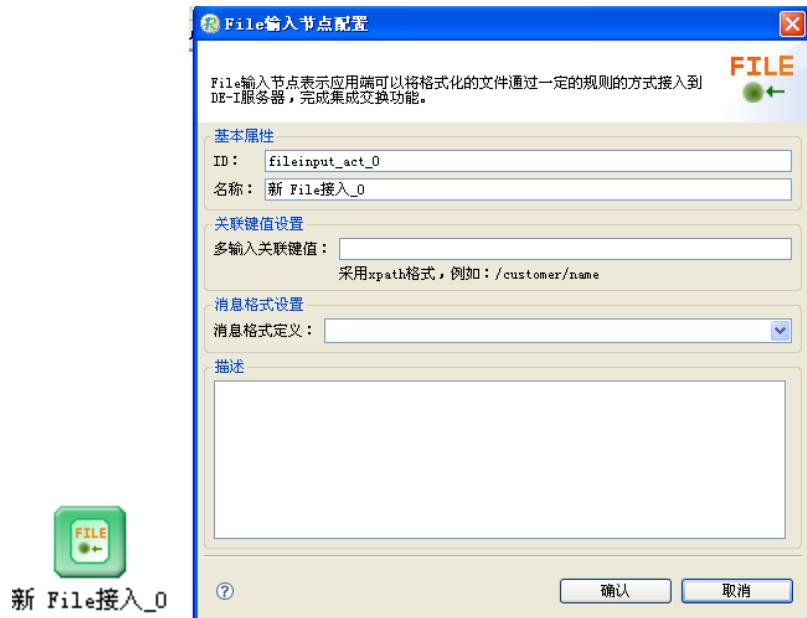
多输入关联键值、消息格式, 同 DEAgent 输入。



MQ 输入属性对话框

3、文件输入节点

当接入方式为文件输入方式时, 也即对应的 RCloud 云 EI 服务的文件连接器, 参数配置同 DEAgent 输入。



4、HTTP 输入节点

当接入方式为 HTTP 方式时，也即对应 RCloud 云 EI 服务的 HTTP 连接器。

参数配置同 DEAgent 输入。

5、SOAP 输入节点

同 DEAgent。但是当该 SOAP 输入节点支持消息有响应时，需要配置响应的相关属性。

接受响应信息的Web服务配置

设置用于接受响应信息的web服务,该服务必须由客户提供,并已发布到统一资源库.

统一资源库:	gr10	<input type="button" value="配置资源库"/>
Web服务名称:	SaleService	例如: SaleDataService
Web服务版本:	1.0	例如: 1.0
Web服务命名空间:	http://de.ro.icss.com/xsd	例如: http://de.ro.icss.com/xsd
Web服务操作名:	getSaleData	例如: getSaleData
<input type="checkbox"/> 服务操作允许处理附件		

统一资源库：选择一个已经配置的 RCloud 统一资源库，如果下拉列表为空，可以点击“配置统一资源库”按钮，配置一个新的 RCloud 统一资源库（参考 RCloud [统一资源库配置](#) 说明）。

Web 服务名称：接收响应消息的客户端 Web 服务在 RCloud 统一资源库中的注册服务名。

Web 服务版本：接收响应消息的客户端 Web 服务在 RCloud 统一资源库中的

注册的版本。

Web 服务命名空间: 访问该服务操作的命名空间, 用于查找服务的操作。如果填写错误, 可能无法正确访问到该服务操作。

Web 服务操作名: 用于接收响应消息的服务操作。

6、TLQ 输入节点

接入方式为 TLQ 方式时, 即对应 RCloud 云 EI 服务的 TongLINKQ 连接器。多输入关联键值、消息格式, 同 DEAgent 输入。



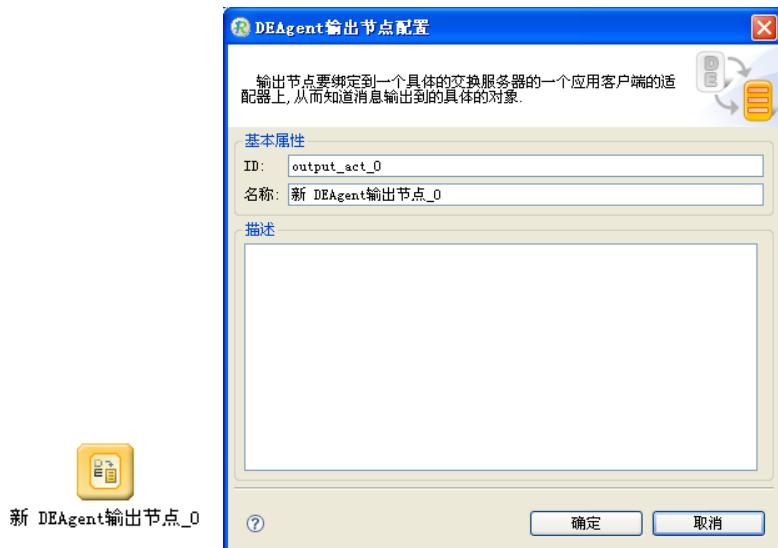
TLQ 输入属性对话框

三、输出节点

输出节点与输入节点相对应, 作为消息向外发送的渠道之一。发送目的地可以为本地应用客户端, 也可以是远程 RCloud 云 EI 服务服务器管理的客户端, 并且在这个客户端, 存在相应的消息处理器负责消息处理, 输出目的地也可以是 MQ 输出队列。

1、DEAgent 输出节点

DEAgent 输出节点对应 RCloud 云 EI 服务的 DE Agent 连接器类型, DEAgent 输出节点属性页面, 如下图所示:



DEAgent 输出

2、MQ 输出节点

MQ 输出节点对应 RCloud 云 EI 服务的 MQ 连接器类型。

注意：在 RCloud 云 EI 服务设计工具中创建 MQ 节点后，要在对应的 MQ 服务器上创建名为“RONEDE. SVRCONN”（可在 RODEConfig.xml 文件中配置）的服务连接通道。

属性如下：



MQ 输出

3、TLQ 输出节点

TLQ 输出节点对应 RCloud 云 EI 服务的 TongLINKQ 连接器类型。

属性如下：



TLQ 输出

四、代理服务封装

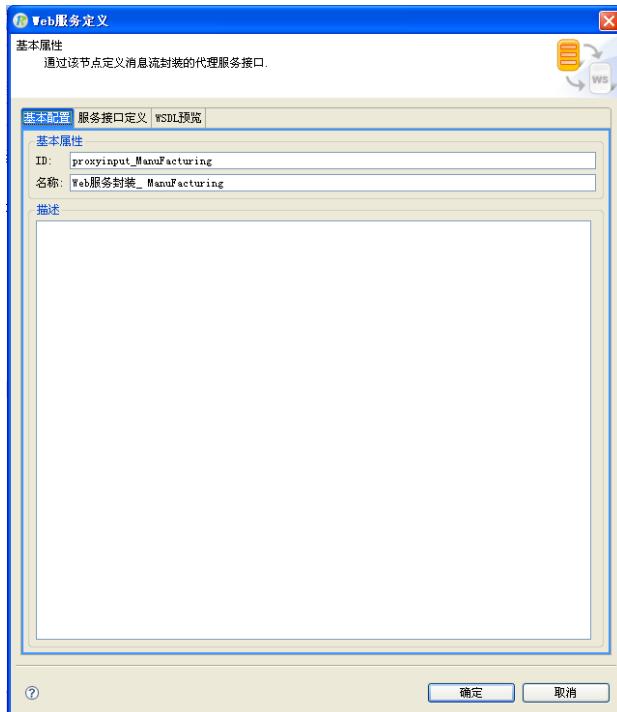
该组中的节点主要用于代理服务流程设计使用。

1、WEB 服务封装节点

用于对业务服务的封装，对外提供的代理服务接口。

属性页面包含三个标签：基本属性、接口设计与映射、预览。

1) 基本配置



基本配置

2) 服务接口定义

A、服务接口的定义方式有两种：

指定服务接口：根据已经设计好的代理服务的 WSDL 文件定义服务接口；

自定义服务接口：用户自己定义服务接口信息。

B、配置服务正常访问所需的信息，包括：

服务名称：所请求的服务的名称；

操作名称：服务提供的方法名称；

命名空间：服务的命名空间，在描述服务的 WSDL 文件中获取；

输入参数：对应服务提供的方法中的输入参数；

输出参数：对应服务提供的方法中的输入参数，XPath 的内容来自请求服务的内容，指定服务接口和自定义服务接口的方式都需要设置 XPath 中内容。注：XPath 中内容从点击  弹出的对话框中选取，选取后如需修改再进行手动修改，不建议从别处拷贝粘贴，否则会丢失 XPath 中命名空间的信息。

异常：对应服务提供的方法中的异常信息。

C、生成消息格式

根据输入输出参数的定义生成代理服务的输入输出的消息格式。生成的消息格式将用于消息格式转换中的消息映射，以及稍后介绍的服务调用节点中的服务调用接口映射。



服务接口定义

3) 预览

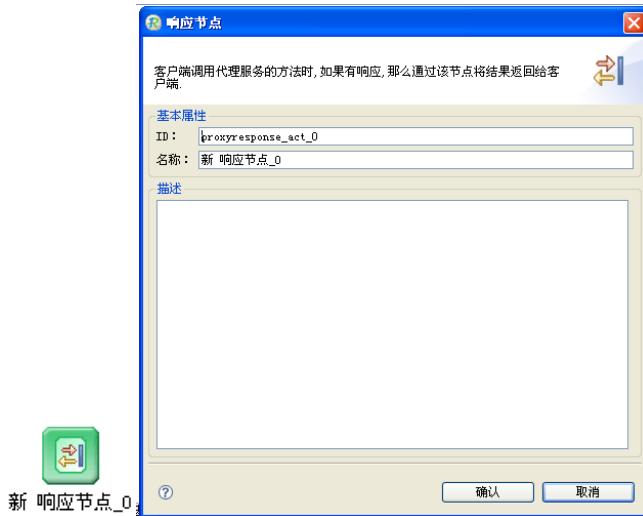
根据服务接口定义所填写的内容生成的 WSDL 预览。



预览

2、WEB 服务响应节点

当服务请求有响应或反馈信息的时候需要加入服务响应节点，属性如下：

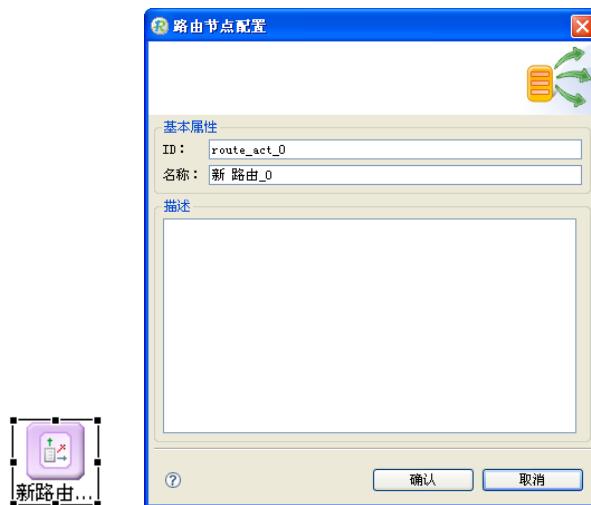


五、普通节点

1、路由节点

路由节点是一种特殊节点，不执行实际业务，也不会对消息数据进行任何存取操作，其目的只是为了控制消息的复杂流向控制。

路由节点属性页面，如下图所示：



路由节点

路由节点没有特殊的属性配置，图示属性在前文『节点操作概览』时已经做了说明。

2、JAVA 扩展节点

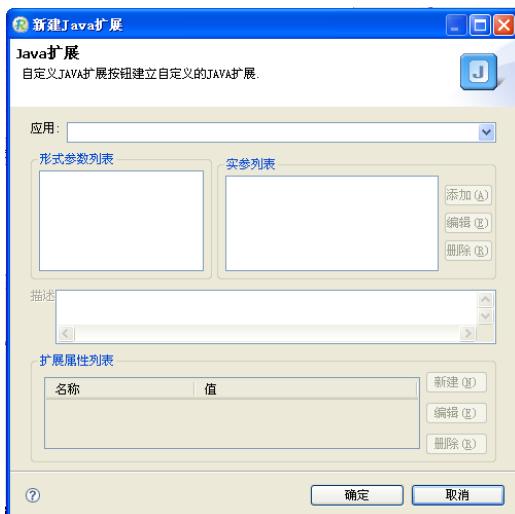
Java 扩展节点是支持用户编程的节点，可以通过 Java 扩展应用来实现流程中的特殊业务。Java 扩展节点属性页面，如下图所示：



JAVA 扩展

Java 扩展属性操作的属性中，要素是『Java 扩展引用』属性。

一个 Java 扩展节点可以引用多个 Java 扩展，图中的列表可以通过『新建』、『编辑』、『删除』三个按钮进行维护，引擎将按照一定顺序依次执行列表中的 Java 扩展。『新建』和『编辑』弹出的属性页面如下所示：



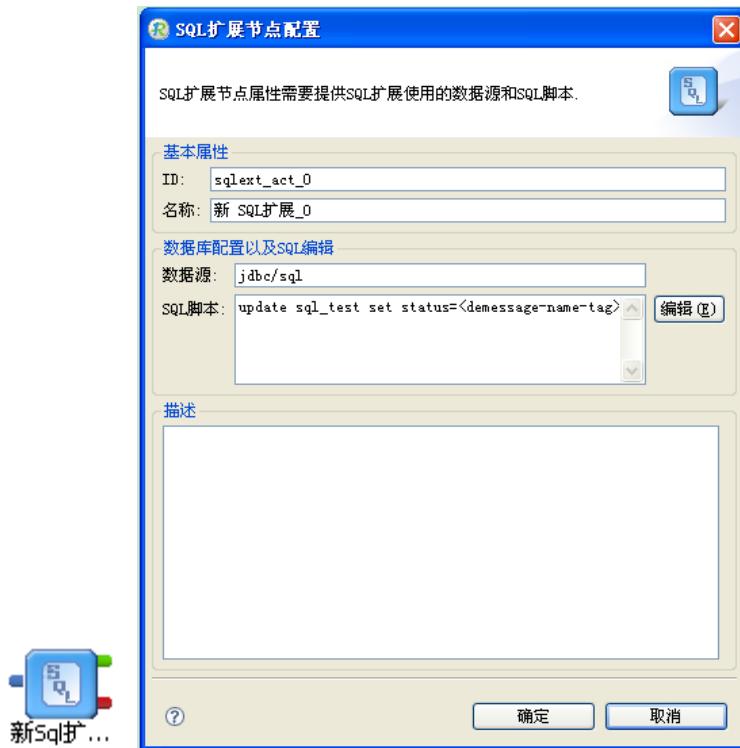
自定义 JAVA 扩展

在『自定义 Java 扩展引用』属性，可以通过下拉列表选择当前消息流包中的 Java 扩展应用。关于如何创建消息流包中的 Java 扩展应用，可以在『3.2.7.7 自定义 JAVA 扩展设置』一节中查看详细内容。图中所示的形参列表、实参列表、扩展属性列表这三个属性，是为版本升级使用，在当前版本填写信息无效。

3、SQL 扩展节点

SQL 扩展节点可以根据流程中的消息数据，动态的执行 SQL 语句，直接对中软国际有限公司

数据库进行操作。SQL 扩展节点属性页面，如下图所示：



SQL 扩展

『数据源』属性，需要选择 SQL 脚本执行的数据库所对应的数据源。这个数据源可以在 DE 的管理控制台进行配置。

『SQL 脚本』属性，可以填写复杂的 SQL 语句，支持用“;”分割的多条 SQL 语句，也支持以“--”开头的 SQL 语句注释标记。点击『编辑』按钮，可以在比较友好的窗口进行 SQL 编辑工作(见<图:SQL 编辑器>)。



SQL 编辑器

需要特殊说明的是，SQL 语句中可以支持消息数据的动态支持，通过以<demessage-name-tag>和</demessage-name-tag>标签起始结束，引擎在执行过程中可以根据标签中的值提取消息中的相应数据。比如下面的例子，使用 XPATH：“/root/customername”，查询得到 XML 消息中的值，插入到数据库表中。

例如：SQL 语句为：

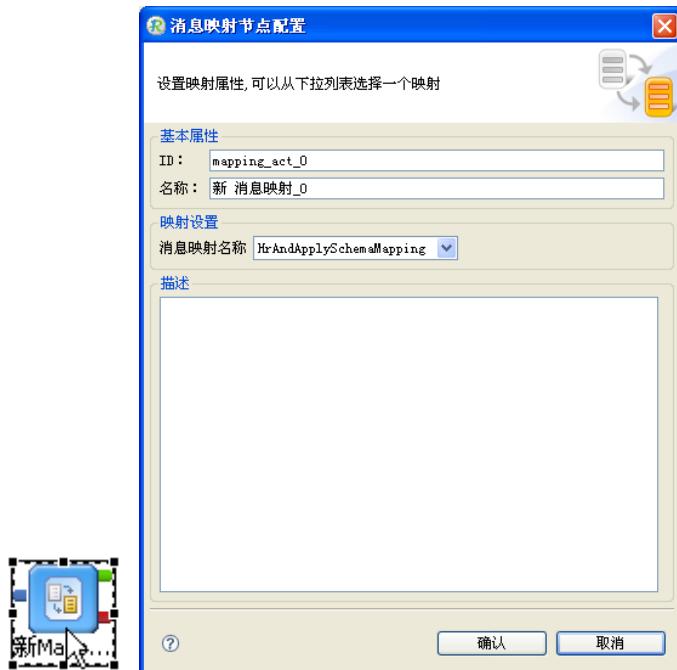
```
INSERT INTO YOUR_TABLE (COLNAME) VALUES
('<demessage-name-tag>/root/customername</demessage-name-tag>
');
```

在引擎运行时，可以从 XML 消息中提取 XPATH 为“/root/customername”对应的值，假设为“sampledata”，则在运行时，会按照下面这样的 SQL 语句实际执行：

```
INSERT INTO YOUR_TABLE (COLNAME) VALUES ('sampledata');
```

请注意：标签的 XPATH 必须是全路径名，比如 XPATH：“/root/customername”，虽然可以写“//customername”，也可以搜索到 customername 元素的值，但是，DE 引擎不会正确处理“//customername”。

4、消息映射节点



消息映射节点

消息映射节点是用来在 RCloud 云 EI 服务引擎中进行消息映射的，可以选择设置一个消息映射定义。映射下拉菜单选项是来自消息映射定义的。

5、交换服务节点

交换服务节点与输出节点类似，作为消息向外发送的另外一种渠道。发送目的地必须是远程的 RCloud 云 EI 服务服务器，并且在这个远程服务器端，存在相应的消息流负责消息接受和处理。交换服务节点属性页面，如下图所示：



交换服务

『消息流 ID』属性：填写远程 RCloud 云 EI 服务服务器端，负责接受这一消息输出的消息流 ID。

『消息流版本』属性：填写远程交换服务器端，负责接受这一消息输出的消息流版本。

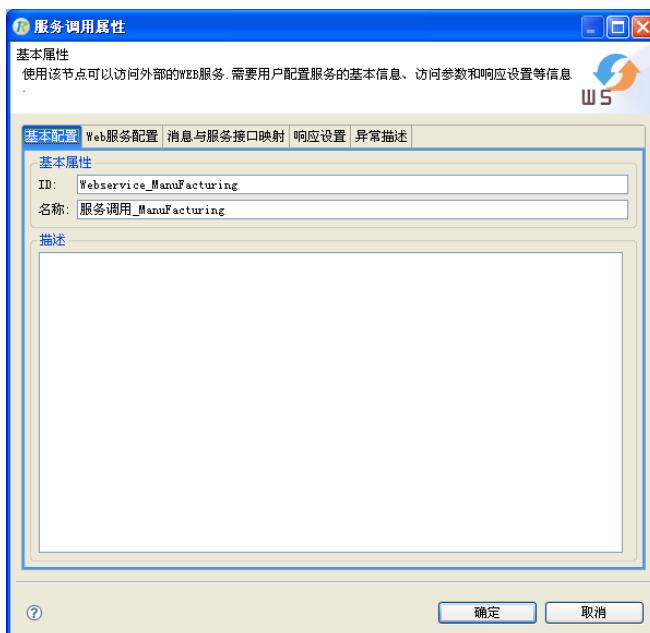
消息流 ID 和消息流版本属性，可以唯一确定远程交换服务器端的消息流定义。

6、服务调用节点

此节点用以访问 web service 资源，在属性界面中需要填写 web service 资源信息和选择响应方式。RCloud 统一资源库可以选择已经配置的，也可以重新配置一个新的。服务调用节点信息配置界面如下：

1) 基本配置

包括服务调用的 Id 和名称。



服务调用节点——基本配置

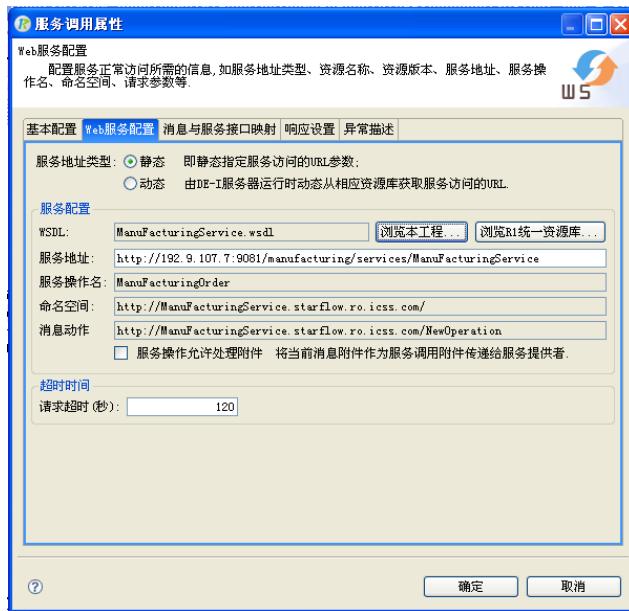
2) Web 服务配置

指定服务地址有两种类型：静态和动态。这两种方式的主要区别在于：静态方式是在流程设计时指定服务地址，如果一旦服务访问地址发生变更，需要重新修改流程；动态方式是在运行时从 RCloud 统一资源库中动态查

找服务访问地址。

静态类型：

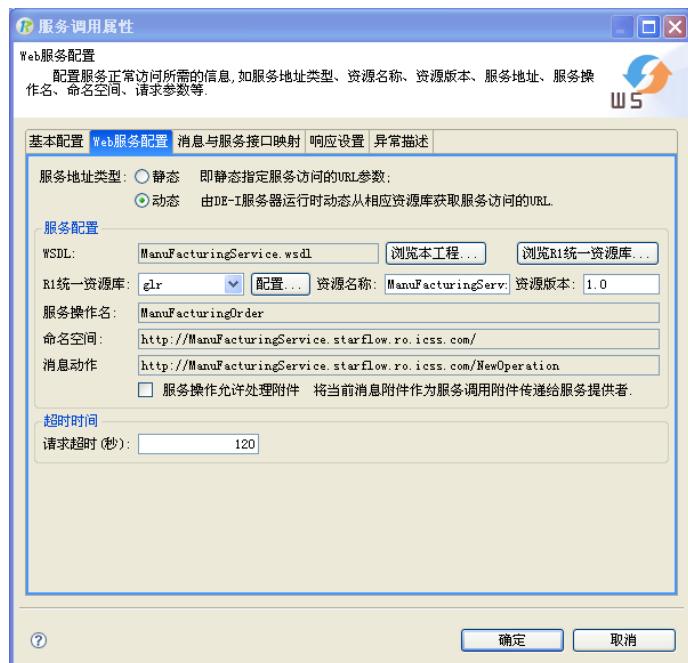
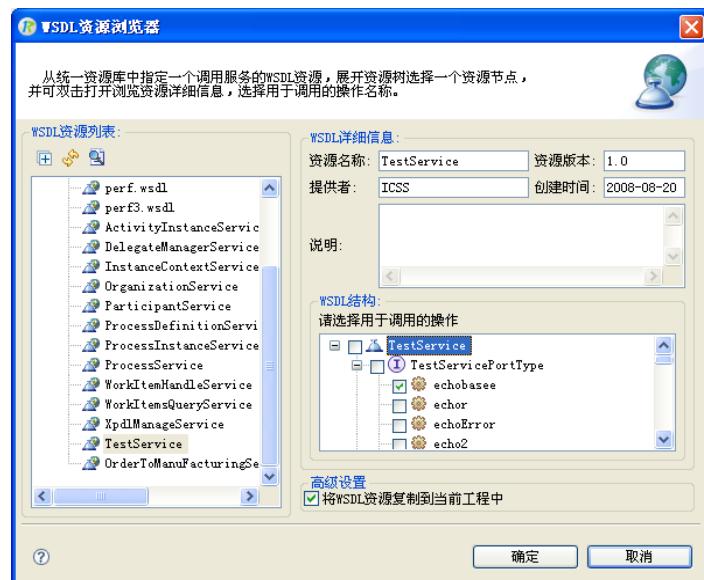
点击“浏览远程...”或“浏览本地...”按钮，从远程或本地资源库选择一个 WSDL 资源，选择要访问的服务操作，点击“确定”。如果该服务操作能够接收带附件的消息，则勾选“服务操作允许处理附件”的选项。



动态类型：

点击“浏览远程...”或“浏览本地...”按钮，从远程或本地资源库选择一个 WSDL 资源，选择要访问的服务操作，点击“确定”。如果该服务操作能够接收带附件的消息，则勾选“服务操作允许处理附件”的选项。

统一资源库选择该服务注册的资源库，如果列表中没有值，点击“配置...”，配置一个统一资源库地址；资源名称项和版本项填写统一资源库中注册的该服务的服务名称和版本。



服务调用节点——WebService 配置

3) 消息与服务接口映射

用于指定服务访问参数的取值（以 XPATH 方式指定）。

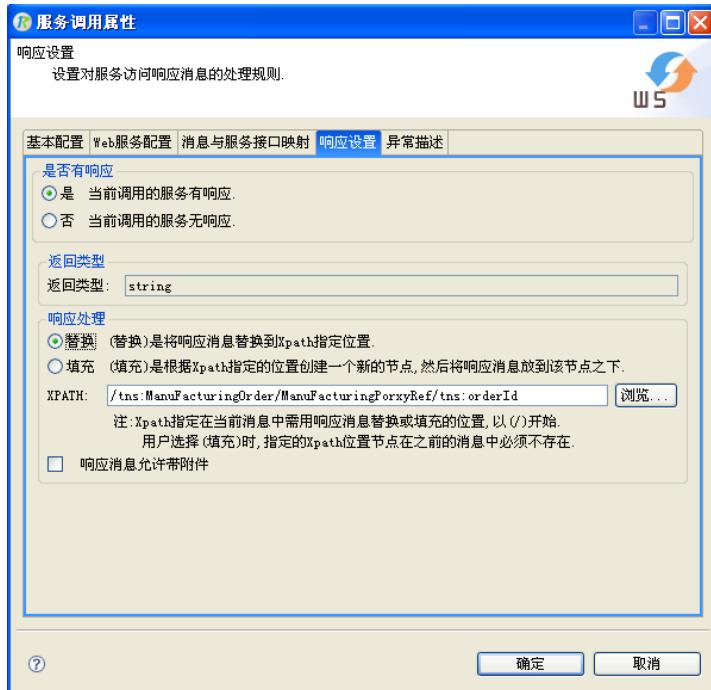
通过对消息建立映射来指定服务请求参数，用户在不知道 XPATH 语法规则的情况下也能完成请求参数指定。



服务调用节点——消息与服务接口映射

4) 响应处理

设置对响应消息以及附件的处理



服务调用节点——响应处理

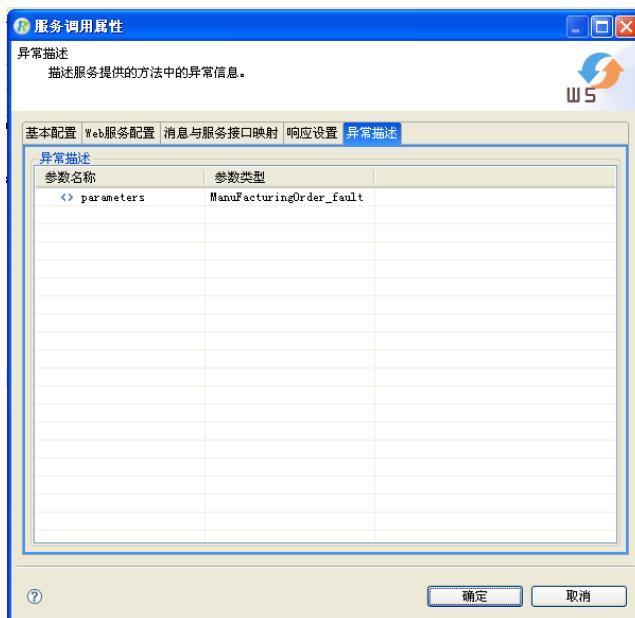
返回类型：所访问服务操作的返回值类型，目前支持的返回类型有简单类型和 Bean 类型。

响应处理：用户选择对响应消息的处理，有“替换”和“填充”两种选择。“替换”表示将响应消息替换 XPATH 指定位置的值；“填充”表示将响应消息填充到 XPATH 指定位置。填充所指定的 XPATH 位置在当前消息结构中必须不存在（但根节点必须和当前消息结构一致）。注：Xpath 中内容从点击 **浏览...** 弹出的对话框中选取，选取后如需修改再进行手动修改，不建议从别处拷贝粘贴，否则会丢失 Xpath 中命名空间的信息。

响应消息允许带附件选项：如果勾选该选项，如果响应消息带有附件，RCloud 云 EI 服务服务器会将该附件引用添加到消息中，并将该附件传送到消息最终目的地。如果没有勾选，即使响应消息中带有附件，也不会处理。

5) 异常描述

描述服务提供的方法中的异常信息。



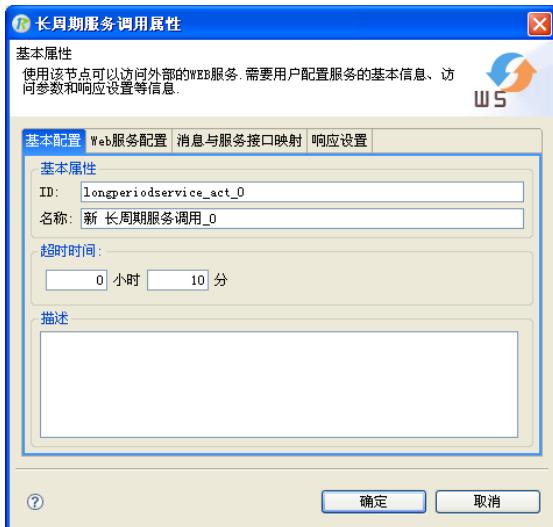
服务调用节点——异常描述

7、长周期同步调用节点

1) 基本配置

需要填写长周期服务调用的 Id 和名称，如果业务处理的时间比较长，而又要保持处理的同步，那么在传输过程中选择此节点。属性界面的填写信息基本与服务调用节点相同，不过需要设定超时时间。消息通过长周期

服务调用节点发送后开始，如果在设置的超时时间内能接收到响应消息，认为该响应消息有效，并继续往下执行。如果时间超过了设置的超时时间，即使接收到响应消息，也不会处理。



长周期服务调用——基本配置

2) Web 服务配置

同服务调用节点的 Web 服务配置。

3) 消息与服务接口映射

同服务调用节点的消息与服务接口映射配置。

4) 响应设置

同服务调用节点的响应配置，但长周期必须有响应。

六、其他节点

1、错误处理节点



错误处理节点

错误处理节点是用来引用在 JAVA 扩展中定义的 JAVA 扩展类。当一个中间节点（如服务调用节点）处理出现异常后，如果该节点设计有错误处理节点分支，系统会自动执行流程注册的错误处理代码。用户可以扩展实现符合自己需求的错误处理逻辑。有关错误处理类的开发请参考《RCloud4.1Exchange-Integration 产品开发手册》中的说明。点击“新建”按钮，可以新建一个 java 扩展，在“应用”列表中可以进行选择：

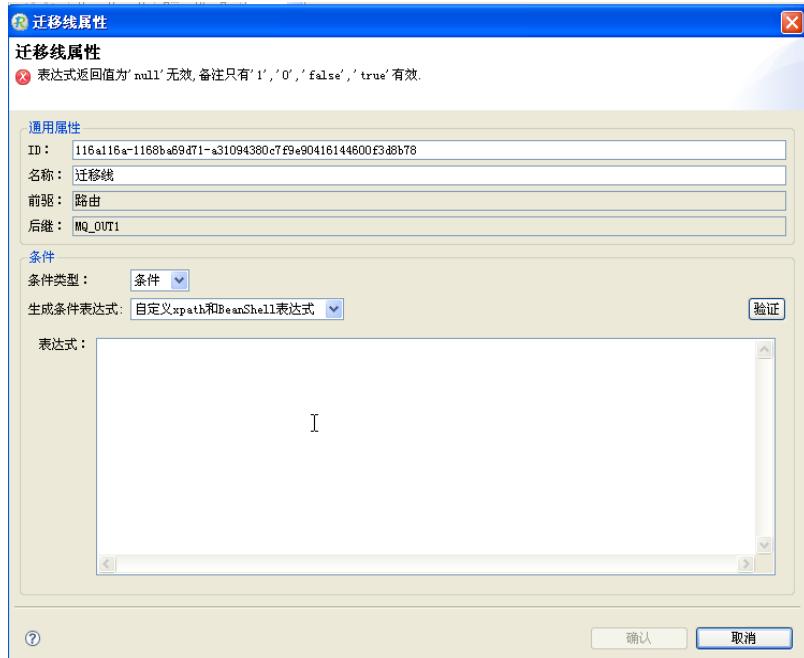


七、迁移线

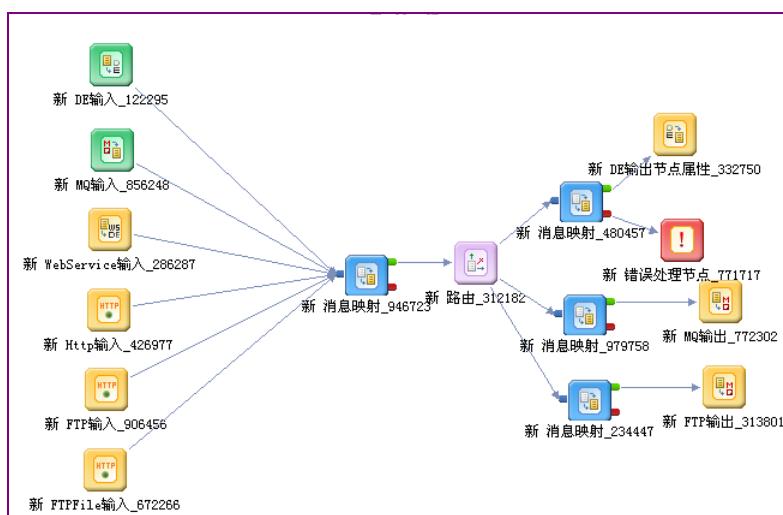
迁移线在 RCloud 云 EI 服务设计工具中表现为所有消息处理节点之间的连接线段。引擎通过这些线段来控制流程的流转顺序与方向。



迁移线



迁移线属性



多迁移线进入

只有消息映射节点能够有多迁移线进入(注:多输入节点做消息合并只能连接到消息映射节点)。

迁移线的“表达式”可以使用 BeanShell 来写，当返回 true 时，消息会送到迁移线指向的节点，否则不会送到该节点。

迁移线属性配置中条件表达式有三种类型：

1、自定义 Xpath 和 BeanShell 表达式：这种条件可以用 BeanShell 脚本来编写；可参照下面的脚本来写：

```
xpath __operand1=/Book/SN;

List __operand2 = new java.util.ArrayList();

__operand2.add("HK");

boolean stringOperandEval_expl(java.util.List
operand1,java.util.List operand2) {

boolean __result=false;

if(!operand1.isEmpty() && !operand2.isEmpty()){

String s1=(String)operand1.get(0);

String s2=(String)operand2.get(0);

__result=(s1.indexOf(s2)>=0);

}

return __result;

};

boolean expl_result=
stringOperandEval_expl(__operand1,__operand2);

return expl_result;
```

BSH(Beanshell)是一种脚本语言，一种完全符合 Java 语法的 Java 脚本语言，并且又拥有自己的一些语法和方法。BeanShell 能理解标准的 JAVA 语句，表达式，和方法宣告。语句和表达式的内容可以是：变量，宣告，赋值，方法调用，循环，条件等。

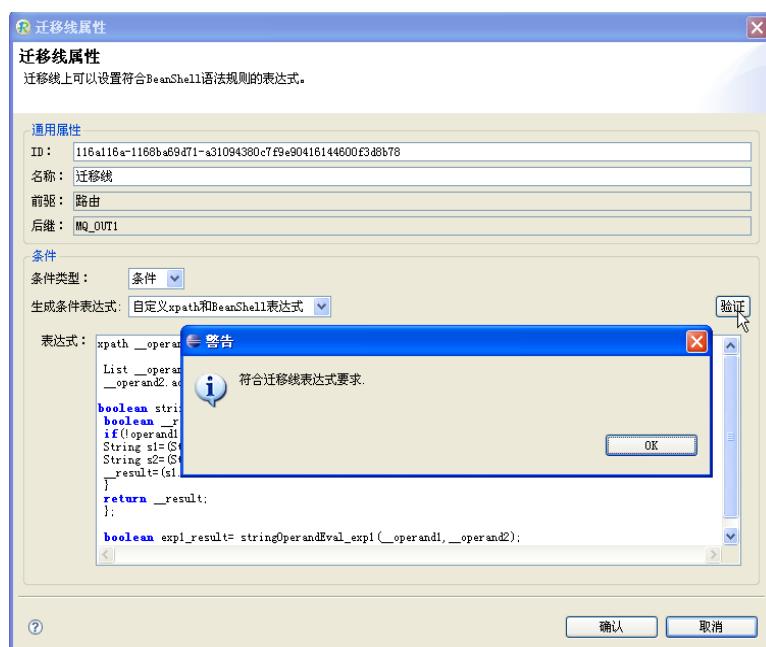
BSH 是一种松散类型的脚本语言(这一点和 JavaScript 类似)。也就是说，可以在写脚本时不进行变量类型的宣告(在原始数据类型和对象都可以)。更详细的语法介绍可以参考：<http://www.beanshell.org>。

例：使用 XPATH：“/销售量/华北销售量/河北销售量/石家庄销售量”从 xml 消息中取得值，如果包含一个“500hebei1”，则消息可以送到迁移线指向的节点，否则，消息不会流到该节点。

注意：如果使用 xpath，则首先需要在脚本开始声明 xpath 变量，以 xpath 开头，变量名命名规则与 java 的相同。声明 xpath 变量语句每行均以分号“;”结尾，而且每行只能声明一个变量，xpath 路径为全路径。

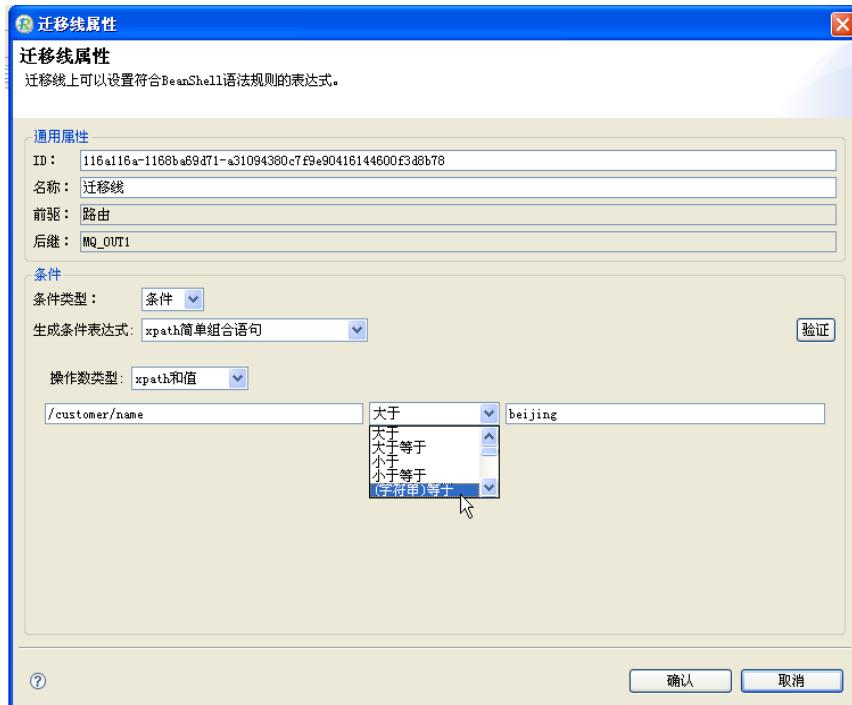
非 xpath 声明部分，语法完全按照 BSH 的脚本语法，返回值是字符串“true”或者布尔值 true，或者非零的数字，也认为条件成立。可以使用或者前面声明的 xpath 变量，变量的数据类型均为 List 类型，List 的每个元素为 String 类型，也就是根据 xpath 变量的 xpath 从 XML 消息中查询出的数据。

编写脚本后可以验证脚本的正确性，点击“验证”来检测，如下图：



迁移线属性

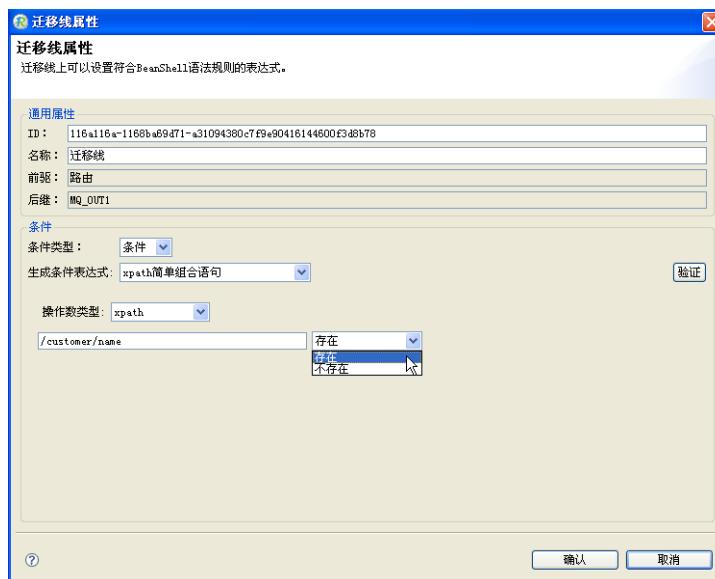
2、xpath 简单组合语句：选择这个条件表达式后，出现如下对话框：



迁移线类型

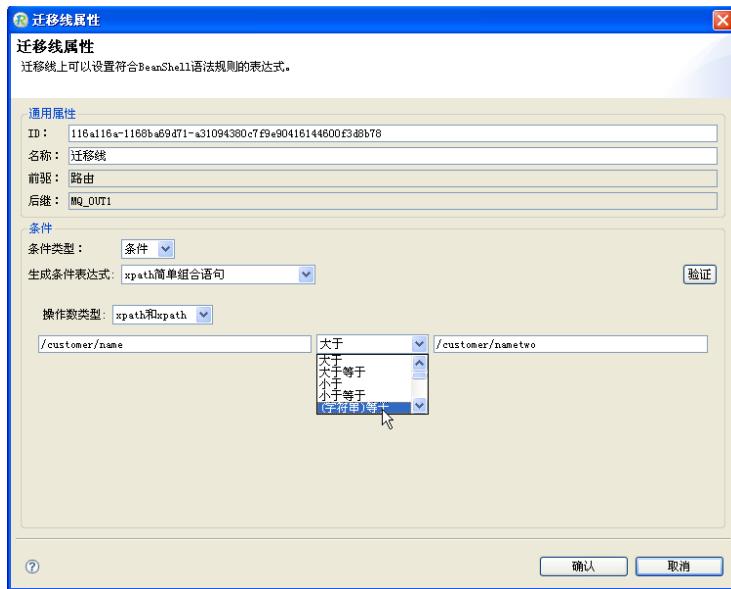
操作数类型有 xpath ,xpath 和 xpath, xpath 和值, 值和 xpath 这几类。选择每种类型的填写方法各不相同：

1) 选择“xpath”时：当满足填写的 xpath 值存在或者不存在时条件成立，能发送消息。如下图：



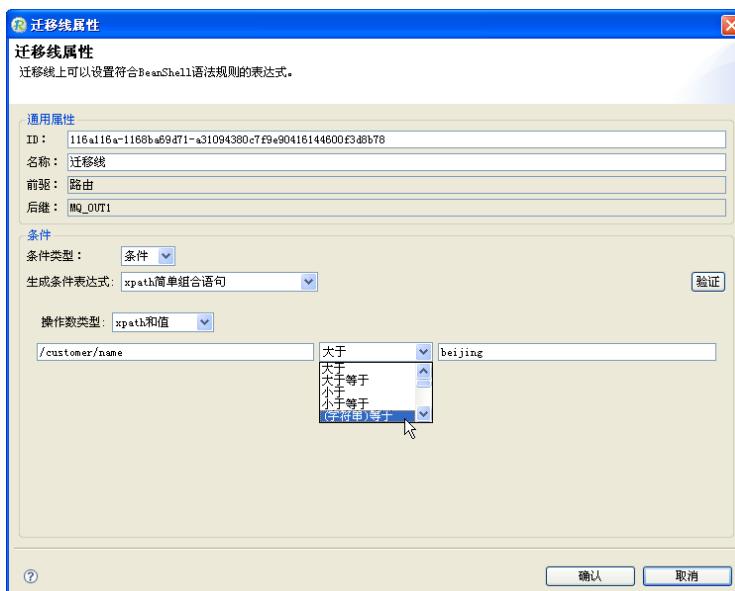
迁移线类型

2) 选择“xpath 和 xpath”时，在输入框中填写的两个 xpath 值满足中间所选条件时，成立，消息能发送，如下图：



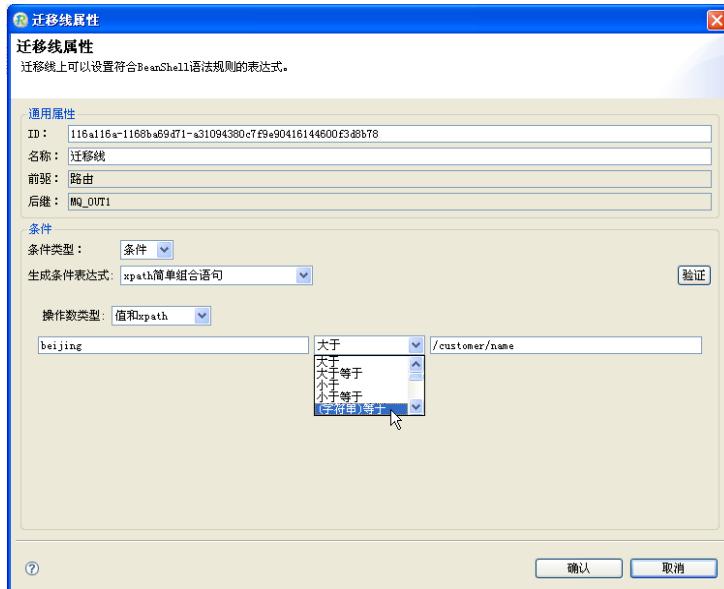
迁移线属性值

3) 选择“xpath 和值”时，在输入框中第一个填写 xpath，最后面填写值，当 xpath 的值和填写的值满足中间所选条件时，成立，消息能发送，如下图：



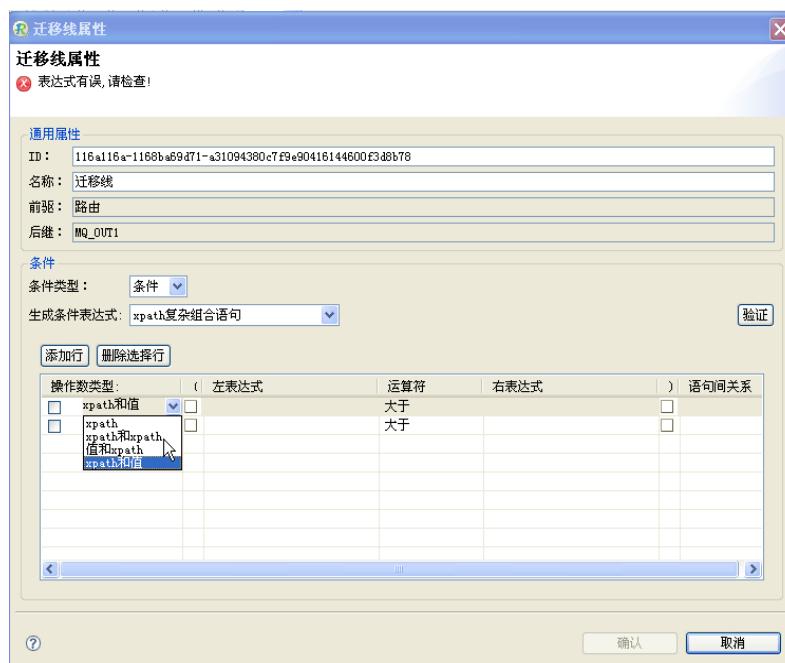
迁移线属性值

4) 选择“值和 xpath”时，在输入框中第一个填写值，最后面填写 xpath，当 xpath 的值和填写的值满足中间所选条件时，成立，消息能发送，如下图：



迁移线属性值

3、xpath 复杂组合语句：这种条件表达式是多种“xpath 条件语句”的组合，当都满足时，条件才成立。如下图：



添加迁移线表达式

可以选择“添加行”来添加一个条件表达式，在添加的列表中“操作数类型”列可以选择不同的操作数类型，选择好操作数类型后，填写方式和“xpath 简单组合语句”相同。

注：当设计的代理服务消息流中有条件分支时，只支持“有条件，否则”这种

组合，不支持“有条件，无条件”，即条件分支的迁移线的条件不能设为“无条件”。

3.2.5.4 关于 Dex 文件

Dex 文件不允许手动更改文件内容，RCloud 云 EI 服务设计工具 3.5.1 以后版本可以编辑打开前一版本设计 Dex 文件，但可能会存在部分问题。如果打开出错只能参照以前流程图使用新版本重新设计。

3.2.6 代理服务设计与发布

代理服务的功能：通过消息流封装服务的访问并对外提供代理服务访问接口，屏蔽对后端服务的直接访问，实现服务中介的功能。

3.2.6.1 创建流程

创建代理服务流程和创建普通消息流程一样，只需在新建时，选择“对服务进行封装”（如下图）



如果已经设计好了代理服务的 WSDL 文件，则在创建消息流的时候就可以指定代理服务的接口，然后点击“Finish”完成代理服务消息流的创建，否则，直

接点击“Finish”完成：



3.2.6.2 设计代理服务接口

用户设置对外暴露的代理服务接口，外部访问通过该代理服务接口访问真正的业务服务。

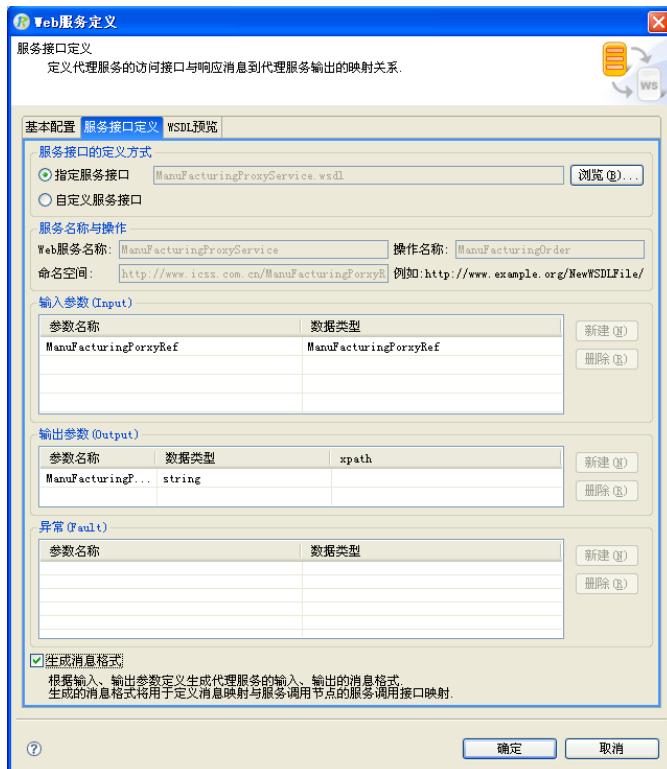
双击流程的 web 服务封装节点，弹出的属性窗体，有三个 tab 页：基本配

置、接口设计与映射设计、预览。基本配置 tab 页主要填写 web 输入节点的 ID、名称；预览 tab 页主要是查看设计的代理服务的 WSDL；接口设计与映射设计设计代理服务的名称、操作、命名空间、以及输入输出参数等。

一、基本配置

填写当前节点的 ID 和显示名称，以及描述信息。

二、服务接口定义



1、服务接口的定义方式：

指定服务接口：根据已经设计好的代理服务的 WSDL 文件定义服务接口；

自定义服务接口：用户自己定义服务接口信息。

2、服务名称与操作：

主要填写代理服务的名称、操作方法名、操作命名空间

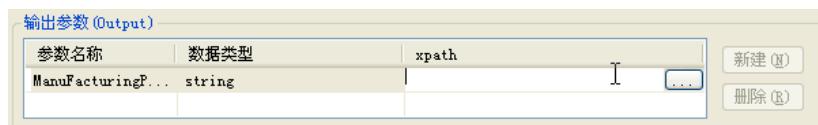
3、输入参数：

点击右边的“新建”按钮用于增加一个输入参数，可以编辑参数名、选择参数数据类型，数据类型支持 boolean,in,double,float,String,以及用户设计的 Schema。如果用户设计类型，首先需设计一个消息格式，然后在这里

数据类型选择“浏览...”，弹出数据格式选择列表，选择所需的格式文件。

4、输出参数：

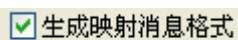
操作和输入相似，但是可以指定消息中的一部分作为响应消息返回（通过 XPATH 指定），如果不指定 XPATH，则将全部消息作为响应返回。注：Xpath 中内容从点击 [...] 弹出的对话框中选取，选取后如需修改再进行手动修改，不建议从别处拷贝粘贴，否则会丢失 Xpath 中命名空间的信息。



5、异常

对应服务提供的方法中的异常信息。

6、生成映射消息格式选择框：



选择“生成映射消息格式”，会在消息格式定义目录生成一个代理服务请求消息格式 XSD 文件。



三、WSDL 预览

查看设计的代理服务的 WSDL 描述，主要检查设计的代理服务的 WSDL 描述是否正确。

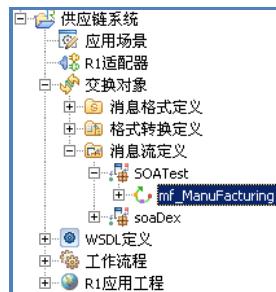
3.2.6.3 配置业务服务访问

参考 3.2.5.3 中的 服务调用节点的相关说明。

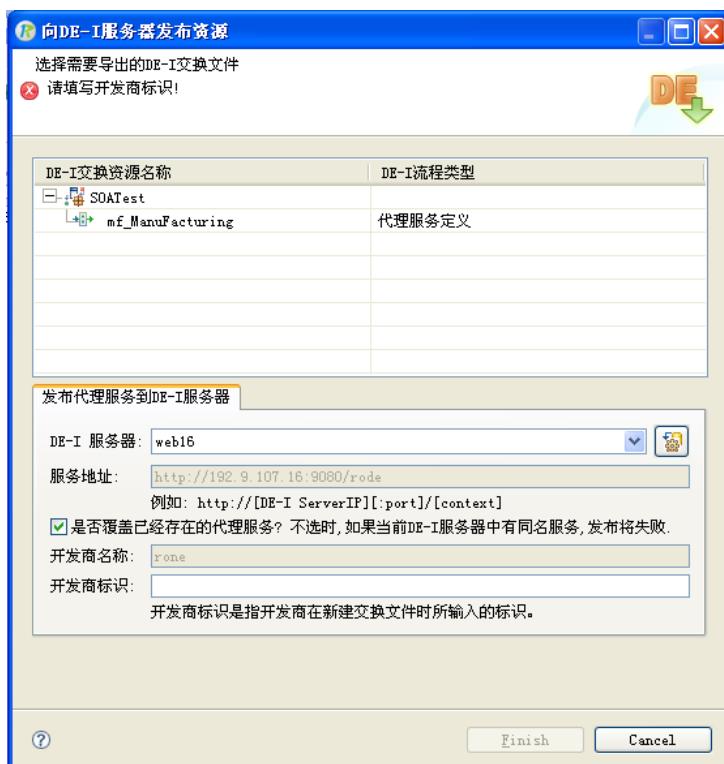
3.2.6.4 代理服务发布到 RCloud 云 EI 服务服务器

代理服务流程设计完成后，必须发布到 RCloud 云 EI 服务服务器上才能被使用。

第一步：



鼠标选中要发布的代理服务流程（如上图的 SOATest），点击鼠标，选择鼠标右键菜单中的“发布消息流定义”。弹出如下页面：



第二步：

指定代理服务流程发布的 RCloud 云 EI 服务服务器，填写开发商标识，并选择“是否覆盖已经存在的代理服务？”，点击“完成”。如果 RCloud 云 EI 服务服务器下拉菜单中没有所需的连接，点击右边的配置按钮配置所需连接（参考 [3.2.2.2](#)）。

第三步：

当该代理服务之前已经在 RCloud 云 EI 服务服务器上存在时，需要重启 RCloud 云 EI 服务服务器以使对该代理服务的更改生效。

3.2.6.5 代理服务发布注意事项

当使用 jdbc 方式更新一个已启动的代理服务时，需要先在管控中停掉该代理服务。

3.2.6.6 代理服务发布到 RCloud 统一资源库

设计代理服务时，同时会将该代理服务的 WSDL 描述保存到当前工程下的 WSDL 定义目录下，用户可以将该 WSDL 资源发布到 RCloud 统一资源库，其他用户可以通过 RCloud 统一资源库访问该代理服务。

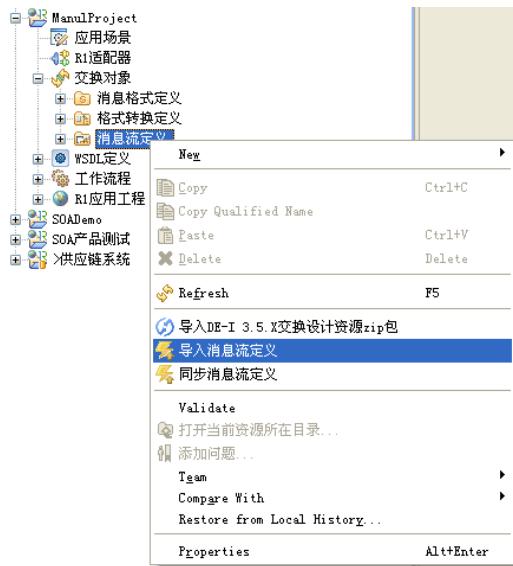
在 WSDL 定义目录，选中要发布的代理服务 WSD 资源，打开鼠标右键菜单 → 发布至 RCloud 统一资源库。

3.2.7 导入、导出设计资源

3.2.7.1 导入消息流到 RCloud Studio

需要将已经部署到 EI 服务器的流程导入到 RCloud 云 EI 服务设计工具中进行编辑，可以进行如下操作：

在当前 RCloud 工程下，右击“消息流定义”弹出右键菜单。如下图所示：



导入菜单

导入消息流定义：是从文件系统中导入消息流，如下：



DE 流程资源导入

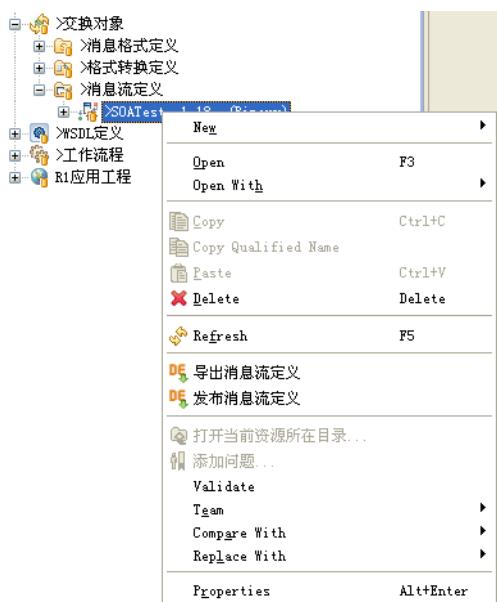
同步消息流定义：从 EI 服务器中的消息流列表中导入需要的消息流，如下：



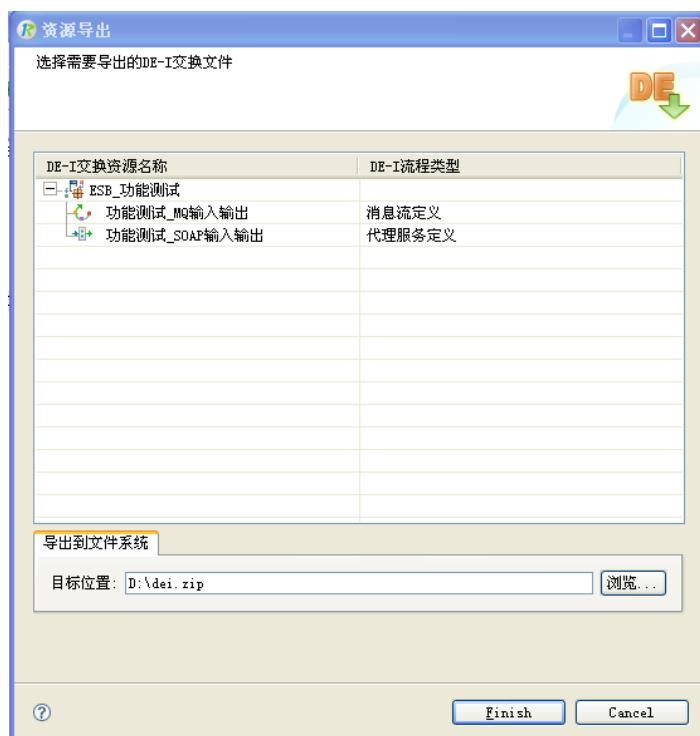
选择 EI 服务器(在 Connection 视图中创建), 填写“开发商标识”; “开发商标识”是在新建消息流文件时创建的, 然后点击“同步”按钮, 那么使用该“开发商标识”创建的、并且保存到数据库中的 RCloud 云 EI 服务文件会在流程文件列表中显示出来。选择一个要导入的消息流文件, 点击 **Next >** 按钮, 即可以从数据库中导入消息流文件到工程中。

3.2.7.2 导出或发布 Dex 资源文件

在当前 RCloud 工程下, 选择要导出或发布的 Dex 资源文件, 点击鼠标右键菜单的 Export 菜单, 在出现的对话框中选择 Export, 如下图所示:

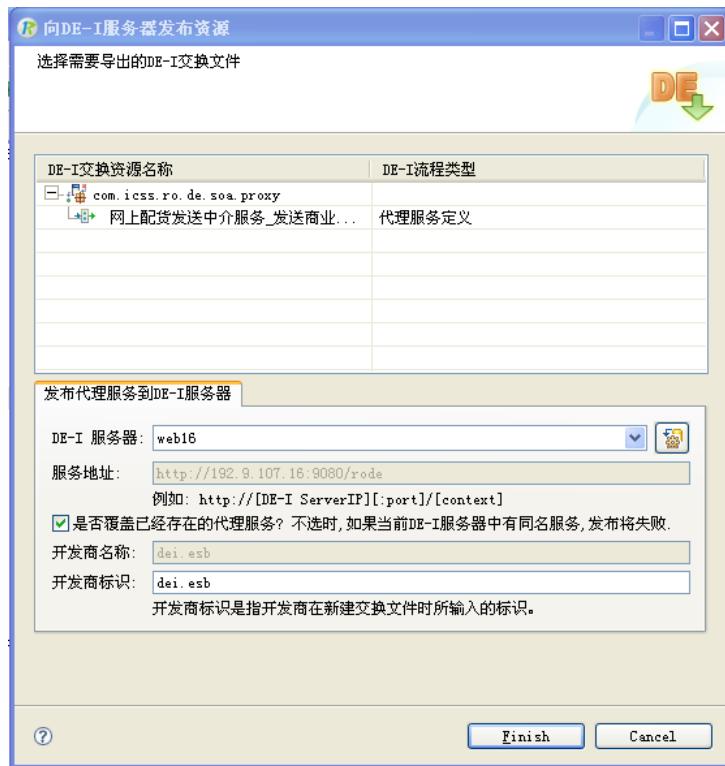


1、导出消息流定义：点击上图中的“导出消息流定义”，如下图，然后指定导出的目标位置，点击“Finish”完成。



导出界面

2、发布消息流定义：

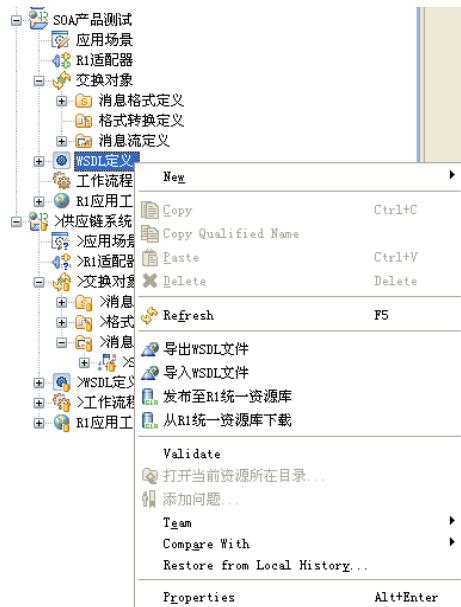


发布消息流定义至 RCloud 云 EI 服务服务器

注：向 RCloud 云 EI 服务服务器发布消息流定义时，如果该消息流之前在 RCloudEI 服务器上已经存在了，则需要重启 RCloud 云 EI 服务服务器才能生效。

3.2.7.3 导入导出 WSDL 资源到 RCloud Studio

右击工程中的“WSDL 定义”，弹出邮件菜单，如下图：



1、导入 WSDL

(1) 在 RCloud 项目管理器的 RCloud 工程中, 右键点取 WSDL 定义组件-->导入 WSDL 文件。

(2) 选择本地 WSDL 文件资源的路径, 点 **Finish**, 此 WSDL 文件以及其引用的相关的 WSDL 将被导入 WSDL 定义下, 引用的 XSD 文件将以消息格式定义的形式被引入到消息格式定义下。

如准备导入文件 ConfigService.wsdl, 其引用了文件 configuration.xsd, 导入到系统后的结构为下图所示:



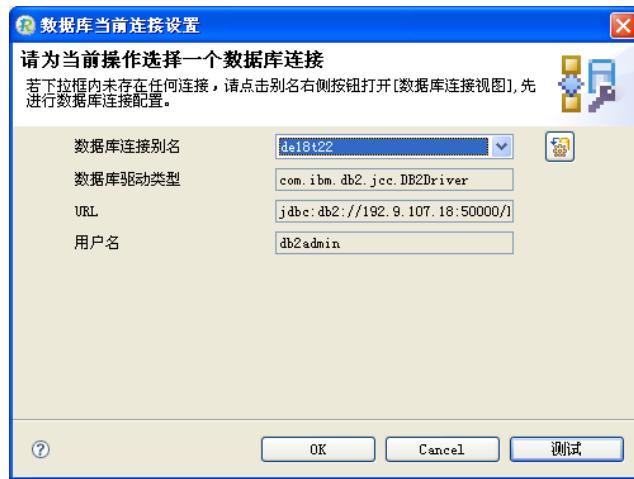
2、导出 WSDL

(1) 在 RCloud 项目管理器的 RCloud 工程中, 右键点取 WSDL 定义组件导出 WSDL 文件。

(2) 选择导出文件夹路径, 点 **Finish**, 即可完成导出 WSDL 资源。

3.2.7.4 数据库连接设置

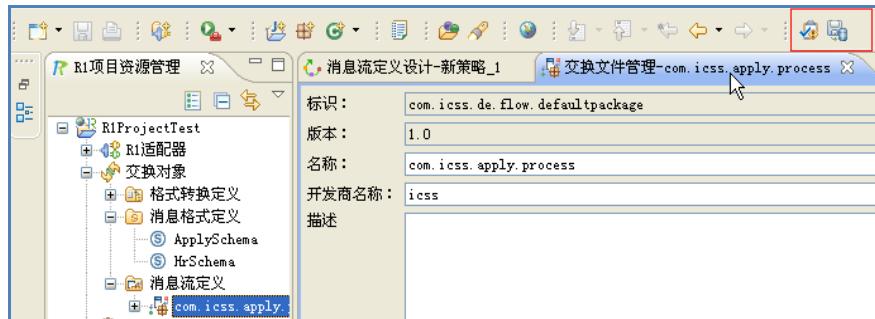
数据库连接设置可以切换当前设计工具设置的数据库连接。数据库连接(别名)的配置位于<数据库配置>中, 详见 3.2.2 章节“数据库配置”。如图所示:



数据库连接

3.2.7.5 保存到数据库

消息流文件新建、编辑后，不需要验证就可以保存覆盖原来的 dex 文件。



交换流程

进行有效性验证、没有错误以后，可以点击 (保存到数据库)按钮保存到数据库(需要保证 DE 配置数据库连接正常)。点击该按钮后，弹出“资源导出”对话框，在这里需要选择一个数据库连接(如果还没有配置数据库连接，可以点击 旁边的 进入数据库配置界面进行配置)，输入在新建 dex 文件时输入的“开发商标识”。设计工具验证“开发商标识”无误后就会把 dex 设计保存到数据库中。另外还提供了将设计文件保存到本地的功能，只要指定导出的目标位置本地路径。

如果在消息格式定义列表中大于一个消息格式的设置为“创建模型对应表”时(详见<图: Schema 定义保存数据库选项>)，保存出现对应表对话框。可以指定

或创建一个数据库的 Schema，否则直接保存到数据库，没有上述描述。



消息格式操作

保存成功后会弹出对话框，如果保存失败，需要根据提示信息排除错误。

如下图所示的错误信息：表示在 Connection 视图中创建的 DE 配置数据库连接错误，需要在 Connection 视图中重新打开该连接。



错误信息

注：向 RCloud 云 EI 服务服务器发布消息流定义时，如果该消息流之前在 RCloudEI 服务器上已经存在了，则需要重启 RCloud 云 EI 服务服务器才能生效。

3.2.7.6 验证文件有效性

在消息流设计的过程中，可以随时点击编辑器工具栏的 (验证有效性) 按钮，以检验流程设计的正确性。在保存到数据库时，需要首先通过有效性验证，验证通过才能保存至数据库中。对于一个示例消息流，点击 按钮，可能会出现如下验证对话框：



DE 设计验证

如上图所示，该消息流设计还有错误，那么就不能保存到数据库中(但保存到文件不需要验证)。对于消息流，必须满足如下条件，方能验证通过：

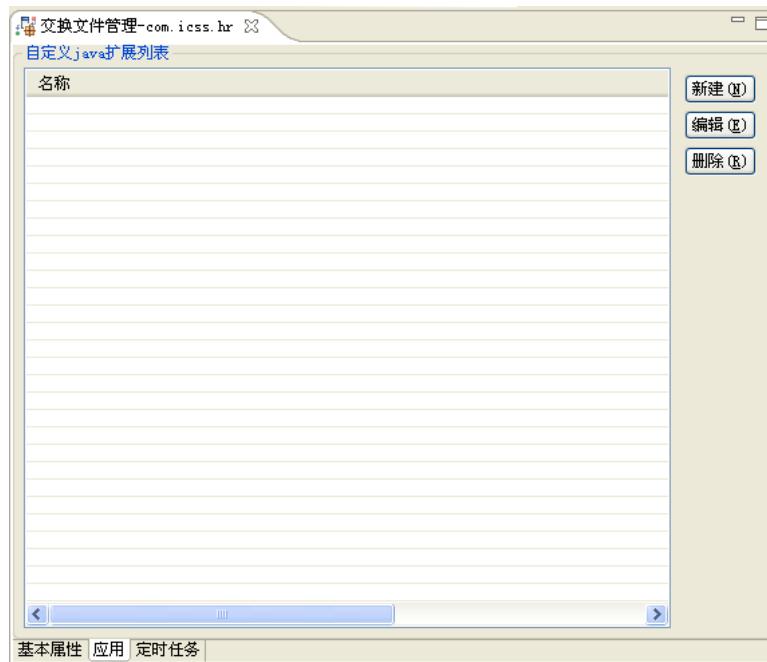
- 1、必须有输入活动
- 2、只有输入活动前面不能有迁移线，默认不用验证。
- 3、输入活动后面必须有迁移线，而且默认状态下只能有一条迁移线
- 4、除了输入活动外，其他的活动前面都得有迁移线
- 5、“SQL 扩展节点”必须选择一个数据源。
- 6、“SQL 扩展节点”的 SQL 脚本为必填项。
- 7、“JAVA 扩展节点”必须至少引用一个“自定义”JAVA 扩展。

3.2.7.7 自定义 JAVA 扩展设置

开发商可以自己定义 JAVA 类和方法，实现自己需要的功能，自定义的 JAVA 类和方法作为“自定义 JAVA 扩展”，可以添加到“自定义 JAVA 扩展”列表中。“Java 扩展节点”需要引用自定义的 Java 扩展。当消息流引擎处理到“Java 扩展节点”时，

通过 JAVA 的“反射”机制，就可以执行该 JAVA 类方法的操作。自定义的 JAVA 扩展类需要继承

com.icss.ro.de.server.flow.engine.impl.WfAbstractJavaExtensionApplication 抽象类，具体信息参看《RCloud DataExchang-Integration 开发手册》。点击消息流包编辑器界面底部的“应用”，出现如下对话框：



自定义 JAVA 扩展

点击 新建 按钮，可以新建 JAVA 扩展，如下图所示：



新建 JAVA 扩展

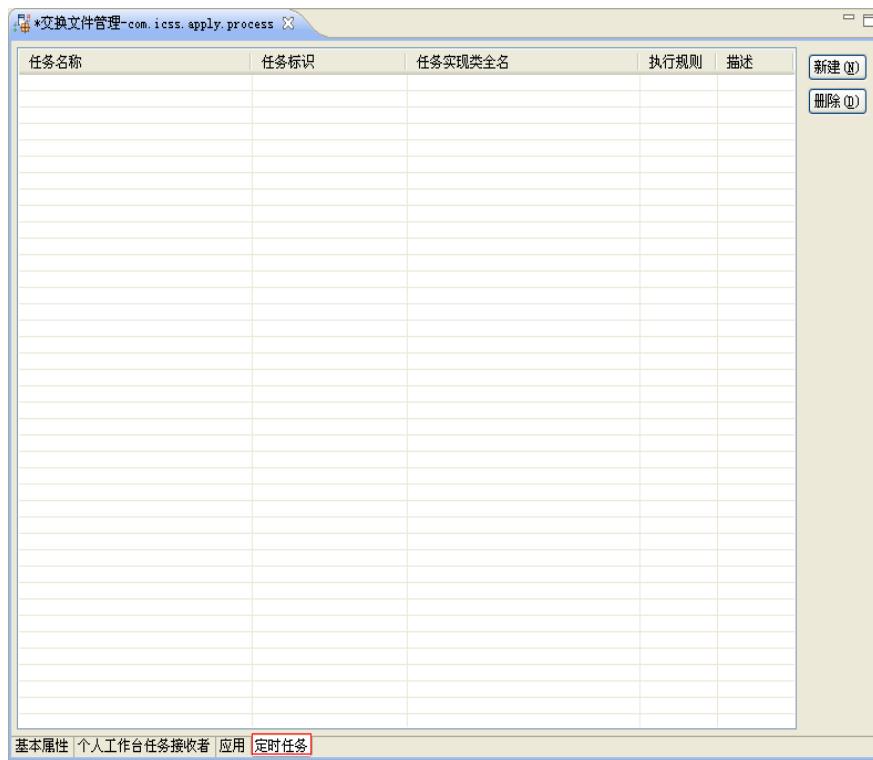
如上图所示，该类：

com.icss.ro.de.demo.targetapp.javaext.ErrorHandlerJavaExtApp 继承自
com.icss.ro.de.server.flow.engine.impl.WfAbstractJavaExtensionApplication，方法
名必须是： run。(具体信息参看《RCloud Exchange-Integration 开发手册》)。

形式参数列表暂时无用不需定义。可能在版本升级时会使用该特性。如果
需要定义 JAVA 扩展使用的额外的扩展属性，比如阀值、JAVA 扩展使用的数据
源等，那么就需要在“扩展属性列表”中定义。不过，需要额外的扩展属性的情况
在目前应用还很少用到。

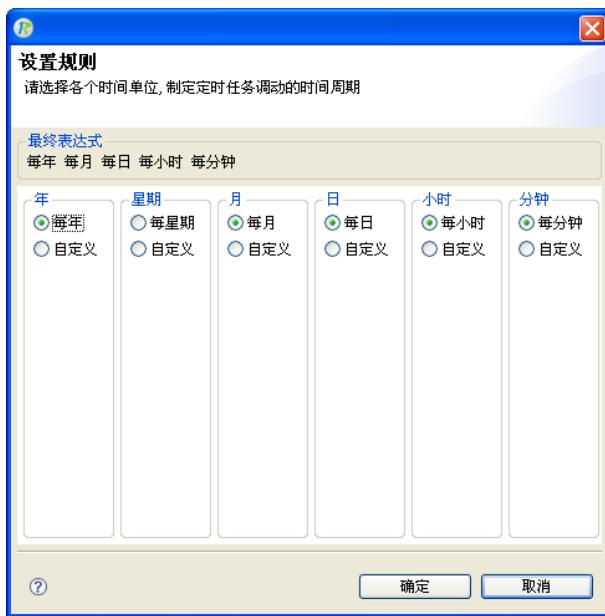
3.2.7.8 定时任务

要创建或修改定时任务，在当前项目工程下，双击消息流包目录，进入“交
换流程包管理”界面，再点击底部的“定时任务”，切换到定时任务配置列表。点
击“新建”按钮添加新的定时任务，单击默认名称可更改任务名称、任务标识和
任务实现类名。任务名称是该定时任务的描述，任务标识是查找该定时任务的
唯一标识，任务实现类全名是指继承抽象类
com.icss.ro.de.server.schedule.AbstractJob、并要实现 executeJob()方法的类的全
称。此方法通过 RCloud 云 EI 服务应用服务器调用并执行，要实现此调用需把
此实现类及其相关引入类打一个 jar 文件，放到 RCloud 云 EI 服务服务器的应用
lib 目录下。点击“删除”按钮可以删除定时任务。



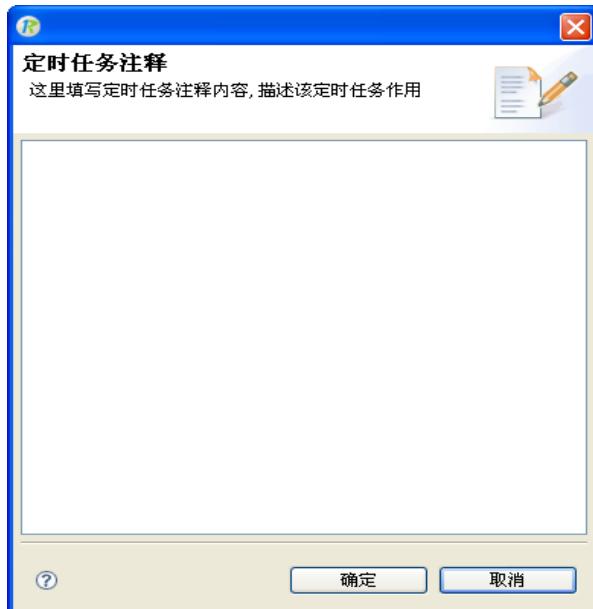
定时任务

点击执行规则列的“编辑规则”可编辑定时规则。定时规则按照年、月、星期、日、小时、分设置规则表达式，其中星期和日为二选一，其它都是必选项。



规则设置

点击描述列的“编辑描述”填写任务说明，这是个可选项，可以对当前定时任务进行有效的描述。



定时任务注释

最终编辑完成后，此时对定时任务的操作是在内存中存储的，可以点击保存按钮把此信息保存到 `dex` 文件中。最终设计完成消息流后可以保存到数据库中，在管理控制台就可以管理定时任务。

3.2.8 WSDL 定义

该目录下保存工程下可用的 WSDL 资源，包括通过工具设计的、从 RCloud 统一资源库引入的或从本地导入的，也可以将该目录下的 WSDL 资源发布到 RCloud 统一资源库中。详细操作请参考《RCloud Studio 4.0 产品用户使用手册》。

3.3 RCloud 云 EI 服务管理控制台使用指南

在 RCloud Studio 的 RCloud 云 EI 服务设计工具中完成服务流的设计后，设计数据被保存在 RCloud 云 EI 服务服务器中。此时可以通过管理控制台图形化地创建服务总线，并注册、配置及管理业务应用发布各类资源，构建整个以总线为基础的面向服务的业务交换场景。

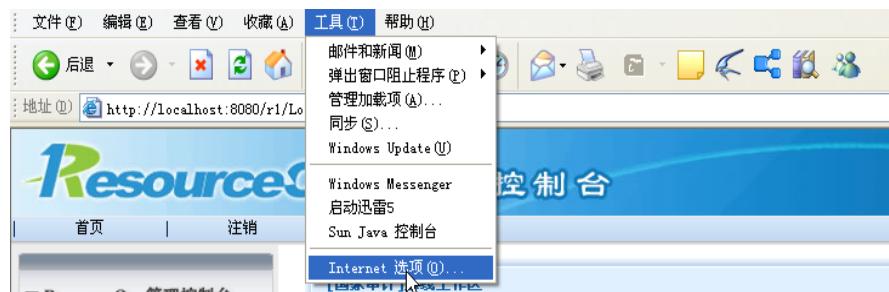
RCloud 云 EI 服务除了可以作为 ESB 产品提供总线服务外，也可以兼容传统 EAI 应用集成方式。如果现有应用系统本身不是以面向服务架构设计，则可

以使用 RCloud 云 EI 服务的基于消息流的处理机制完成特定两个应用之间的信息交换，同时还提供适配器机制兼容现有中间件和遗留系统。

3.3.1 访问控制

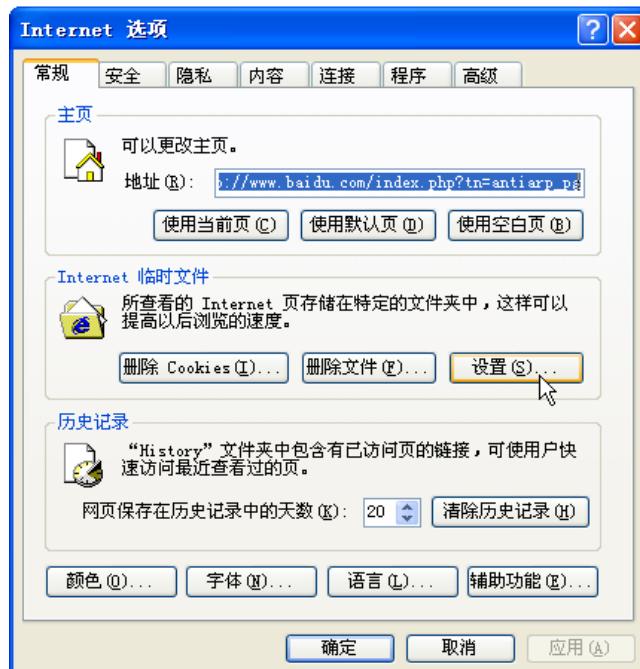
需要特别注意：在运行管理控制台时，推荐设置浏览器的一些选项，以便正常使用管理控制台，建议将 RCloud 云 EI 服务管理控制台加入信任域。设置步骤如下：

1、点击浏览器菜单上的“工具”菜单；如下图：



浏览器工具菜单位置界面

2、进入“Internet 选项”后在“常规”页面中点击“设置”按钮，如下图：



Internet 选项界面

3、进入到设置页面后，选中“每次访问此页时检查”选项。如下图



网页设置界面

3.3.2 首页概述



RCloud 云 EI 服务管理控制台首页界面

1、首页结构说明

当进入 RCloud 管理控制台的首页面时，会看到上图的界面。

1) 根据登陆用户所在组的权限不同，会有不同的显示方式，如，用 Administrator 组的用户（如 admin）登录后，会看到上面的界面，顶部区域主要提供首页、注销及系统帮助功能。

2) 左侧导航栏显示区域，主要提供 RCloud 云 EI 服务总线的新建和管理入口，所有已配置的总线都会在这里列出，用户可以很方便地管理每一

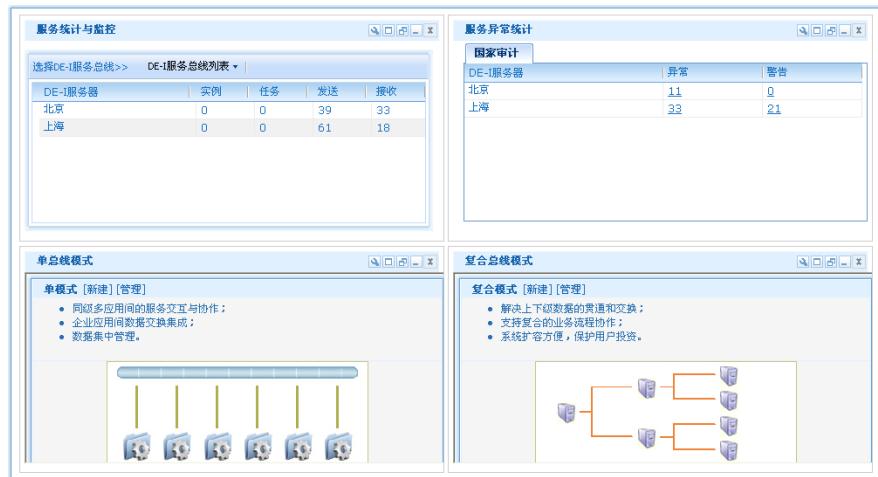
个 RCloud 云 EI 服务总线。

3) 工作区显示区域 (右侧区域), 是一个欢迎界面, 有对 RCloud 产品的文字介绍。

2、首页工作区显示内容配置

RCloud 云 EI 服务总线提供了用户自定义内容显示功能, 用户可以自定义工作区显示的内容。

点击左侧导航栏的“RCloudDE-1 管理控制台”, 右侧显示工作区主界面(如下图):



RCloudDE-1 管理控制台

用户想自定义工作区的显示内容, 只要点击工作区中如上图鼠标所指图标 , 就会进入工作区管理界面, 列表中列出的就是当前所有的工作区可显示内容, 用户可以根据需要显示或隐藏某一内容, 变更显示区域大小。

首页配置管理						
配置首页显示的Portal窗体。						
显示顺序	名称	窗体加载类型	是否显示	宽度	高度	操作
0	服务统计与监控	页面加载	显示	50%	220	
1	服务异常统计	页面加载	显示	50%	220	
2	单点平级模式	页面加载	显示	50%	220	
3	多级互连模式	页面加载	显示	50%	220	

工作区管理界面

如需增加一个新的显示模块, 点击列表上方的“添加”按钮, 填入相应信息。

显示顺序	名称	窗体加载类型	是否显示	宽度	高度
0	服务统计与监控	页面加载	显示	50%	220
1	服务异常统计	页面加载	显示	50%	220
2	单点登录模式	页面加载	显示	50%	220
3	多点互联模式	页面加载	显示	50%	220

添加界面

填入内容说明：

- 1) 名称：用于窗体标题显示。
- 2) 是否显示：选择“显示”，则该模块显示在工作区内；如果选择“隐藏”，则不显示。
- 3) URL：新增加模块的主页面 URL。
- 4) 显示顺序：必须是从 0 开始的整数。工作区显示时会根据数字的大小顺序从小到大排列。
- 5) 宽度：窗体占总工作区宽度的百分比。
- 6) 高度：窗体显示高度。
- 7) 类名：用户实现类，该类必须实现：
`com.icss.ro.de.adminconsole.frame.portal.IHomeModule` 接口。
- 8) 窗体加载类型：如果选择页面加载，则会根据用户填写的 URL 内容加载页面；如果选择的是类加载，则会根据用户填写的类的全路径来加载页面。
- 9) 描述：关于该模块的简单说明。

3.3.3 新建 RCloud 云 EI 服务总线

RCloud 云 EI 服务总线提供两种模式的配置，即单总线和复合总线。用户可以根据实际使用场景选择适合的服务总线模式进行配置。

单总线使用场景：

- 1、同级多应用间的服务交互与协作。
- 2、企业应用间数据交换集成。
- 3、数据集中管理。

复合总线使用场景：

- 1、解决上下级数据的贯通和交换。
- 2、支持多级的业务流程协作。
- 3、系统扩容方便，保护用户投资。

3.3.3.1 新建单总线

单总线的使用场景为，只涉及到一个 RCloud 云 EI 服务服务器，各集成对象通过该 RCloud 云 EI 服务服务器交互。

新建单总线，有以下几个步骤：

一、点击左侧导航栏的“RCloudEI 管理控制台”链接，在工作区中找到“单总线交换模式”区域，点击“新建”(如下图)，进入新建总线信息填写页面：



单总线交换模式

二、正确填写完整新建总线的各种必须信息，点击 按钮创建总线。

需要必须填写的信息包括：总线名称、RCloud 云 EI 服务 服务器名称、路由名称、代理服务。

三、因为是单总线，创建的总线也是 RCloud 云 EI 服务服务器。所以直接进入 RCloud 云 EI 服务服务器配置页面。

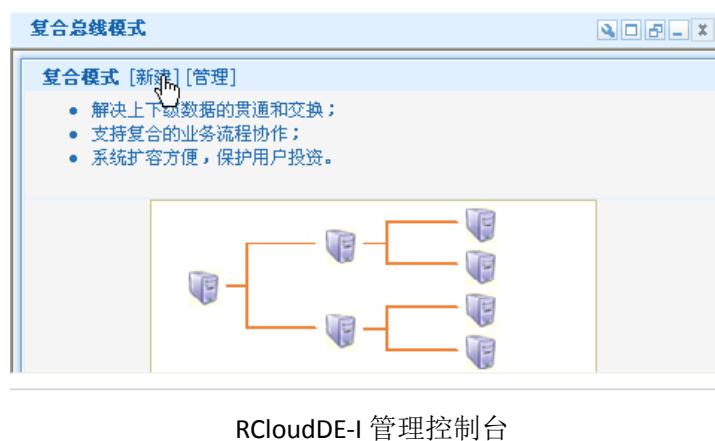
3.3.3.2 新建复合总线

复合总线主要使用在需要多个 RCloud 云 EI 服务服务器交互的场合。

新建复合总线，有以下几个步骤：

3.3.3.2.1 新建复合总线

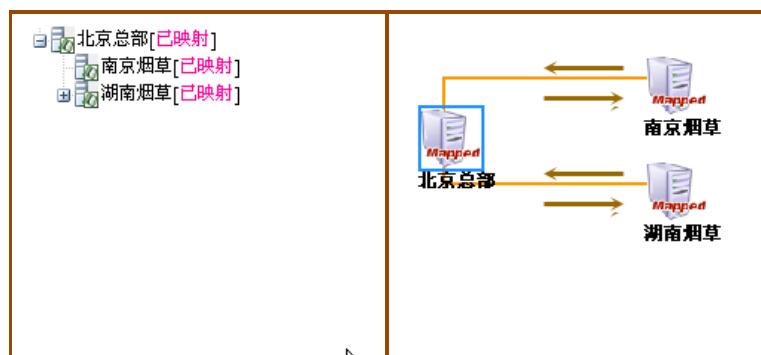
1、点击左侧导航栏的“RCloudEI 管理控制台”链接，在出现的工作区里找到“复合总线交换模式”区域，点击“新建”(如下图)，进入新建总线信息填写页面。



RCloudDE-I 管理控制台

2、正确填写总线名称(必须唯一)，点击 按钮创建总线。

总线工作区提供两种显示消息：树型视图和图形化视图。用户可以通过上方的切换按钮 进行切换。

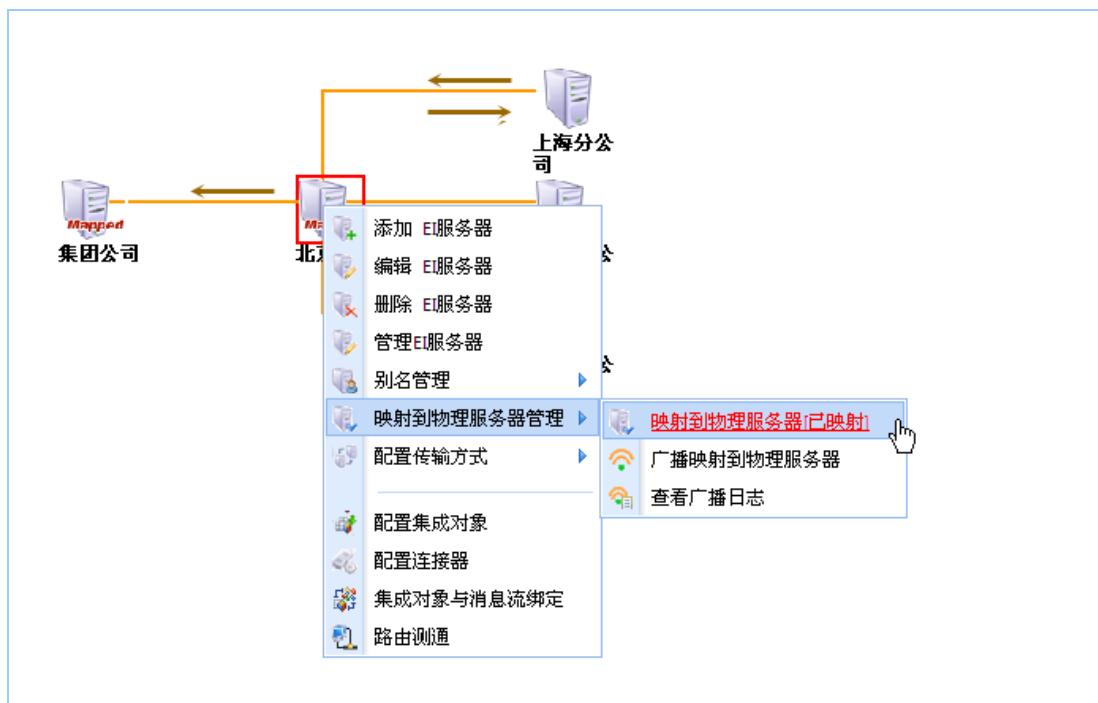


(树型视图)

(图形化视图)

3.3.3.2.2 根据实际需要，添加多个 RCloud 云 EI 服务服务器。

对于复合总线模式，在添加 RCloud 云 EI 服务服务器时，需要选中某个 RCloud 云 EI 服务服务器，点击右键“添加 EI 服务器”进行添加。在所有的 RCloud 云 EI 服务服务器配置完成后，必须将 RCloud 云 EI 服务服务器路由配置信息映射到物理服务器，然后再重启动 RCloud 云 EI 服务服务器。选中未映射的服务器，点击右键菜单中的“映射到物理服务器”完成映射工作。



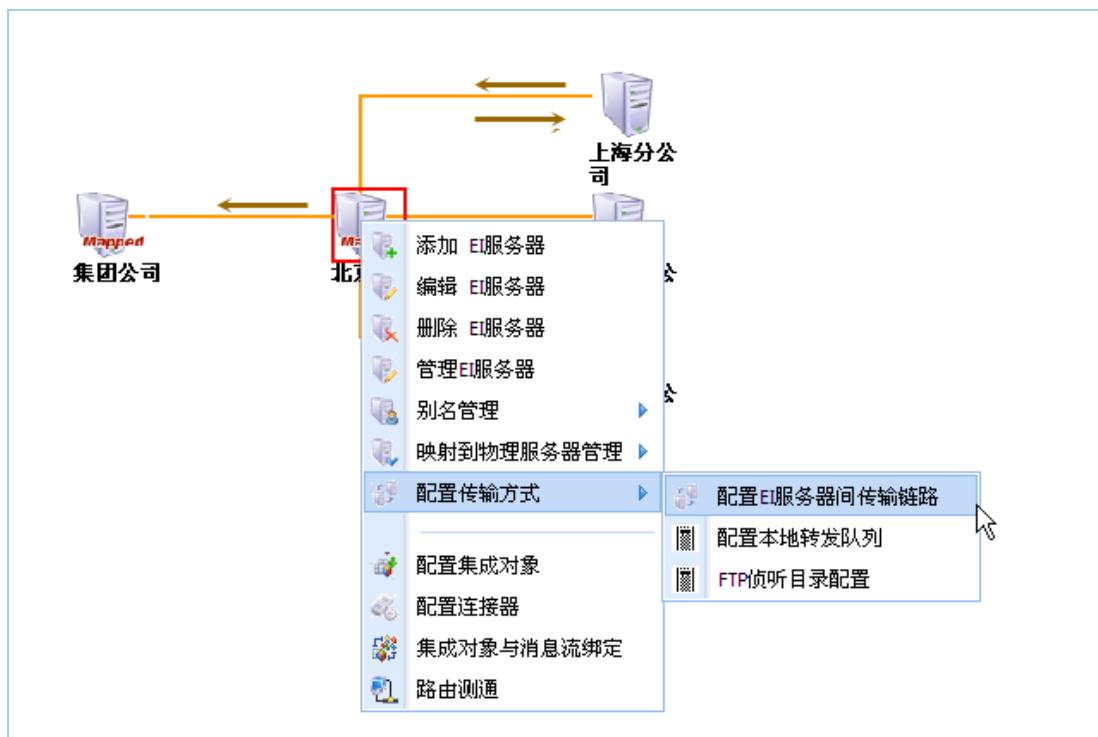
映射到物理服务器

3.3.3.2.3 配置 RCloud 云 EI 服务服务器之间的传输方式

选中需要配置服务器间传输方式的 RCloud 云 EI 服务服务器，点击鼠标右键菜单中的“配置传输方式”，在弹出的页面中选择合适的传输方式配置。

配置各个 RCloud 云 EI 服务服务器之间的传输方式操作步骤如下：

- 1、点击鼠标右键“配置传输方式”，如下图：



配置传输方式

2、配置传输方式：

The screenshot displays the 'Transport Path Configuration' screen. At the top, it says '配置当前 EI 服务器到与之相连的其他 EI 服务器之间的传输链路！注意：两个 EI 服务器间如果是http方式的传输链路，则只能配置一条！'. Below is a table titled '[集团公司] 到 [北京总部] 的传输链路'. The table has columns: 选择 (Select), 链路名称 (Link Name), 链路标识 (Link Identifier), 链路类型 (Link Type), and 操作 (Operations). One entry is shown: http_cop_bj, http_cop_bj, HTTP. At the bottom, there are navigation buttons (Back, Forward, Home) and a status bar indicating '第 1 - 1 条 共 1条'.

选择配置方式

如上图：首先从下拉列表中选择一个和当前服务器相连的其他服务器（如上图服务器北京总部），下面列出当前服务器和选择的其他服务器之间已经配置的链路信息。可以对已配置的链路信息进行编辑、删除。

如果当前还没有配置链路或没有合适的链路，可以点击“添加传输链路”按钮添加一个新的链路。

新建一个链路，首先选择链路的传输方式，目前提供四种传输链路类型，

分别如下：

1) “HTTP”

两个服务器之间只能配置一条 HTTP 类型的传输链路。在下拉菜单中选择“HTTP”，填写相关信息：

“链路名称”：给当前配置的链路指定一个显示名称

“链路标识”：给当前配置的链路指定一个标识，消息发送时可以通过指定改标识来决定消息传输链路，可由英文字母、数字或下划线组成。

URL：目标服务器访问地址（<http://host:port/context>）

点击“确定”按钮，即可。如下图：



HTTP 传输方式配置

2) “IBM-MQ”

在下拉菜单中选择“IBM-MQ”，并填写好相应信息

“链路名称”：给当前配置的链路指定一个显示名称

“链路标识”：给当前配置的链路指定一个标识，消息发送时可以通过指定改标识来决定消息传输链路，可由英文字母、数字或下划线组成。

“IP”：源服务器上的 MQ 队列管理器 IP 地址

“队列管理器名称”：源服务器上的 MQ 队列管理器名称

“端口”：源服务器上的 MQ 队列管理器访问端口。

点击“确定”即可，如下图所示：



IBM-MQ 传输方式配置

选择	链路名称	链路标识	链路类型	参数	操作
<input type="checkbox"/>	http	sdf	HTTP	http://192.9.107....	
<input type="checkbox"/>	mq	asd	IBM-MQ	192.9.107.60\$QM...	IBM-MQ通道绑定或注册
<input type="checkbox"/>	tlg	tlq	TongL...	\$qcu\$\$\$\$1	TongLINK/Q连接注册

(链路列表)

“IBM-MQ”链路类型可以绑定多个传输通道，每一个通道都会有一个绑定标识，不同的消息可以通过指定不同的传输链路标识和通道绑定标识来走不同的通道传输，提高传输的效率。

在链路列表中，每个 IBM-MQ 类型的链路后面都会有一个“IBM-MQ 通道绑定或注册”按钮（如上图），通过点击该按钮可以进入通道绑定配置管理页面，对通道进行绑定或管理。

选择	应用/消息绑定	远程队列名称
<input type="checkbox"/>	mq_16_18_57_c1	RQ_18_57_1
<input type="checkbox"/>	mq_16_18_57_c2	RQ_18_57_2

(通道配置管理)

如果需要绑定一个 IBM MQ 通道，首先点击“引入/注册 IBM-MQ 通道”，下面显示传输通道创建所需的信息：

应用/消息绑定：通道的绑定标识，消息发送时使用该标识指定消息所走的通道。

远程队列名称: 选择一个远程队列名称, 下拉列表中显示的是所有当前传输链路对应的 MQ 队列管理器上所配置的远程队列名称。如果列表中没有合适的远程队列, 可以通过点击后面的“队列配置”按钮, 进入 MQ 队列管理模块中创建一个新的远程队列。队列的创建请参考 3.3.4.10.1 队列管理介绍。

注: 链路标识和应用/消息绑定名称需要在整个总线中统一。

3) “TongLINK/Q7”

在下拉菜单中选择“TongLINK/Q7”, 并填写好相应信息

“**链路名称**”: 给当前配置的链路指定一个显示名称

“**链路标识**”: 给当前配置的链路指定一个标识, 消息发送时可以通过指定改标识来决定消息传输链路, 可由英文字母、数字或下划线组成。

“**队列管理单元名称**”: TongLINKQ 队列管理单元的名称

“**连接模式**”: 有两种, 一种是绑定 TongLINKQ 服务器模式, 指连接和当前 EI 服务器安装在一台主机上的 TongLINKQ 服务器, 这种模式不需要填写代理 IP 和代理端口; 另一种是客户端连接模式, 指通过 TongLINKQ 的客户端代理连接和当前 EI 服务器安装在不同主机上的 TongLINKQ 服务器。

“**代理 IP**”: TongLINKQ 的客户端代理的 IP 地址

“**代理端口**”: TongLINKQ 的客户端代理的端口。

点击“确定”即可, 如下图所示:



TongLINK/Q7 传输方式配置

[zll] 到 yt 的传输链路					
选择	链路名称	链路标识	链路类型	参数	操作
<input type="checkbox"/>	http	sdf	HTTP	http://192.9.107....	IBM-MQ通道绑定或注册
<input type="checkbox"/>	mq	asd	IBM-MQ	192.9.107.60\$QM...	IBM-MQ通道绑定或注册
<input type="checkbox"/>	tlq	tlq	TongLINK/Q	\$qcu\$\$\$\$\$1	TongLINK/Q连接注册

(链路列表)

“TongLINK/Q”链路类型可以绑定多个传输通道，每一个通道都会有一个绑定标识，不同的消息可以通过指定不同的传输链路标识和通道绑定标识来走不同的通道传输，提高传输的效率。

在链路列表中，每个 TongLINK/Q 类型的链路后面都会有一个“TongLINK/Q 连接注册”按钮（如上图），通过点击该按钮可以进入通道注册管理页面，对通道进行注册或管理。

TongLINK/Q连接注册
在“注册TongLINK/Q连接”中，填写队列名称

链路名称 [tlq] 链路参数 [\$qcu\$\$\$\$\$1]

选择	应用/消息绑定	注册TongLINK/Q连接	队列名称
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
		确定	重置
选择	应用/消息绑定	队列名称	

(通道配置管理)

下面显示传输通道创建所需的信息：

应用/消息绑定：通道的绑定标识，消息发送时使用该标识指定消息所走的通道。

队列名称： TongLINK/Q 队列的名称。

注：链路标识和应用/消息绑定名称需要在整个总线中统一。

4) “FTP”

此处 FTP 链路类型是指两台 EI 服务器之间通过 FTP 的方式实现消息的传输。在“选择链路类型”下拉框中选择“FTP”，填写如下信息：

链路名称： 给当前配置的链路指定一个显示名称

链路标识： 给当前配置的链路指定一个标识，消息发送时可以通过指定该

标识来决定消息传输链路，可由英文字母、数字或下划线组成。

FTP 服务器：可以选择“本地模式”或者“远程模式”，前者是指当前 EI 服务器与 FTP 服务器在同一台物理服务器上，后者反之。

(1) “本地模式”

此时不用填写 FTP 地址等信息，只需填写 FTP 目录在本地的绝对路径即可，如下图所示：

[贵州省] 到 贵阳市 的传输链路

添加传输链路

选择链路类型 FTP

链路名称	*	链路标识	*
FTP服务器	<input type="radio"/> 远程 注：FTP服务器与本EI服务器不在同一台物理服务器上 <input checked="" type="radio"/> 本地 注：FTP服务器与本EI服务器在同一台物理服务器上		
*本地目录		绝对路径，例如：D:/ftp_up	

确定 清除 取消



本地目录：FTP 目录在本地的绝对路径

(2) “远程模式”

此时需要填写 FTP 服务器的访问信息，如下图所示：

[贵州省] 到 贵阳市 的传输链路

添加传输链路

选择链路类型 FTP

链路名称	*	链路标识	*
FTP服务器	<input checked="" type="radio"/> 远程 注：FTP服务器与本EI服务器不在同一台物理服务器上 <input type="radio"/> 本地 注：FTP服务器与本EI服务器在同一台物理服务器上		
被动模式	<input type="checkbox"/>	是否支持以被动模式访问	
*IP地址		*端口	
*登录名		*登录密码	
*远程目录		相对目录，如： /ftp_up	

确定 清除 取消



IP 地址：FTP 服务器的访问 IP

端口：FTP 服务器的访问端口

登录名：FTP 的登录用户名

密码：FTP 的登录密码

远程目录：远程 FTP 访问目录

被动模式：FTP 是否以被动方式工作

填写好后点击“确定”按钮即可。

使用 FTP 传输链路时还需要在接收端的 EI 服务器上配置 FTP 倾听目录，参见章节 3.3.3.2.6 配置 FTP 倾听目录。

3.3.3.2.4 配置 RCloud 云 EI 服务服务器间的异常消息传输方式

异常消息传输方式主要是用来提供异常反馈消息传输时使用的。配置方式和正常的 RCloud 云 EI 服务 服务器间的传输方式一样。

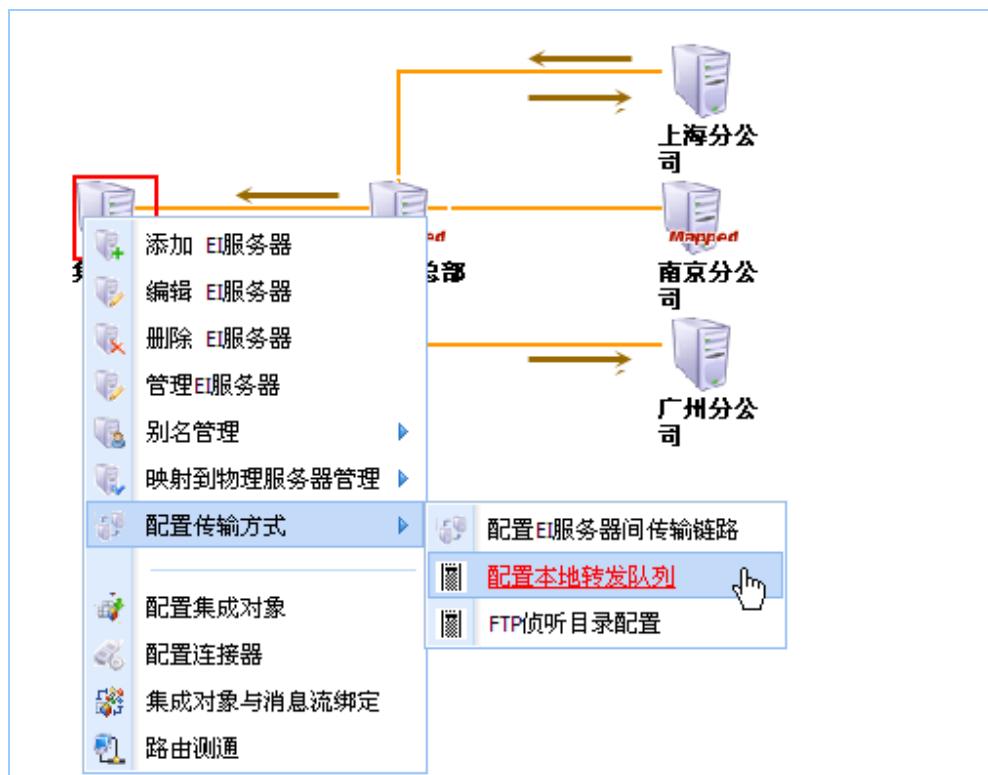
注意：

- (1) 如果是 IBM-MQ 和 TongLINK/Q 传输方式时在配置传输链接的时候的链路 code（即链路标识）需以"error_"开头，对通道标识的命名不限制以"error_"开头。
- (2) 如果用户没有配置专用的异常反馈消息传输链路，那么系统会从当前已经配置的传输链路中自动选择一条发送异常反馈消息。

3.3.3.2.5 配置本地转发队列

如果两个 RCloud 云 EI 服务服务器之间配置了 IBM MQ 或 TongLINK/Q 传输链路，那么在每个 RCloud 云 EI 服务服务器上都要为每种类型的传输链路配置至少一个本地转发队列，本地转发队列主要用于接收来自远程队列或 MQ 传输方式接入的消息，然后根据消息的目的地进行处理或者转发。

配置如下：



1、配置 MQ 本地接收转发队列



队列管理器名称：本地转发队列所在队列管理器名。

本地队列名称：本地转发队列名。

主机地址：队列管理器所在地址。

侦听端口：队列管理器侦听端口。

2、配置 TongLINKQ 本地转发队列



队列控制单元名称：TongLINKQ 的队列控制单元的名称。

接受收队列名称：本地转发队列名。

主机地址：TongLINKQ 的客户端代理的 IP 地址。

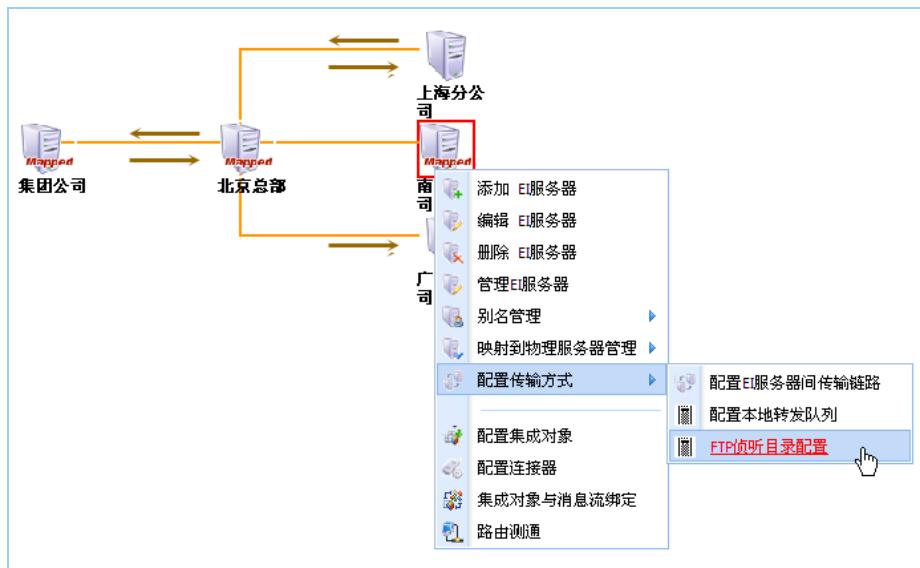
侦听端口：TongLINKQ 的客户端代理的端口。

3.3.3.2.6 配置 FTP 侦听目录

配置 FTP 侦听目录目的是让 EI 服务器侦听指定 FTP 的指定目录，以便对发送到 FTP 的消息进行处理。以下几种情况下需要配置 FTP 侦听目录：

- 1、EI 服务器之间传输链路为 FTP 时
- 2、DEAgent 连接器的传输方式为 FTP 时
- 3、配置 win32Agent 连接器时

配置 FTP 侦听目录步骤如下：



数据填写项如下：



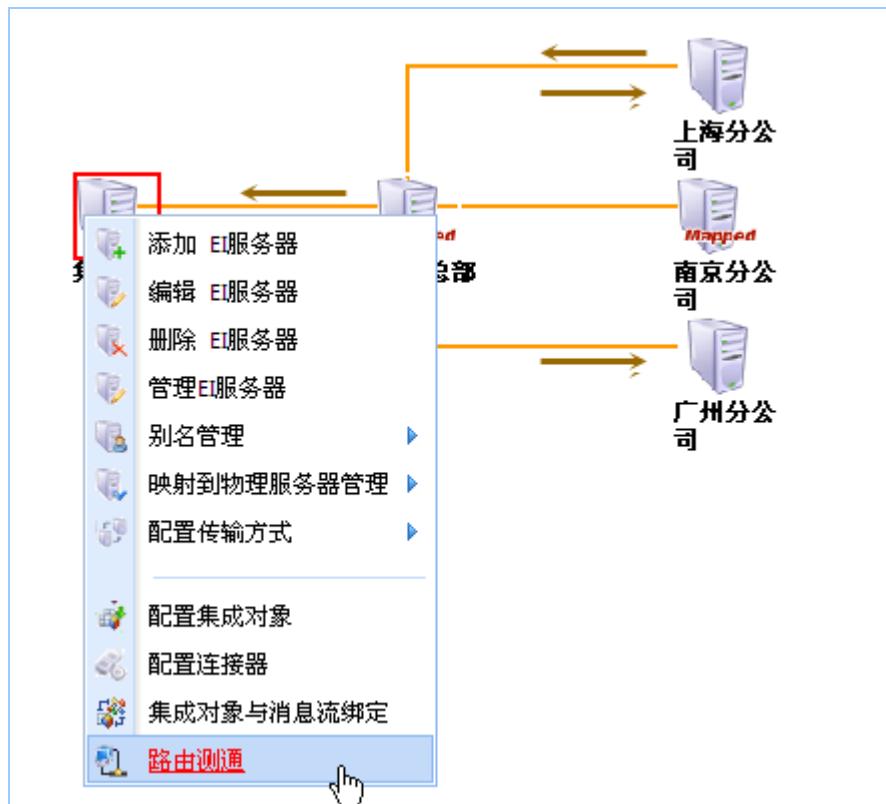
(远程听听目录模式：EI服务器和FTP服务器不在同一台主机上)



(本地目录侦听模式：EI 服务器和 FTP 服务器在同一台主机上。这里的远程目录需要填写绝对路径，并且目录中的斜线必须是“/”，如： D:/ftp_up)

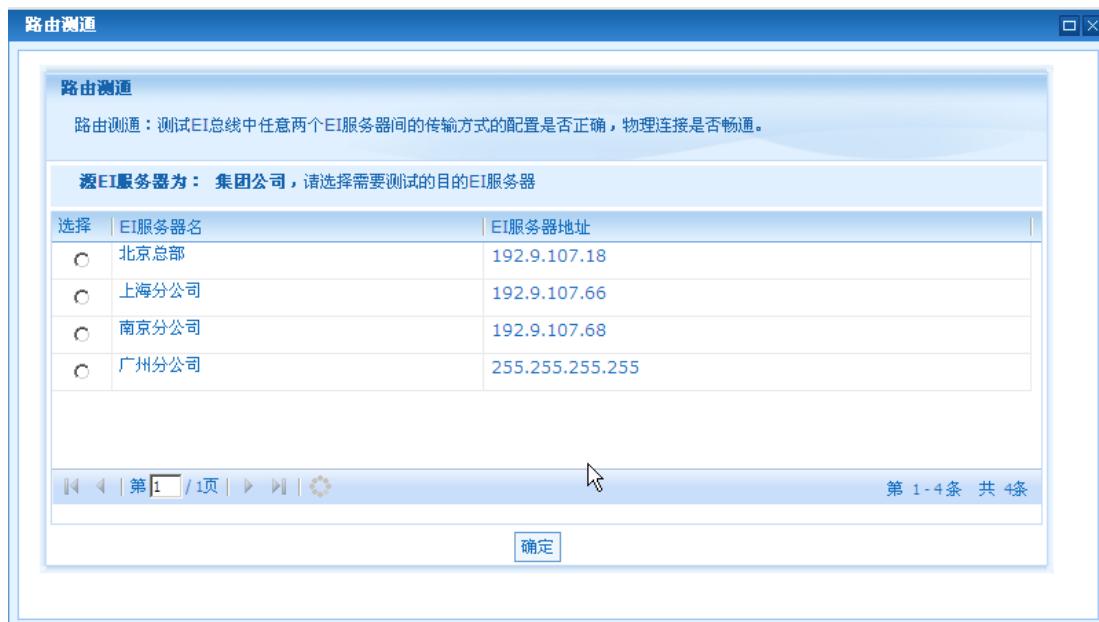
3.3.3.2.7 路由测通

复合总线中，两台 EI 服务器间的传输链路配置好以后，可以通过路由测通，测试 EI 服务器间的链路是否畅通，如下图，点击 EI 服务器的右键菜单中的“路由测通”。



(路由测通)

然后，选择需要测试的目的地 EI 服务器，如下图。



点击“确定”后，进入下图的页面，填写两个 EI 服务器间的传输链路（mq 和 TongLINKQ 需要填写传输链路名称和通道名称，中间用“.”号相隔）。



点击测试按钮，测通结果就会显示出来，如下图。



3.3.3.3 创建集成对象与连接器

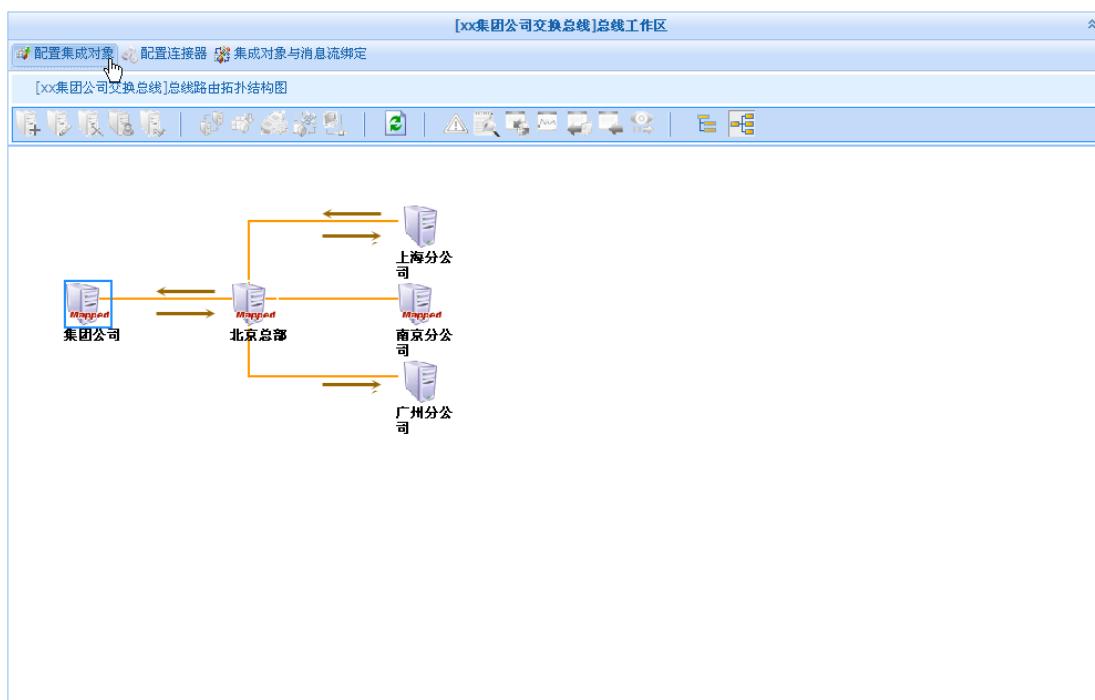
创建好总线和 RCloud 云 EI 服务服务器后，需要将集成对象添加到相应 RCloud 云 EI 服务服务器上。

3.3.3.3.1 配置集成对象

集成对象即指客户端应用程序，目前提供六种类型的集成对象(WEB 应用、IBM-MQ、TongLINKQ7.0、关系型数据库、Win32 集成对象、FTP 集成对象)接入。集成对象和 RCloud 云 EI 服务服务器之间通过 RCloud 云 EI 服务连接器进行数据的传输。在 RCloud 云 EI 服务服务器上配置集成对象的操作如下：

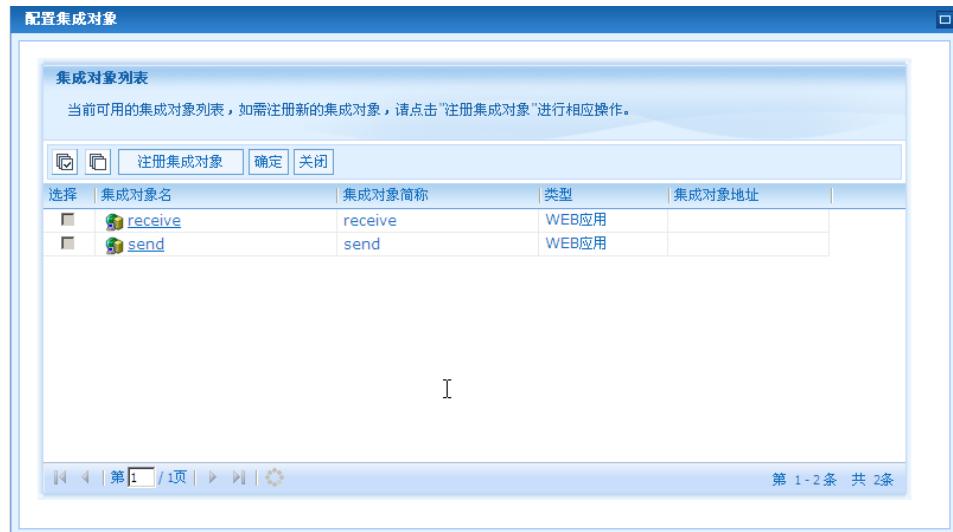
一、配置集成对象。

选中当前 RCloud 云 EI 服务服务器，点击上方“配置集成对象”按钮，如下图：



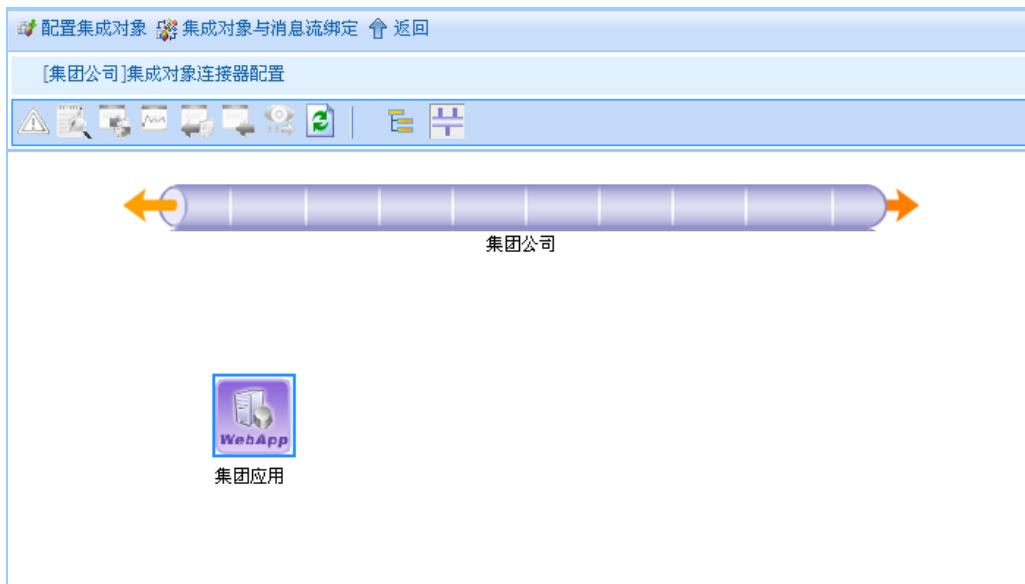
配置集成对象

在出现的“配置集成对象”窗口中添加需要和当前 RCloud 云 EI 服务服务器交互的集成对象，如下图：



配置集成对象界面

添加完成后再点“确定”按钮引入。如下图：



配置集成对象后

二、删除集成对象

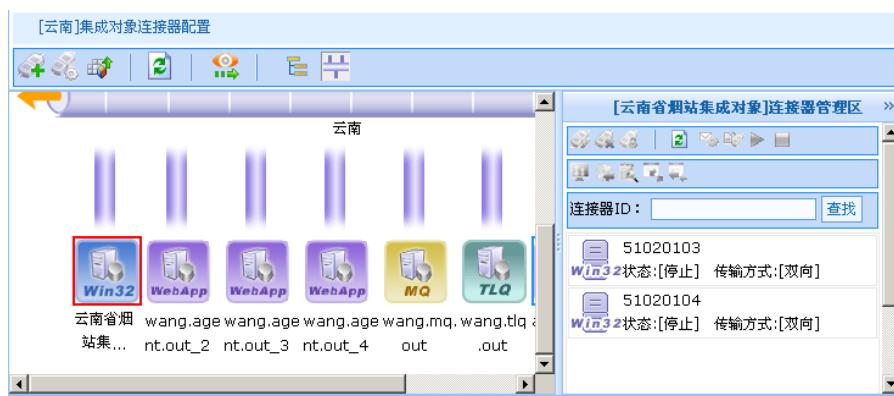
如果要删除集成对象，在管控中点击相应总线，在“总线工作区”中选择相应 server 后点击“配置连接器”，在配置连接器页面中选择要删除的集成对象，点击右键，选择“移除集成对象”。注意：删除集成对象将把配置在集成对象上的连接器也一并删除，请谨慎操作。



删除集成对象界面

3.3.3.3.2 为集成对象添加连接器

选中当前服务器下的某一集成对象，点击上方的添加连接器图标添加。另外，选中一个集成对象后的鼠标右键菜单中也提供了添加连接器入口。（连接器配置详细说明请参考小结 3.3.3.4 RCloud 云 EI 服务连接器配置）

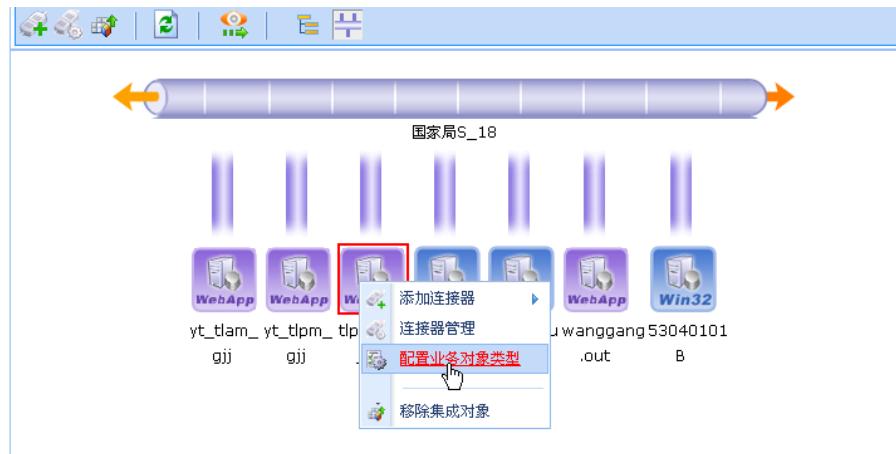


为集成对象添加连接器

3.3.3.3.3 配置业务对象类型

一个集成对象如果支持多种业务消息类型，可以将连接器绑定到多个不同中软国际有限公司

格式的消息流的输入或输出节点，每一种消息类型绑定一个消息流，即一个连接器可以启动多个消息流。右击连接器，点击菜单中的“配置业务对象类型”，如果服务器上没有对应的业务对象消息类型，则注册新的业务消息类型，填写业务消息类型的名称、ID 和说明，要确保在设计工具中设计相同的业务消息类型，并发布到服务器上；如果服务器上有对应的业务对象消息类型，则从服务器中导入业务消息类型。



配置业务对象类型

This screenshot shows a registration form for a business message type. The title bar says '业务消息类型'. The main area contains three input fields: '业务消息类型名称' (Name), '业务消息类型ID' (ID), and '说明' (Description). Below these fields are buttons for '确定' (Save), '重置' (Reset), and '取消' (Cancel). At the bottom, there are navigation buttons for '选择' (Select) and '业务消息类型名称' (Name), and a search bar with '说明' (Description) and '业务消息类型ID' (ID).

注册业务消息类型



导入业务消息类型

3.3.3.3.4 配置消息处理器

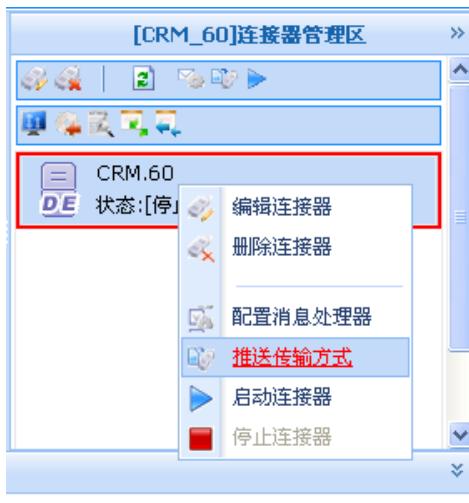
配置消息处理器是只有 DEAgent 连接器的配置才有的步骤，DEAgent 输出连接器以及有相应的 DEAgent 输入连接器需要绑定到消息处理器，因此需要为 DEAgent 连接器配置消息处理器。配置消息处理器的详细步骤请参照 3.3.4.3 小节的 DEAgent 连接器。

3.3.3.3.5 推送

如果是 DE Agent 连接器或者文件连接器，需要做推送的操作，将客户端信息和传输链路推送到客户端配置文件中，这样客户端才能正确访问 RCloud 云 EI 服务服务器。推送操作的入口有两个：一是“连接器管理区”的图标按钮，另一个是选中某一 DE Agent 连接器的鼠标右键菜单。



连接器管理区



右键菜单

注意：当添加连接器或修改连接器信息后，都必须重新执行一次推送操作。另外，要使推送的信息生效，应客户端应用需重新启动，并在管控中启动连接器。另外建议可以在连接器配置操作顺序中，按照最后推送并重启的方式。

3.3.3.3.6 集成对象与消息流绑定

在没有指定目的地的情况下，集成对象只有绑定到流程中的节点上才能正常运行。要执行绑定操作，首先必须先用 RCloud 设计工具根据实际业务设计出消息流，并发布到 RCloud 云 EI 服务服务器上。点击工作区上方的“集成对象与消息流绑定”按钮。进入绑定页面(如下图):

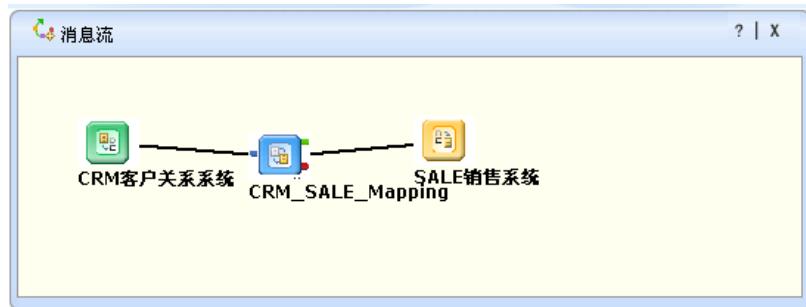


集成对象与消息流绑定

- 1) 左侧区域：显示当前 RCloud 云 EI 服务服务器上所保存的所有消息流程

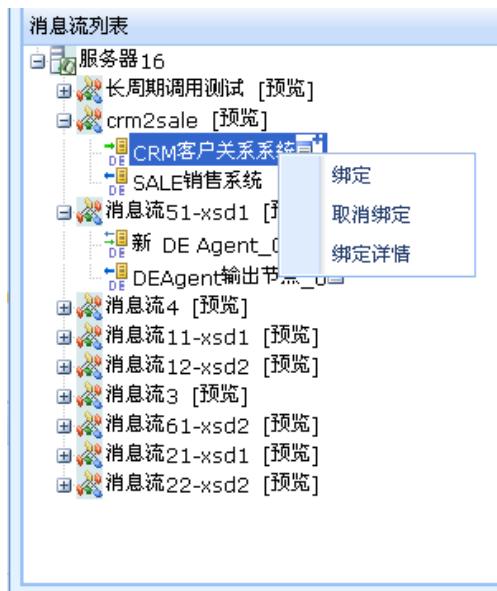
列表。

点击每个消息流名称后面的”[预览]”，可以查看消息流在 RCloudStudio 中的原设计图，如下图：



消息流图形化界面

右击消息流节点下面的输入/输出节点，将出现快捷菜单：



快捷菜单

2) 右侧区域：集成对象和业务对象列表

3) 消息流绑定与取消绑定

先选中左侧区域中将要绑定的消息流输入(输出)节点，再选中右侧区域中的集成对象(业务对象)，点击上方工具栏中的 绑定 按钮，或在选中的集成对象中的右键选择绑定菜单便可完成一次绑定操作，绑定后的节

点后面会出现 图标以标识该消息流节点或集成对象已经具有绑定关系。

取消绑定时，先选中左侧已经绑定的消息流节点，右侧区域则会自动定位到与之绑定的集成对象(业务对象)节点，点击工具栏中的  取消绑定 按钮，可以取消绑定。

4) 绑定规则

- 输入端绑定的业务消息类型要与消息流输入节点的消息流格式一致；
- 消息流节点的输入输出类型与连接器要一致；
- 一个消息流节点只能绑定一次，再次的绑定将替换之前的绑定关系，因些修改绑定时操作应慎重；
- 消息流的节点类型要与连接器类型匹配（如 MQ 输入节点只能绑定 MQ 输入连接器）；
- 一个集成对象如果支持多种业务消息类型，可以将（WS, HTTP, DEAgent）连接器绑定到多个不同格式的消息流的输入或输出节点，每一种消息类型绑定一个消息流，即一个连接器可以启动多个消息流；
- 在同一个连接器中，一个业务消息类型只能绑定一次；如果绑定业务消息类型时，关联了某个消息处理器或者连接器，则这个连接器或者消息处理器不能在不指定业务消息类型的情况下再次绑定，否则该连接器或消息处理器之前的所有关联绑定将被取消或替换；
- DEAgent 输出连接器需要绑定到消息处理器，如果同时作为输入又作输出则需要绑定业务消息类型；
- 有响应的 DEAgent 输入需要绑定到消息处理器；
- 消息流中的交换服务器节点用于跨服务器连接不同的消息流，需要绑定到输出的目标服务器节点；
- DEAgent 输入如果是来自另一个消息流的输出，则需要绑定到该消息流所在服务器。

3.3.3.3.7 重启客户端应用和连接器

如果是使用 DE Agent 连接器或文件连接器的应用，推送后还需要重启这些应用，并在管控中启动注册的连接器。

在管理控制台中启动连接器有两种途径：

- 1) 在连接器管理区中启动。



启动连接器

- 2) 左侧导航栏。



左侧导航

3.3.3.3.8 客户端异常通知配置

如果客户端希望在发送消息失败后能够获得到异常通知，则可以进行客户端异常通知的配置，配置方法见章节 3.3.4.9.5 异常通知。

3.3.3.4 RCloud 云 EI 服务连接器配置

目前提供 8 种不同的 DE 连接器，它们是 DE Agent 连接器、IBM-MQ 连接器、SOAP/WS 连接器、HTTP 连接器、FTP 连接器，文件连接器、TongLINKQ 连接器、Win32 连接器。对于 IBM-MQ 连接器，只能是 MQ 集成对象才能添加。一个集成对象支持配置多个类型的连接器，DE Agent 连接器和文件连接器在同一

一个 WEB 应用集成对象中只能添加一次，HTTP 连接器和 SOAP/WS 连接器不受此限制，对于 Win32 连接器，只能是 Win32 集成对象才能添加，一个 Win32 集成对象可以添加多个 Win32 连接器，对于 FTP 连接器，只能是 FTP 集成对象才能添加，一个 FTP 集成对象可以添加多个 FTP 连接器。

配置连接器主页面如下图所示：

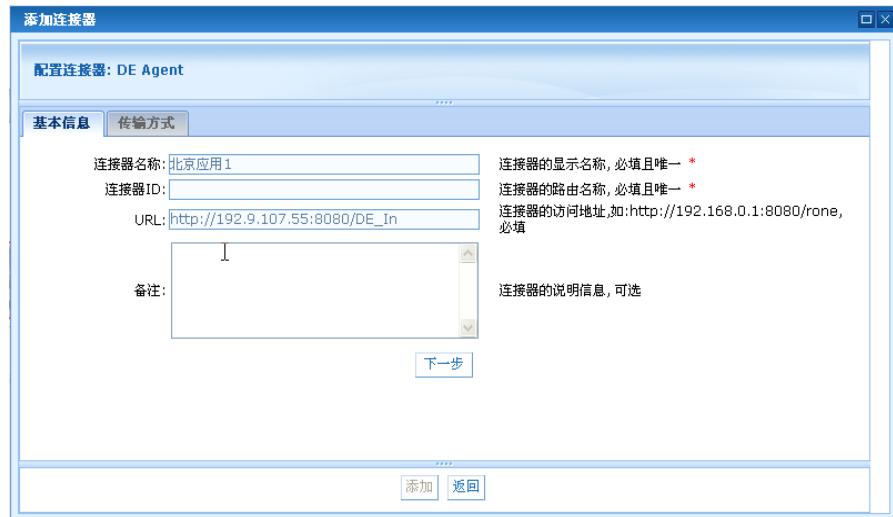


配置连接器主界面

3.3.3.4.1 DE Agent 连接器

DEAgent 连接器主要用来连接那些 DE 客户端应用，DEAgent 连接器配置按如下步骤操作：

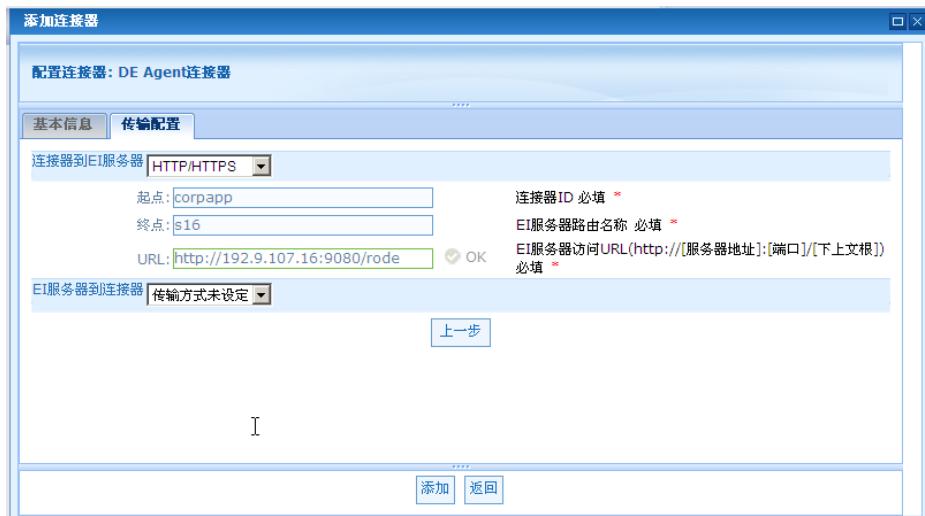
1、填写连接器基本信息，其中连接器 ID 必须唯一：



添加连接器

2、配置传输方式：

首先选择连接器所用的传输方式类型，Agent 连接器支持四种传输方式：HTTP/HTTPS、IBM-MQ、TongLINKQ7 和 FTP。



配置 HTTP/HTTPS 传输方式



配置 IBM-MQ 传输方式



配置 TongLINKQ7 传输方式



配置 FTP 传输方式

1) 在配置连接器到 RCloud 云 EI 服务服务器方向的 HTTP 链路时, 默认

的初始化数据只有 IP 或域名地址，还需要手动将服务器的 URL 补充完整（端口和上下文根）；

2) 如选择 MQ 或者 TongLINKQ7 类型传输链路时，则要求用于接收的队列必须为 RCloud 云 EI 服务服务器上注册的本地转发队列（参考 3.3.3.2.5 配置本地转发队列），否则该连接器应用的消息将无法正确发送给 RCloud 云 EI 服务服务器。

3) 如果连接器到 EI 服务器的传输方式为 FTP，则配置完连接器后，还需要在该目标 EI 服务器上配置 FTP 倾听目录，以便 EI 服务器能够倾听该 FTP 目录，获得消息，参考 3.3.3.2.6 配置 FTP 倾听目录。

3、填写好连接器基本信息和传输方式后点击“添加”按钮，添加连接器信息。

● 关于推送

是指将当前当前配置的客户端信息推送到该客户端。其中客户端配置信息包括：基本信息（连接器名称、ID、URL 等），传输方式。

注意：在推送操作之前：确保连接器所在客户应用已经启动，RCloud 云 EI 服务服务器的配置已经完成，当前配置的连接器信息如 URL 的正确。

● 关于消息处理器

消息处理器是 DEAgent 连接器的组成部分，现在提供了两种消息处理器：普通消息处理器和错误消息处理器。

普通消息处理器用于处理 DEAgent 连接器接收到的消息。用户根据 RCloud 云 EI 服务 SDK 提供的接口实现特定功能的消息处理器，通过在管理控制台中客户端的注册，与指定的客户端关联在一起。

错误消息处理器用于处理异常反馈消息，用户根据 RCloud 云 EI 服务 SDK 提供的接口实现特定功能的消息处理器，通过在管理控制台中客户端的注册，与指定的客户端关联在一起。

双击一个配置有 DEAgent 连接器的集成对象，选中右侧区域中的 DEAgent 连接器，打开消息处理器配置页面(如下图)：



配置消息处理器

This dialog box is titled 'DE Agent 的消息处理器配置' (DE Agent Message Processor Configuration) and has a sub-instruction '为DEAgent配置消息处理器'. It contains several input fields: 'ID' (ResponseAtt), '名称' (QueryResultProcessor), '类名' (agent.processor.ReceiveAndResponseAtt), and a '备注' (Remarks) section. There are also checkboxes for 'OK' and '类型' (Type). At the bottom are '确定' (Confirm), '清除' (Clear), and '取消' (Cancel) buttons.

添加消息处理器

● 配置一个消息处理器必须的信息包括

ID : 唯一标识一个消息处理器，不可重复。

类型：可以选择“普通消息处理器”或“独立运行的适配器”，默认选择普通消息处理器。名称：消息处理器显示名。

类名：消息处理器类的全路径(即客户二次开发时的类路径)。

如果是错误消息处理器还需要选在这个错误消息处理器所适用的业务对象类型，如下图：

DE Agent的消息处理器配置

为DEAgent配置消息处理器

<input type="button" value=""/>	<input type="button" value=""/>	<input type="button" value="删除消息处理器"/>	<input type="button" value="添加消息处理器"/>	<input type="button" value="返回"/>
ID	<input type="text"/> *	类型	<input type="button" value="错误消息处理器"/>	*
名称	<input type="text"/> *			
类名	<input type="text"/> *			
适用的业务对象类型	<input checked="" type="checkbox"/> name2			
备注	<input type="text"/>			
<input type="button" value="确定"/> <input type="button" value="清除"/> <input type="button" value="取消"/>				

3.3.3.4.2 IBM-MQ 连接器

IBM-MQ 连接器只能用于 IBM-MQ 集成对象的连接。

注意：在 RCloud 云 EI 服务设计工具中创建 MQ 节点后，要在对应的 MQ 服务器上创建名为 RONEDE.SVRCONN 的服务器连接通道。 RONEDE.SVRCONN 这个名字可以在 RODEConfig.xml 中进行修改。这是使用 IBM-MQ 连接器的前提条件。

连接器配置步骤如下：

1、填写连接器的基本信息，其中连接器 ID 必须唯一，如下图：

编辑连接器

配置连接器： IBM-MQ连接器

<input type="button" value="基本信息"/>	<input type="button" value="MQ队列配置"/>	<input type="button" value="传输配置"/>
连接器名称： <input type="text" value="mq_zl"/> *	连接器的显示名称，必填且唯一 *	
连接器ID： <input type="text" value="th"/> *	连接器ID，必填且唯一 *	
<input type="text" value="fgh"/>	连接器的说明信息，可选	
<input type="button" value="下一步"/>		
<input type="button" value="保存"/> <input type="button" value="返回"/>		

基本信息界面

2、配置侦听，如下图：



配置侦听

3、如果选择的输出类型是连接器到 RCloud 云 EI 服务服务器，可以配置传输方式，指定目的地，如果不配置，需要绑定流程，如下图：



配置传输方式

消息目的地：可以直接指定消息发送的目的地（目的地服务器名、客户端路由名）。如果指定了目的地，不需要绑定流程。

服务器间传输链路：指定通过 MQ 输入的消息使用的服务器间的传输链路。

填写格式为：链路 code.通道 code 例如：“LINK1.CHANNEL1”

如果不填，系统自动选择链路进行传输；

如果填写，系统根据用户指定的链路传输；

如果服务器间链路用户选择的是 HTTP 方式，只需指定链路 code；

如果选择的是 MQ 传输方式，需同时指定链路 code 和通道 code；如果只

指定链路 code 而没有指定通道 code, 系统会自动选择一条通道传输。

消息处理器 ID: 如果输出客户端是 DE Agent, 并且指定了消息目的地, 需要指定注册的输出端消息处理器 ID。

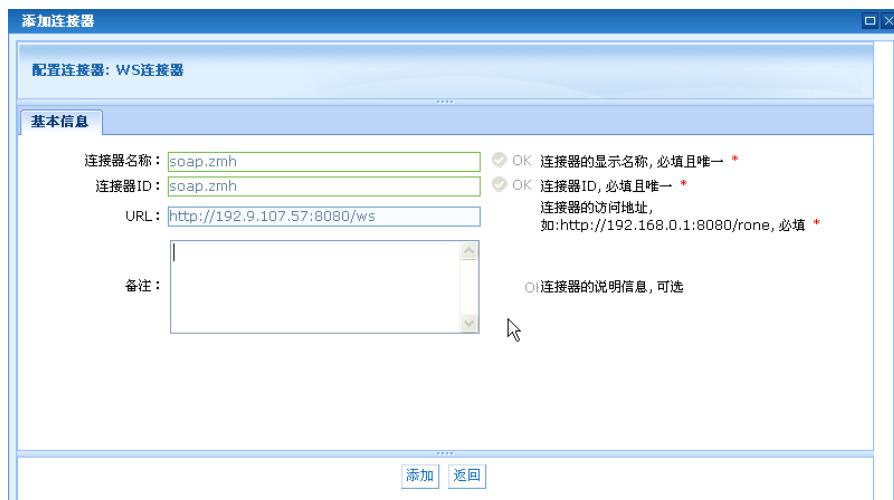
4、填写完以上信息，点击“添加”按钮添加连接器。

3.3.3.4.3 SOAP/WS 连接器

通过 Web service 方式传输消息。

连接器配置步骤如下：

1、填写连接器基本信息，其中连接器 ID 必须唯一，如下图所示：



SOAP/WS 连接器基本信息

连接器名称: 用于在管控中显示

连接器 ID: 连接器路由名称

URL: 连接器所在客户端访问 URL (http://host:port/context)

2、填写好以上信息后，点击“添加”按钮，添加连接器。

通过 SOAP/WS 连接器接入的应用开发详细参考《RCloud4.5Exchange-Integration 产品用户开发手册》中相关部分的说明。

3.3.3.4.4 HTTP 连接器

HTTP 连接器，连接器配置步骤如下：

1、填写连接器基本信息，其中连接器 ID 必须唯一，如下图：



HTTP/HTTPS 连接器基本信息

连接器名称：用于在管控中显示

连接器 ID：连接器路由名称

URL：连接器所在客户端访问 URL (http://host:port/context)

2、填写完以上信息后点击“添加”按钮，添加连接器。

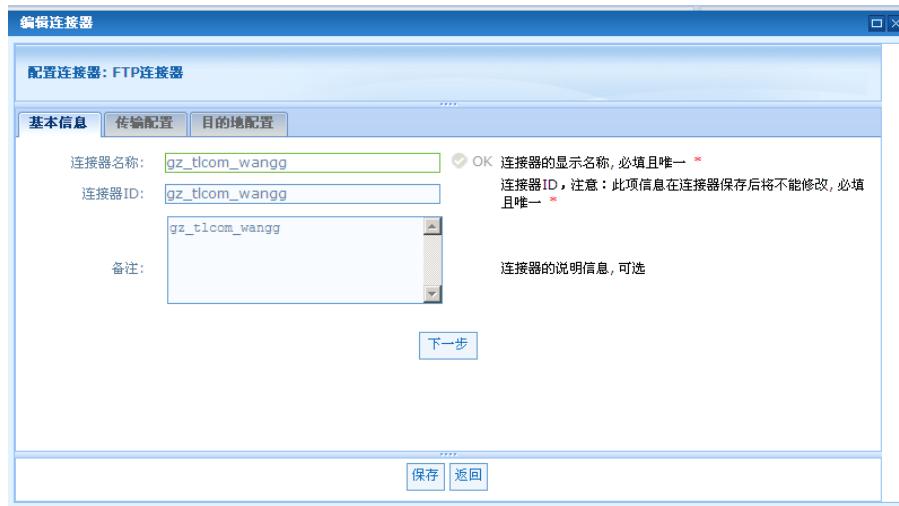
通过 HTTP/HTTPS 连接器接入的应用开发详细参考

《RCloud4.5Exchange-Integration 产品用户开发手册》中相关部分的说明。

3.3.3.4.5 FTP 连接器

连接 FTP 服务器，进行大文件传输，连接器配置步骤如下：

1、填写连接器基本信息，其中连接器 ID 必须唯一，如下图：



连接器基本信息

2、配置传输方式，如下图所示：



配置传输方式

3、配置目的地，如下图所示：



目的地配置

在传输配置中配置了“连接器到 EI 服务器”方向时，如果没有进行绑定流程，这里必须填写目的地服务器路由名称和目的地客户端路由名称，系统会根据这里指定的目的地转发消息，如果接收端连接器不需要消息处理器则无需指定消息处理器 ID。指定服务器间的传输链路和通道，这里填写的是链路 code 和通道 code(参考 3.3.3.4.2 IBM-MQ 连接器中传输链路配置)。

4、填写完以上信息，点击“保存”按钮，添加连接器。

说明：用户系统向 FTP 目录传输文件的过程中，对没有上传完成的文件要以*.tmp 为后缀，这样的文件不会被系统处理，当传输完毕后把*.tmp 文件改成原文件名。(参考产品用户开发手册 3.2.3.2 接入接触 FTP 连接器代码实例)

3.3.3.4.6 文件连接器

对文件数据抽取等操作，连接器配置步骤如下：

1、填写连接器的基本信息，其中连接器 ID 必须唯一，如下图所示：



连接器基本信息

2、配置侦听，如下图所示：



配置侦听

文件类型: 选择该连接器所处理的文件类型

侦听间隔: 指定系统多久侦听一次文件工作目录

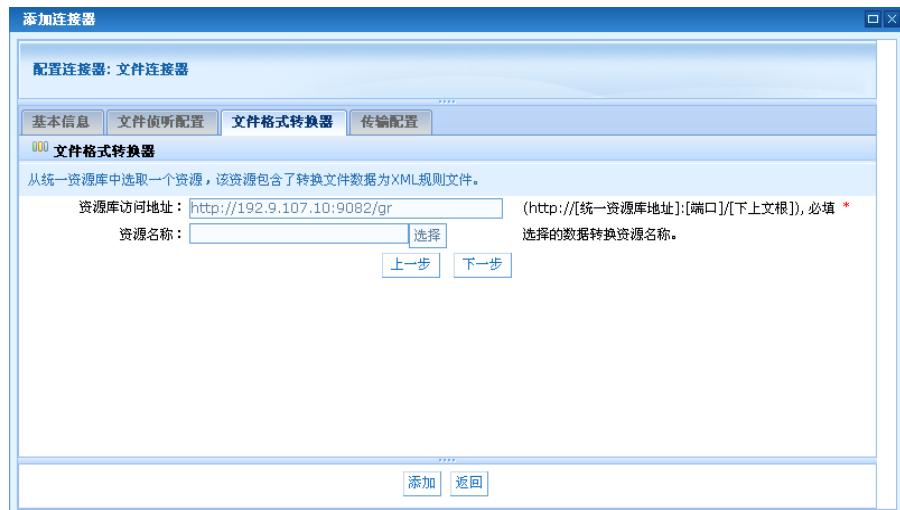
文件工作目录: 要处理的数据文件的放置路径

前缀: 数据文件的前缀，如果指定值，在文件工作目录中，只有文件名以该值开头的文件才会被处理。

包含文字: 如果指定值，在文件工作目录中，只有文件名中包含该值的文件才会被处理。

结尾文字: 如果指定值，在文件工作目录中，只有文件名中以该值结尾的文件才会被处理。(不包括文件后缀)

3、填写转换规则，其中资源名称从统一资源库中选择，如下图所示：



转换规则

填写 RCloud 资源库访问地址 (<http://host:port/context>)，点击“选择”按钮，从弹出的资源列表中选择文件转换规则。文件转换规则是通过 RCloud 云 EI 服务设计工具设计并发布到 RCloud 统一资源库的，详细参考 3.2.4.2。

4、配置传输方式，如下图所示：



配置传输方式

- 1) 配置连接器到服务器的链路信息
 - 2) 指定目的地服务器路由名称和目的地客户端路由名称。如果该连接器是绑定流程的，这里不需要填写；如果该连接器没有绑定流程，这里必须填写，系统会根据这里指定的目的地转发消息。
 - 3) 设置服务器间传输链路：指定服务器间的传输链路和通道，这里填写的是链路 code 和通道 code(参考 3.3.3.4.2 IBM-MQ 连接器中传输链路配置)。
 - 4) 指定输出端注册的消息处理器 ID
- 5、填写完以上信息后点击“添加”按钮，添加连接器。

3.3.3.4.7 TongLINKQ7 连接器

TongLINKQ7 连接器只能用于 TongLINKQ7 集成对象的连接。

连接器配置步骤如下：

- 1、填写连接器的基本信息，其中连接器 ID 必须唯一，如下图：



基本信息界面

2、配置侦听，如下图：



配置侦听

3、如果选择的输出类型是连接器到 RCloud 云 EI 服务服务器，可以配置传输方式，指定目的地，如果不配置，需要绑定流程，如下图：



配置传输方式

4、填写完以上信息，点击“添加”按钮添加连接器。

3.3.3.4.8 Win32 连接器

Win32 连接器只能只能用于 Win32 集成对象的连接

具体配置步骤如下：

1、填写连接器的基本信息，其中连接器 ID 必须唯一，如下图：



2、配置传输配置

2.1：连接器到 EI 服务器的传输配置（目前只支持 FTP 方式）



2.2: EI 服务器到连接器的传输配置（目前只支持 FTP 方式）



在配置 EI 服务器到连接器的传输方式时需注意：

- (1)、如果 EI 服务器和 FTP 服务器在同一台主机上，可以直接配置“下载文件目录”这项，EI 服务器会直接把数据方法这个目录中供连接器下载；
- (2)、如果在界面中填写了“下载文件目录”这项，EI 服务器将按照和 FTP 服务器在同一台主机上的方式处理，界面中填写的 IP 地址、端口等将失效。

如果配置了连接器到 EI 服务器间的传输方式，还需要这个连接器所在的 EI 服务器上配置 FTP 侦听目录。此目录是连接器把数据文件传输到 FTP 服务器上的目录。EI 服务器需要侦听此目录，来获取消息数据。

注意：现在不支持 FTP 服务器的映射文件目录。

配置步骤参见章节 3.3.3.2.6 配置 FTP 侦听目录。

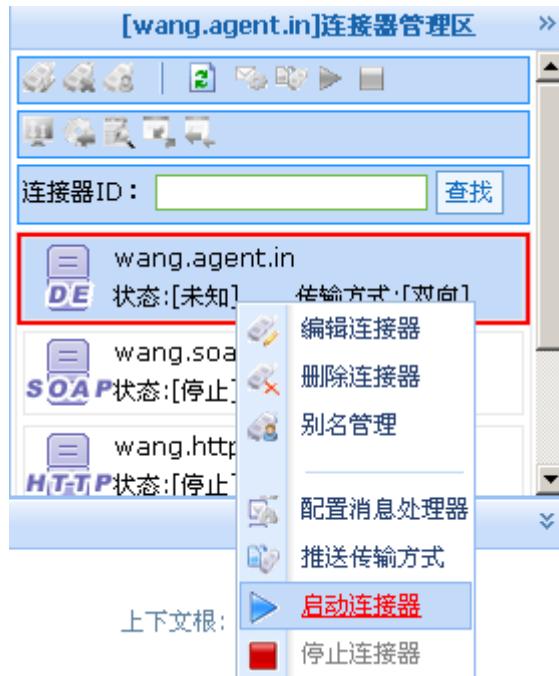
3、Win32 连接器不需要启动，推送。

3.3.3.5 连接器的启动停止

在添加并推送完连接器后，需要启动连接器，使连接器运行起来，启动连接器的方式如下图所示：



启动连接器方式（1）



启动连接器方式（2）

启动的连接器不能再编辑，删除，或移出集成对象，如果需要执行这些操作，需先停止连接器，如下图所示：



停止方式 (1)



停止连接器方式 (2)

注意：在配置完连接器并推送完后，需要重新启动 RCloud 云 EI 服务服务器和客户端应用。然后再启动连接器。

3.3.3.6 连接器搜索

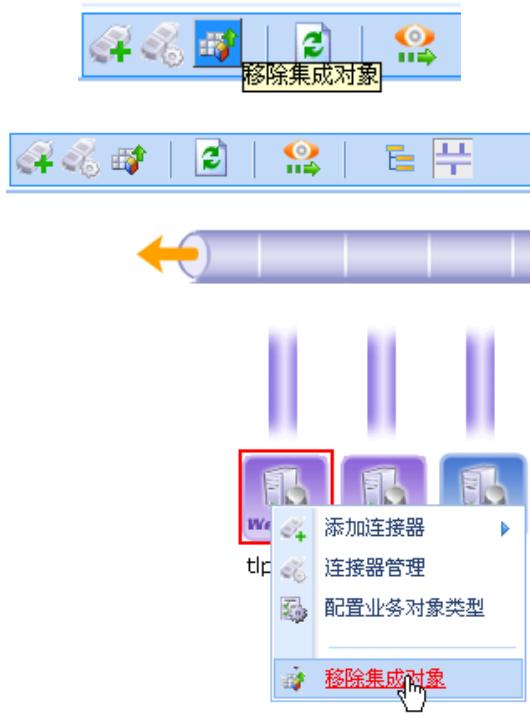
对于连接器数量多的情况下，为了查找方便，可以使用连接器的搜索功能，目前只支持精确搜索。



连接器搜索

3.3.3.7 移除集成对象

可以在管理界面中移除集成对象，点击鼠标右键“移除集成对象”菜单，来移除选中的集成对象。



移除集成对象

3.3.3.8 别名管理

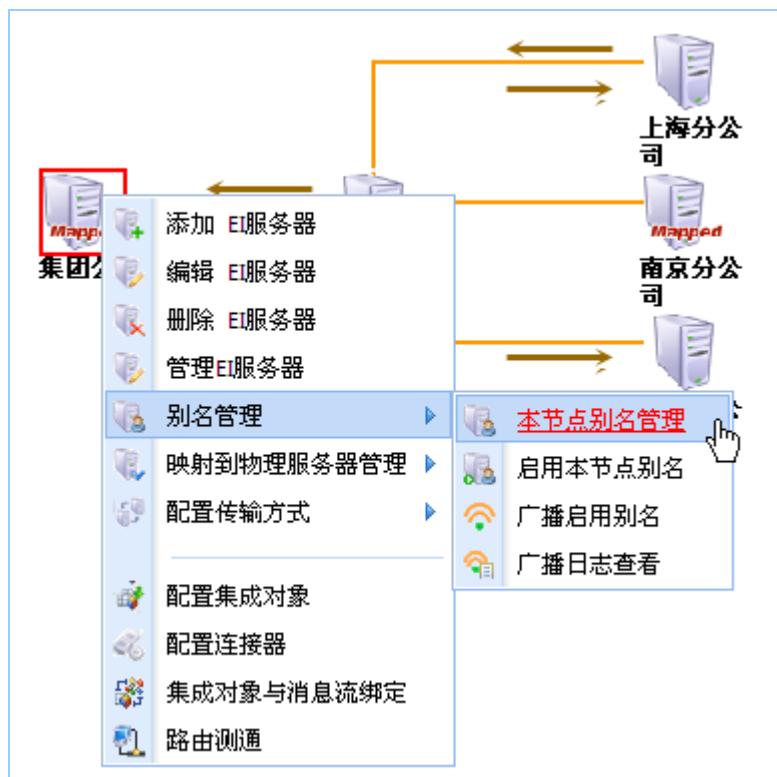
可以给服务器和连接器指定一个别名，通过指定别名也可以访问到服务器和客户端，别名的配置是在管控中完成的。用户在发送消息时，通过在消息头中设置目的地服务器路由名称的别名和目的地客户端 ID 的别名也能够通过

RCloud 云 EI 服务网络将消息传输到目的地（效果和直接指定目的地路由名称一样）。

3.3.3.8.1 服务器别名管理

3.3.3.8.1.1 本节点别名管理

- 1、进入总线工作区，选中需要添加别名的服务器，打开鼠标右键菜单，选择别名管理



- 2、点击“添加别名”按钮添加一个别名

The dialog box title is 'Name Management'. A note says: 'For route node to add alias, users can specify the destination through the specified route, such as: when sending messages via AGENT, specify the alias to indicate the destination.' The 'Add Alias' tab is active. Fields: 'Alias' (填写别名) with placeholder '例如：HQ_Beijing', 'Full Name' (填写全名) with placeholder '例如：北京总部', 'Description' (填写说明). Buttons: 确定 (Confirm), 清除 (Clear), 取消 (Cancel). Table below:

选择	别名	全名	说明
<input type="checkbox"/>	a181	18服务器别名1	
<input type="checkbox"/>	a182	18服务器别名2	

别名：唯一标识，一般表示具体的业务信息，需为英文字母。

全名：用于页面展示

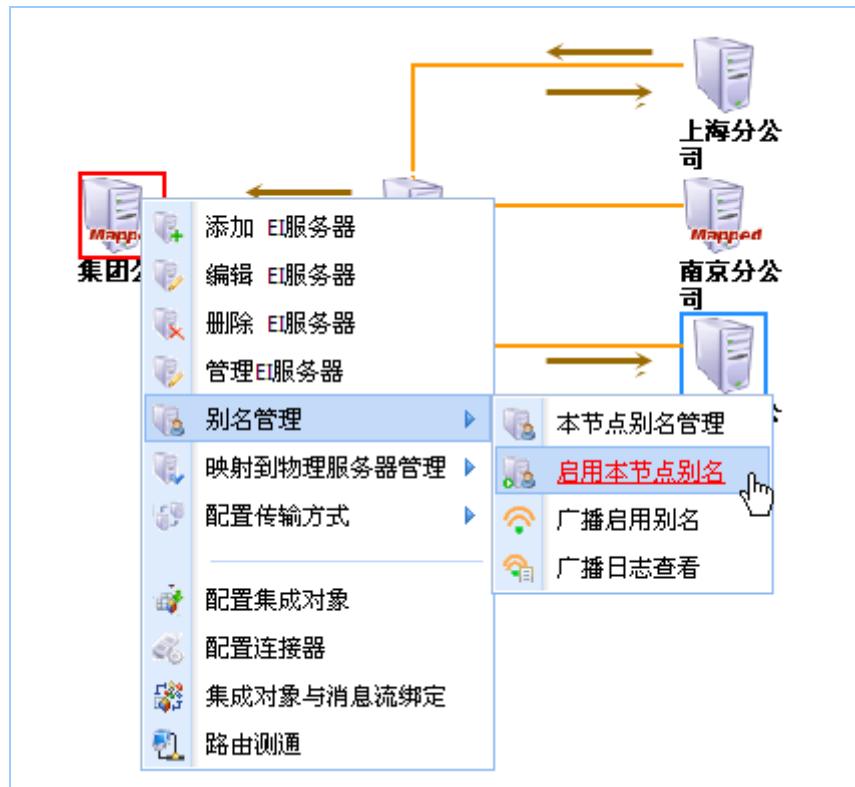
在别名管理界面，可以进行新建、修改、删除等操作。

3、修改、删除别名

点击别名名称，可以查看该别名的详细信息，也可以修改或删除该别名。

3.3.3.8.1.2 启用本节点别名

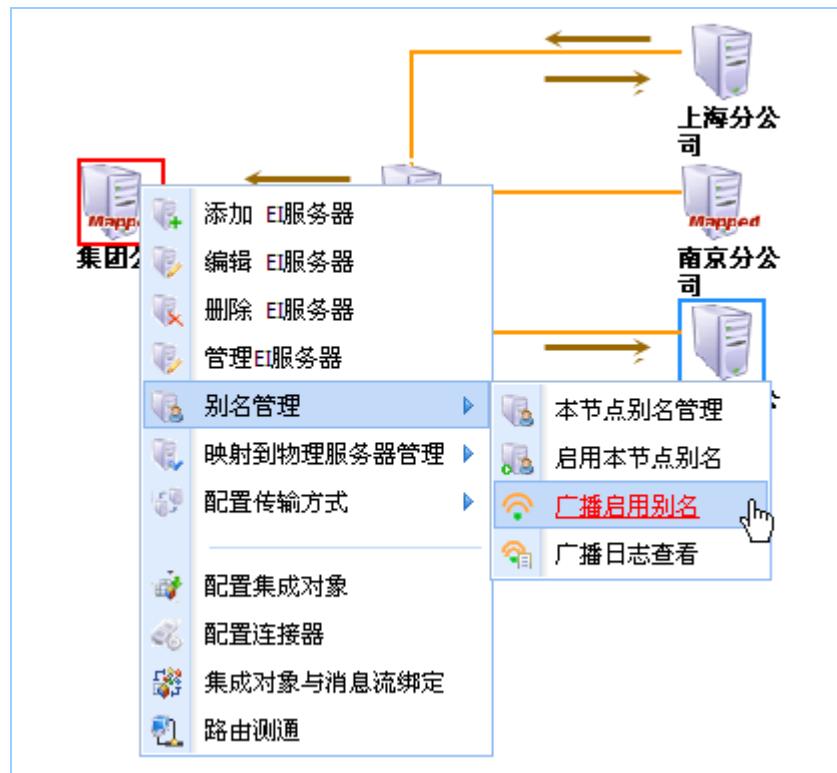
别名配置后，通过启动本节点别名，使其在该节点生效



启用本节点别名

3.3.3.8.1.3 广播启用别名

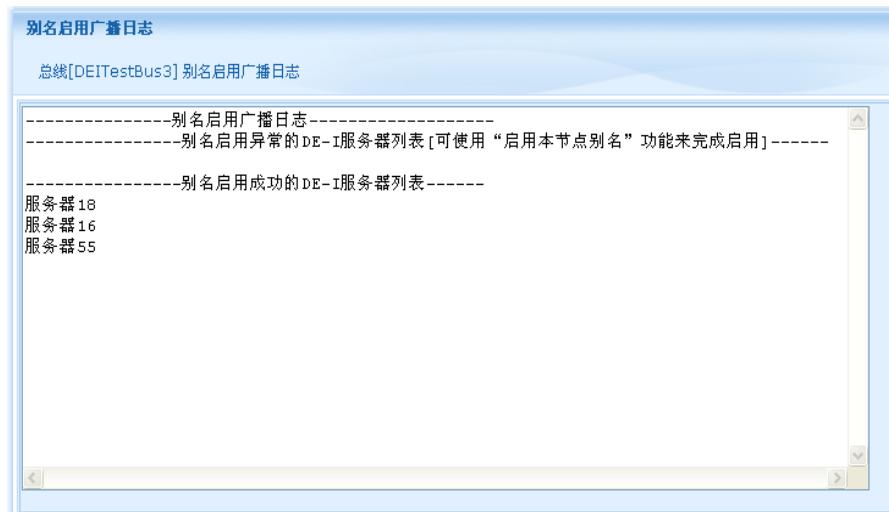
将当前总线中的所有 RCloud 云 EI 服务服务器的别名广播给当前总线下各个 RCloud 云 EI 服务服务器，使每个 RCloud 云 EI 服务服务器都能通过相应的别名找到对应的 RCloud 云 EI 服务服务器。



广播启用别名入口

3.3.3.8.1.4 广播日志查看

查看执行广播启用别名操作的执行情况，广播日志中会记录上次广播到每个 RCloud 云 EI 服务服务器操作成功或失败的情况。管理员通过该日志可以知道向哪些 RCloud 云 EI 服务服务器广播失败，进行补救。



广播日志查看

3.3.3.8.2 连接器别名管理

只有 DE Agent 连接器和 MQ 连接器支持别名，并且配置都相似。下面以 Agent 连接器为例说明别名配置过程。

选择需要添加别名的连接器，打开鼠标右键菜单，选择别名管理进入。



连接器别名管理入口



别名管理

别名编码: 别名唯一标识，一般表示具体的业务信息，需为英文字母。

别名名称: 用于页面展示

在别名管理界面，可以进行新建、修改、删除等操作。

对连接器的别名不需要重启 RCloud 云 EI 服务服务器和客户端应用。

3.3.4 RCloud 云 EI 服务总线管理

总线管理模块主要用于监控和管理注册到当前总线的服务器，包括服务器监控、异常消息管理、数据源配置、定时任务管理、交换信息统计、数据备份、消息流管理、参数配置管理、MQ 对象管理、消息跟踪等。

3.3.4.1 服务器监控

服务器监控模块主要提供管理注册到当前服务器下的连接器、当前服务器上交换引擎的监控和管理以及服务器运行日志查看等功能。

3.3.4.1.1 连接器监控

监控连接器状态并可以对连接器进行启停操作。监控当前 RCloud 云 EI 服务服务器下所有连接器的运行状态，并可以根据需要启动、停止连接器，或者禁止、启用连接器。如果连接器为禁止状态，则该连接器不管是否重启服务器都不可用。

连接器状态有以下几种：

- 1、运行：连接器可正常使用。
- 2、停止：连接器当前不可用，可通过启动来恢复
- 3、禁止：连接器不可用。
- 4、异常：发现当前连接器配置信息或状态信息出现异常情况（一般出现在 DE Agent 连接器服务器和客户端两边的状态不一致，可以重新启动连接器来修复）。
- 5、未知：由于网络或配置原因无法访问到连接器的状态。



总线结构

服务监控与统计

连接器管理监控

北京烟草 >> 连接器列表

序号	连接器名称	连接器状态	连接器类型	管理
1	f1.out.liaoxd	未知	应用客户端	[启动] [禁止]
2	Agent2.out.xiehf	运行	应用客户端	[停止] [禁止]
3	ftp.out.liaoxd	未知	FTP接入	[启动] [禁止]
4	MQ.mq.out.zmh	停止	IBM-MQ接入	[启动] [禁止]
5	MQ.de.out.zmh	停止	IBM-MQ接入	[启动] [禁止]
6	MQ.http.out.zmh	停止	IBM-MQ接入	[启动] [禁止]
7	MQ.soap.out.zmh	停止	IBM-MQ接入	[启动] [禁止]

连接器监控

3.3.4.1.2 交换引擎监控

查看当前服务器下交换引擎的状态，并可以进行交换引擎启停操作。监控当前 RCloud 云 EI 服务服务器上运行的消息流引擎的状态，可以根据需要启动或停止引擎。状态有以下几种：运行、停止、未知。

注意：运行中的 RCloud 云 EI 服务服务器，在停止引擎时请慎重操作，否则会导致异常情况发生。

服务引擎状态管理监控

查看当前服务器下服务引擎的状态，启动或停止服务引擎。

北京烟草 >> 连接器列表

序号	服务引擎名称	服务引擎状态	管理
1	消息流引擎	运行	[启动] [停止]

交换引擎监控

3.3.4.1.3 集成对象日志查看

查看当前服务器下的集成对象的日志。查看当前服务器下所有 DE Agent 客户端记录的 log 信息。并提供下载功能。同一时刻下，只能察看某一交换服务器下的所有客户端的日志信息。

要查看管理控制台页面，登录后单击左侧导航栏中的 某一交换服务器节点 > 服务器监控 > 集成对象日志查看 。将会出现所选中的交换服务器节点的客户端列表。再选择要查看信息的客户端，点击 **查看** 按钮，在右边的日志文件列表中将会列出该客户端下的所有已经备份出来的日志文件，再从这个客户端日志文件列表中选择查看日志文件。

The screenshot shows two main windows. The top window is titled '应用客户端日志列表' (Application Client Log List) and displays a table of clients with columns for 编号 (Number), 客户端名称 (Client Name), and 客户端地址 (Client Address). It includes a 'View' button for each row. The bottom window is titled '集成对象日志查看' (Integration Object Log View) and shows a detailed log for the client '江苏烟草专卖系统'. The log entries are timestamped and show various system messages related to pool file consistency checks and configuration initialization.

编号	客户端名称	客户端地址	操作
1	四川烟草消息读输出	192.9.107.55	查看
2	江苏烟草专卖系统	192.9.107.22	查看
3	四川ftp下载系统	192.9.107.55	查看

江苏烟草专卖系统>> Agent2.out.xield.rode.log>> 日志详细信息

```
2008-04-20 18:47:44,921 INFO - [ com.icss.ro.de.agent.admin.ClientManager ]ClientManager init...
2008-04-20 18:47:44,952 DEBUG - [ com.icss.ro.de.agent.config.RodeClientConfig ]RodeClientConfig init...
2008-04-20 18:47:44,952 DEBUG - [ com.icss.ro.de.agent.config.RodeClientConfig ]read client config:Agent2.out.xield
2008-04-20 18:47:52,622 DEBUG - [ com.icss.ro.de.agent.config.RodeClientConfig ]client config:
ctype=web
cnName=Agent2.out.xield
ServerRouteName=root
serverUrl=
transTypeStr=
statusStr=2
extendAttribute=null
2008-04-20 18:47:52,653 DEBUG - [ com.icss.ro.de.agent.config.RodeClientConfig ]RodeClientConfig init ok.
2008-04-20 18:47:53,122 DEBUG - [ messagePool_to_root ]checking pool file consistency...
2008-04-20 18:47:53,122 INFO - [ messagePool_to_root ]pool Consistency Checking...
2008-04-20 18:47:53,122 INFO - [ messagePool_to_root ]checking hdi file...
2008-04-20 18:47:53,122 INFO - [ messagePool_to_root ]checking hdi file ok.
2008-04-20 18:47:53,122 INFO - [ messagePool_to_root ]checking del file...
2008-04-20 18:47:53,122 INFO - [ messagePool_to_root ]checking del file ok.
2008-04-20 18:47:53,122 INFO - [ messagePool_to_root ]pool Consistency Checking complete.
2008-04-20 18:47:53,122 DEBUG - [ messagePool_to_root ]checking OK.
2008-04-20 18:47:53,122 DEBUG - [ messagePool_rec_root ]checking pool file consistency...
2008-04-20 18:47:53,122 INFO - [ messagePool_rec_root ]pool Consistency Checking...
2008-04-20 18:47:53,122 INFO - [ messagePool_rec_root ]checking hdi file...
2008-04-20 18:47:53,122 INFO - [ messagePool_rec_root ]checking hdi file ok.
2008-04-20 18:47:53,122 INFO - [ messagePool_rec_root ]checking del file...
2008-04-20 18:47:53,122 INFO - [ messagePool_rec_root ]checking del file ok.
2008-04-20 18:47:53,122 INFO - [ messagePool_rec_root ]pool Consistency Checking complete.
2008-04-20 18:47:53,122 DEBUG - [ messagePool_rec_root ]checking OK.
2008-04-20 18:47:53,122 DEBUG - [ messagePool_err_to_root ]checking pool file consistency...
2008-04-20 18:47:53,122 INFO - [ messagePool_err_to_root ]pool Consistency Checking...
2008-04-20 18:47:53,122 INFO - [ messagePool_err_to_root ]checking hdi file...
2008-04-20 18:47:53,122 INFO - [ messagePool_err_to_root ]checking hdi file ok.
```

集成对象日志

另外，还提供了日志下载功能，用户可以将客户端日志保存到本地指定目录。只需点击 log 日志列表上方的日志下载链接，然后选择保存路径即可。

日志文件列表 >> 江苏烟草专卖系统

共计1个日志文件 [江苏烟草专卖系统日志下载](#)

[Agent2.out.xielf.rode.log](#)

日志下载

3.3.4.1.4 服务器日志查看

登录后单击左侧导航栏中的某一交换服务器节点 > 服务器监控 > 服务器日志查看，将会出现所选中的交换服务器节点下的日志文件列表，再从这个服务器日志文件列表中选择对日志信息查看。

服务器端日志列表

服务器端日志列表管理监控

北京烟草日志下载

服务器 北京烟草 日志列表

[deserver.rode.08.04.21_02.24.15.421.log](#)

[deserver.rode.08.04.21_07.42.16.984.log](#)

[deserver.rode.08.04.21_12.47.18.187.log](#)

[deserver.rode.08.04.21_16.15.51.937.log](#)

[deserver.rode.08.04.21_16.32.57.968.log](#)

[deserver.rode.log](#)

服务器日志查看

另外，提供了日志下载功能，用户可以将服务器上日志保存到本地指定目录。只需点击 log 日志列表上方的日志下载链接，选择保存路径即可。

3.3.4.2 异常消息管理

此功能主要是提供查询客户端和服务器端发送和接收的异常消息数和相应的异常消息 ID，并对出现异常的消息提供重发和删除操作。所谓异常消息是指客户端或服务器端接收和发送失败的消息，分为应用客户端消息和服务器消息两大类。

3.3.4.2.1 服务器端异常消息管理

提供当前交换服务器节点下受控的所有服务器异常消息的查看和管理。

要查看服务器消息列表页面，登录后选中当前的受控服务器节点名称，选择异常消息管理 > 服务器，进入服务器失败消息列表。可以根据各种查询条件查询异常消息，可以点击 **重试** 和 **删除** 按钮来重新发送和删除消息。如下图：

The screenshot shows a web-based application titled 'Exception Message Management'. At the top, there are several search and filter fields: 'Time' (from/to), 'Error ID', 'Message Type' (set to 'All'), 'Last Retry Time' (from/to), 'Retry Type' (set to 'All'), 'Retry Result' (set to 'All'), 'Message Identifier', 'Associated Key Identifier', and 'Message Title'. Below these are dropdowns for 'Message Exchange Mode' (set to 'All') and 'Abnormal Phase' (set to 'All'). There are also 'Search' and 'Clear' buttons. A toolbar below the filters includes icons for back, forward, refresh, and search, along with buttons for 'Retry' and 'Delete'. The main area is a table listing 32 exception messages. The columns are: Selection, Message Identifier, Associated Key Identifier, Message Type, Source, Abnormal Phase, Error Code, Time, Last Retry Time, Auto-Retryable, Retry Count, and Result. Each row contains a checkbox, the message identifier, the associated key identifier, the message type ('Request'), the source ('s22.s18.s...'), the abnormal phase ('Transmission'), the error code ('DEI3016' or 'DEI3018'), the time ('2010-08-26 15:...'), the last retry time ('2010-08-26 15:...'), a green checkmark indicating it's auto-retryable, a value between 1 and 4 for the retry count, and the result ('Success'). At the bottom of the table, there are navigation links for page numbers and a note indicating 1-20 of 32 total records.

异常消息

点击列表中消息的关联键值可查看该异常消息的明细信息。如下图：

服务器端异常消息详细信息	
跟踪	
事件属性	值
关联键标识	rk0826003
消息标识	7a047a04-12aad2d965b-0f34e3c0d894a7fb1ca6354eec48cb51
消息大小	807668
消息交换模式	MessageExchangeMode_InOnly
消息类型	messageExchangeDirection_req
消息标题	tt0826003
消息schema	sc1
前一节点路由名称	s18.s16
前一节点类型	1
异常阶段	传输
错误代码	DEI3016
错误描述	HTTP方式发送消息异常,连接不到服务器或者链接超时,服务器URL为:{0}
时间	2010-08-26 15:03:29,906
上次重试时间	2010-08-26 15:26:18,234
重试次数	3
重试结果	成功
下一节点路由名称	out16
下一节点类型	客户端
消息池名称	err_to_out16
URL	http://192.9.107.22:9082/out16/DETServlet
源EI服务器路由名称	s22.s18.s16
源连接器ID	in
目的地EI服务器路由名称	s16
目的地连接器ID	out16
[DEI3016] HTTP方式发送消息异常,连接不到服务器或者链接超时,服务器URL为:.....	

异常消息明细

点击明细页面的“跟踪”按钮，可跟踪该消息的整个交换过程。

3.3.4.2.2 客户端异常消息管理

提供当前交换服务器节点下受控的所有应用客户端异常消息的查看和管理。要查看客户端页面，登录后选中当前的受控服务器节点名称，选择异常消息管理 > 客户端，进入客户端消息汇总列表。进入页面后可以看见各个应用客户端的名称列表及每个应用客户端的异常消息数，异常消息数是未知的情况说明远程访问此应用客户端服务失败。

应用客户端异常消息管理			
应用客户端异常消息管理			
连接器名称	连接器地址	发送异常数	接收异常数
fl.out.liaoxd	http://192.9.107.55:8080/fs_out	未知	未知
Agent2.out.xielf	http://192.9.107.22:8080/Agent2out	未知	未知
ftp.out.liaoxd	http://192.9.107.55:8080/ftp_out	未知	未知

客户端信息管理

点击链接，出现此客户端的发送失败消息或接收失败消息列表。如下图：



依据消息 ID 复选列表中的消息 ID，可以点击 **重试** 和 **删除** 按钮来重新发送和删除消息。点击列表中消息的关联键值可以查看该消息的明细信息。

3.3.4.2.3 异常消息自动重试

服务器端异常消息和客户端异常消息还可以进行自动重试，即对于服务器判断可以进行自动重试的异常消息，按照规则自动进行出现异常的消息的重新发送或接收，不用人手工进行参与。

自动重试一般适用于在发送消息时出现 EI 服务器未启动，或客户端应用未启动而出现的异常消息。在异常消息列表中，“是否可自动重试”字段显示为 的消息可以进行自动重试。使用自动重试功能可以很大程度的降低在服务器或者网络出现故障时，进行异常消息重试的工作量，保证消息交换的高效和及时。

自动重试又分为短重试和长重试，短重试两次重试之间间隔时间短，长重试两次重试之间间隔时间长。自动重试时先进行短重试，当到达设置的最大次数仍然没有重试成功时进行长重试，这样可保证能够重试成功的消息尽早发送出去，长重试超过设定的最大次数仍未成功时则不再进行自动重试，等待人工进行干预。

使用自动重试时，可通过配置文件来设置自动重试的如上所述的属性，配置文件为 \${RCLOUDHOME}/DataExchange 目录下的 RODEConfig.xml 文件，其中，自动重试相关的属性及其意义如下：

```
<MessageRetry>
```

```
<NeedMsgAutoRetry>false</NeedMsgAutoRetry><!--是否需要消息自动重试 true|false, 若设置成 true 则开启自动重试功能-->

<ShortRertyMaxTimes>3</ShortRertyMaxTimes><!--单位: 次; 短重试最大次数-->

<LongRertyMaxTimes>3</LongRertyMaxTimes><!--单位: 次; 长重试最大次数-->

<ShortRertyInterval>5</ShortRertyInterval><!--单位: 分钟; 短重试时间间隔, 默认为 5 分钟-->

<LongRertyInterval>120</LongRertyInterval><!--单位: 分钟; 长重试时间间隔, 默认为 2*60=120 分钟-->

</MessageRetry>
```

NeedMsgAutoRetry: 表示是否需要消息自动重试, true 则开启自动重试, false 则关闭。

ShortRertyMaxTimes: 短重试的最大次数, 进入短重试后超过此次数进入长重试阶段。

LongRertyMaxTimes: 长重试的最大次数, 进入长重试后超过此次数不再进行自动重试。

ShortRertyInterval: 短重试两次重试之间时间间隔

LongRertyInterval: 长重试两次重试之间时间间隔

注: 关于自动重试次数的设定。自动重试是基于当前的问题可能会在一段时间内恢复的判断而进行的, 比如网络暂时中断或者服务器重启等情况, 所以自动重试存在成功的可能, 但并不是 100% 能重试成功, 比如网络一直中断自然不可能重试成功, 这个功能是为了尽可能避免人工干预, 而并非解决异常的万能法宝。所以, 建议根据实际情况将重试次数设置为合理水平, 而不是无限多次, 否则可能会造成问题的隐藏和服务器资源的严重浪费, 造成对正常业务的影响。

3.3.4.3 数据源配置

此功能主要是对数据源的管理以及给交换服务器端的定时任务和 java 扩展、SQL 扩展节点访问数据库时用，按数据源类型分为应用服务器数据源、普通 JDBC 数据源和 DE 连接池数据源。在 RCloud 云 EI 服务设计工具中和流程中的节点关联。

3.3.4.3.1 应用服务器数据源

应用服务器数据源即：由当前 RCloud 云 EI 服务服务器所在的应用服务器提供。要查看应用服务器数据源页面，登录后选中当前的受控服务器节点名称，选择数据源配置> 应用服务器数据源，进入服务器数据源管理列表。进入页面后可以查看服务器数据源列表，点击 **新建数据源** 和 **删除** 按钮分别添加和删除服务器数据源，数据源名称不能重复，是唯一标示。



数据源管理

3.3.4.3.2 普通 JDBC 数据源

要查看普通 JDBC 数据源页面，登录后选中当前的受控服务器节点名称，选择数据源配置> 普通 JDBC 数据源，进入普通 JDBC 数据源管理列表。

进入页面后可以查看普通 JDBC 数据源列表，点击 **新建数据源** 和 **删除** 按钮分别添加和删除服务器数据源。添加数据源时数据名不能重复而且所有项是必填项，数据源名称不能修改。

普通JDBC数据源

*数据源名称:

*用户名:

*密码:

*连接url: 例如: jdbc:db2://[ServerIP]:[Port]/[dbname]

*数据库驱动: 例如: com.ibm.db2.jcc.DB2Driver

提交 清空 取消

JDBC 管理

3.3.4.3.3 DE 连接池数据源

要查看 DE 连接池数据源页面，登录后选中当前的受控服务器节点名称，选择数据资源配置>DE 连接池数据源，进入 DE 连接池数据源管理列表。

进入页面后可以查看 DE 连接池数据源列表，点击 **添加**、**修改** 和 **删除** 按钮分别添加、修改和删除服务器数据源。添加数据源时数据名不能重复而且所有项是必填项，数据源名称不能修改。

DE连接池数据源

*数据源名称:

用户名:

*密码:

*连接url: 例如: jdbc:db2://[ServerIP]:[Port]/[dbname]

*数据库驱动: 例如: com.ibm.db2.jcc.DB2Driver

提交 清空 取消

连接池管理

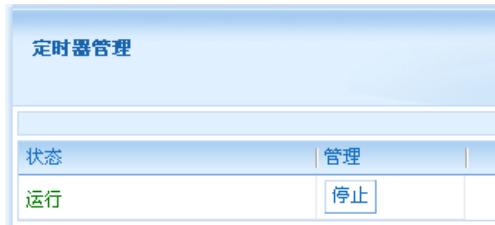
3.3.4.4 定时任务

此功能主要是提供查看定时服务器和定时任务的状态，并通过启停来管理定时服务器和定时任务。所谓定时任务是按着预先设定的时间来进行自动的消息触发和启动以及执行特定的任务，分定时任务控制和定时服务器控制两大类。

3.3.4.4.1 定时服务器控制

提供当前交换服务器节点下受控的定时服务器状态查看和管理。

要查看定时服务器控制页面，登录后选中当前的受控服务器节点名称，选择定时任务>定时服务器控制，进入定时服务器管理页面。进入页面后可以查看当前受控节点下的定时服务器状态并能作相应的启动和停止操作。



定时服务器控制

注意：只有当定时服务器启动的状态下，定时任务才能运行。

3.3.4.4.2 定时任务控制

提供当前交换服务器节点下受控的所有定时任务的查看和启停管理。

要查看定时任务页面，登录后选中当前的受控服务器节点名称，选择定时任务>定时服务器控制，进入定时任务列表。进入页面后可以查看各个定时任务的 ID、定时任务的名称及其状态列表，如果标题栏显示定时服务器已启动，这时既可以查看定时任务状态也能做相应的启动和停止操作。如果标题栏显示定时服务器已停止，则此时只能查看任务状态，**启动** 和 **停止** 按钮是灰色的不能操作。要想使**启动** 和 **停止** 按钮能操作，要先启动定时服务器。



定时任务控制

3.3.4.5 交换信息统计

统计 RCloud 云 EI 服务服务器、客户端和消息流的交换记录数和流量，可以按年、月、日查看，以图表的形式显示，使用户一目了然。

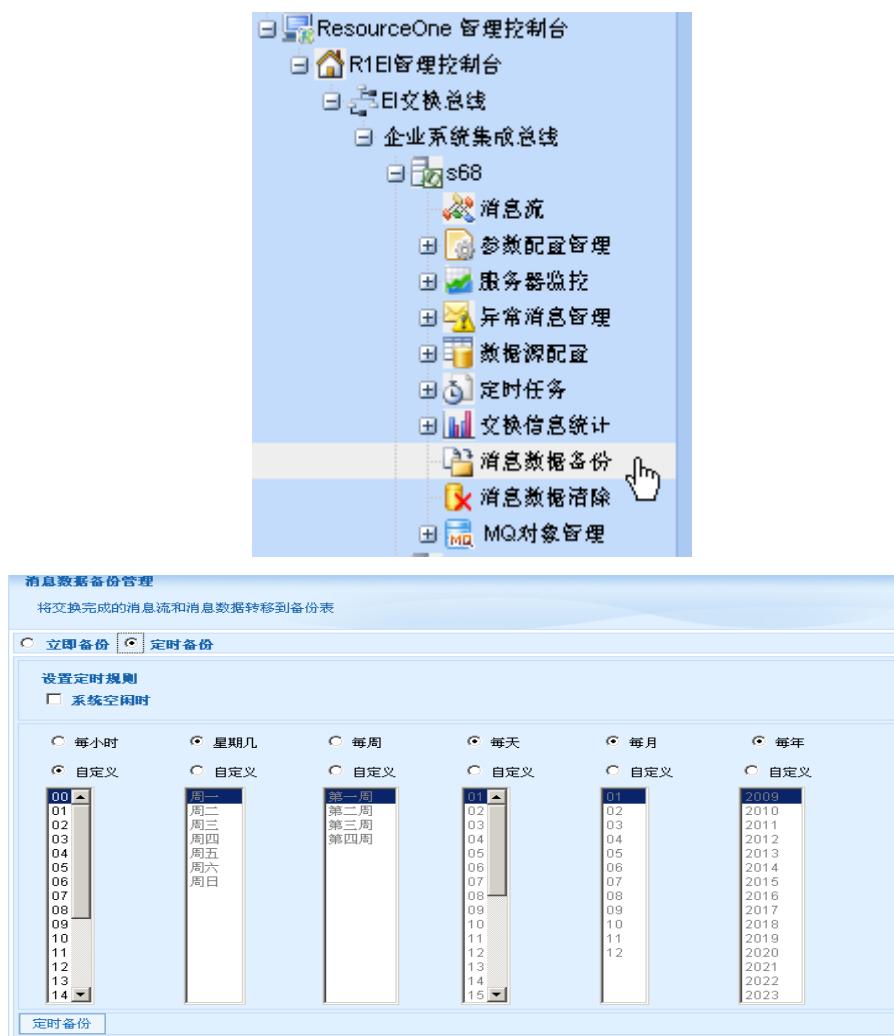


交换信息查看

3.3.4.6 消息数据备份

此功能主要是提供消息流的定时备份，这里仅仅提供简单的定时规则备份，间隔时间分别按年、月、周、日，也可以选择实时备份。

要查看消息数据备份页面，登录后选中当前的受控服务器节点名称，选择消息数据备份，进入数据备份管理。自定义定时规则，点击**定时备份**，就可以按照选择的规则进行定时备份。点击**立即备份**，系统立即执行备份。



消息数据备份

如果选择是定时备份，可以通过选择定时任务 > 定时任务控制，进入定时任务列表来管理这个任务。

如果“系统空闲时”规则被选中，RCloud 云 EI 服务服务器将在系统空闲时进行消息数据备份。判断系统空闲的条件有：JVM 内存使用率、当前系统线程池中正在执行的任务数、当前系统线程池中等待执行的任务数、当前系统运行的消息流数。

数据备份的定时任务有个固定 ID 是 backupDB，通过点击 **启动** 和 **停止** 按钮来启、停此数据备份的定时任务。

3.3.4.7 消息数据清除

此功能主要是提供消息流的定时清除，这里仅仅提供简单的定时规则清除，间隔时间分别按年、月、周、日，也可以选择实时清除。

要查看消息数据清除页面，登录后选中当前的受控服务器节点名称，选择消息数据清除，进入数据清除管理。自定义定时规则，点击**定时清除**，就可以按照选择的规则进行定时备份。点击**立即清除**，系统立即执行清除。

清除的消息可以使备份的消息数据，也可以普通的消息数据，可以进行立即清除以及定时的清楚，操作和消息数据备份过程类似，界面如图



消息数据清除

3.3.4.8 消息流管理

消息流管理模块主要提供管理和监控当前服务器上流程状态的功能。如流程绑定、流程删除、启用或停用代理服务流程等。

The screenshot shows the Rcloud Management Console's left navigation tree. Under 'ResourceOne 管理控制台' > 'R1EI 管理控制台' > '日交换总线' > '企业系统集成总线' > 's68' > '消息流', a context menu is open over the '消息流' item. The menu items are: 参数配置管理 (Parameter Configuration Management), 服务器监控 (Server Monitoring), 异常消息管理 (Exception Message Management), 数据源配置 (Data Source Configuration), 定时任务 (Timed Tasks), 交換信息统计 (Exchange Information Statistics), 消息数据备份 (Message Data Backup), 消息数据清除 (Message Data Purge), and MQ 对象管理 (MQ Object Management). Below this, the '消息流管理' (Message Flow Management) window is displayed. It has a header stating: '消息流的管理包括输入输出节点和连接器绑定管理，和单一服务调用输入节点类型的消息流发布为WEB服务管理' (Message flow management includes input/output node and connector binding management, and publishing single service call input nodes as WEB service management). The main area is a table titled '消息流列表' (Message Flow List) with columns: 序号 (Sequence Number), 名称 (Name), 消息流ID (Message Flow ID), 版本 (Version), 服务 (Service), and 管理 (Management). The table contains four rows:

序号	名称	消息流ID	版本	服务	管理
1	代理服务消息流定义	flow_discount	1.0	DiscountProxy_ 已启用	绑定管理 删除
	服务名	DiscountProxy		代理服务名称	
	服务接口定义 (WSDL)	DiscountProxy.wsdl		点击查看WSDL内容	
	备注			服务的描述信息	
	状态	已启用		停用	

流程绑定：将当前流程和连接器绑定（参考 [3.3.3.3.6](#)）。

删除：删除当前流程，删除前请仔细确认当前流程是否不在使用。

服务启用/停用：此功能只适用于代理服务流程。普通流程没有此功能。只有代理服务消息流是启动状态，代理服务才能够被访问。

3.3.4.9 参数配置管理

3.3.4.9.1 系统参数配置

配置 RCloud 云 EI 服务服务器在运行时需要使用的参数的值。

入口：RCloud 管理控制台左侧总线下每个注册的服务器下面，都有一个“参数配置管理”入口。点击进入系统参数配置面板，点击“系统参数”，用户可以根据实际运行情况来优化系统参数。



目前可配置的参数有：

参数名	描述	默认值
线程池中正在运行的任务数阀值	判断当前 RCloud 云 EI 服务系统是否空闲的参考值之一，任务执行线程池中的实际正在运行的线程数多于该数值时，认为系统处于繁忙状态。填写一个整数值。0 表示不作为判断条件。	0
线程池中等待执行的任务数阀值	判断当前 RCloud 云 EI 服务系统是否空闲的参考值之一，任务执行线程池中的等待运行的线程数多于该数值时，认为系统处于繁忙状态。填写一个整数值。0 表示不作为判断条件。	0
正在运行的消息流实例数阀值	判断当前 RCloud 云 EI 服务系统是否空闲的参考值之一，当前正在运行的消息流实例数多于该数值时，认为系统处于繁忙状态。填写一个整数值。0 表示不作为判断条件。	0
当前 JVM 堆内存使用率	判断当前 RCloud 云 EI 服务系统是否空闲的参考值之一，当前 JVM 内存使用率大于该数值时，认为系统处于繁忙状态。填写一个 0 到 100 的数值	30
检查系统是否空闲的时间间隔	用于空闲时消息数据备份任务，每隔多长时间检查一次 RCloud 云 EI 服务系统是否空闲。单位为毫秒。	60000
最多检查多少次系统是否空闲	用于空闲时消息数据备份任务，检查 RCloud 云 EI 服务系统是否空闲，检查该值次数后，如果系统仍然处于繁忙状态，就不再执行检查，备份任务结束。	10
单向消息流引擎状态	用于控制单向消息流引擎是否可用，启用状态值为：udrouter.state.enabled，停用状态值为：udrouter.state.disabled	udrouter.state.enabled

是否缓存消息流实例	运行时是否缓存消息流实例，启用缓存会提高运行时性能。启用状态值为：1，禁用状态值为：0	1
是否缓存消息流定义	运行时是否将需要的消息流定义缓存，启用会提高运行时性能，但更新消息流定义时，需要重启 RCloud 云 EI 服务服务器。启用状态值为：1，停用状态值为 0	1
是否缓存消息流环节定义	运行时是否将需要的消息流环节定义缓存，启用会提高运行时性能，但更新消息流定义时，需要重启 RCloud 云 EI 服务服务器。启用状态值为：1，停用状态值为 0	1
是否缓存消息流迁移线定义	运行时是否将需要的消息流迁移线定义缓存，启用会提高运行时性能，但更新消息流定义时，需要重启 RCloud 云 EI 服务服务器。启用状态值为：1，停用状态值为 1	1

3.3.4.9.2 RCloud 统一资源库管理

管理注册到当前 RCloud 云 EI 服务服务器上的 RCloud 统一资源库，可以进行增加、修改、删除操作。RCloud 云 EI 服务服务器在运行时会通过注册到当前 RCloud 云 EI 服务服务器上的 RCloud 统一资源库地址访问资源。

入口：

The screenshot shows the 'Parameter Configuration Management' interface for the R1 EI service. It includes sections for 'Message Forwarding Strategy', 'Exception Notification', 'Transmission Options', 'Automatic Retransmission', 'System Parameters', 'Audit Log', and 'Log Configuration'. The 'Unified Resource Library' section is highlighted with a red box. Below it, the 'Unified Resource Library Configuration Management' interface is shown, where a new resource library is being added with the identifier 'r1grs' and URL 'http://192.9.107.10:9082/gr'.

(RCloud 统一资源库注册)

输入框说明如下：

资源库标识：注册到当前 RCloud 云 EI 服务服务器的唯一标识， RCloud 云 EI 服务通过该标识查找访问地址。（必填）

资源库名称：资源库的显示名称

资源库 URL：资源库访问地址。如 <http://host:port/context> (必填)

描述：简单对该资源库进行说明。

3.3.4.9.3 审计日志参数配置

配置审计日志是否被记录。在消息跟踪功能模块的审计日志列表中可以看到一个交换过程在当前服务器上的审计日志列表。

用户根据实际情况在需要记录的点后面打上“√”，消息在执行到该点时会自动记录执行情况。

3.3.4.9.4 消息转发策略配置

转发策略的配置主要是在一个 EI 服务器中的 \RCloudHome\DataExchange\forwardpolicy 中配置转发策略文件，用来在消息到达此 DE-I 服务器上时与之匹配并转发到其他的 DE-I 服务器或连接器。

在管控中提供了策略文件的添加、修改、复制、删除、启用、停用、导入、导出功能。其中：

添加：添加一个消息转发策略；

复制：复制转发策略，可以对复制出的转发策略进行修改；

删除：删除一个或多个转发策略；

启用：只有处于启用状态的策略才可以约束消息是否可以进行转发；

停用：处于停用状态的策略则没有约束作用；

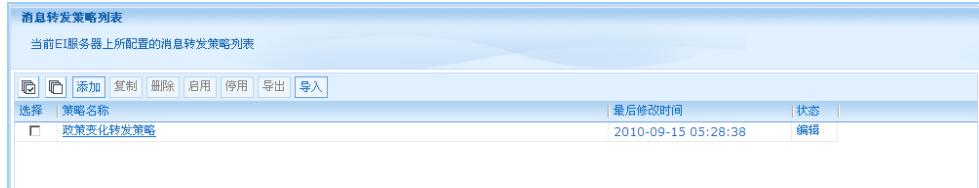
导入：将本地的策略文件导入到 EI 服务器上；

导出：将 EI 服务器的策略导出为本地文件。

消息转发策略配置的入口是管控中左侧导航树中 EI 服务器下面的“参数配置管理”->“消息转发策略”，如下图：



点击进入下图所示的消息转发策略列表：



1、添加及修改

在转发策略列表页面中，点击“添加”按钮可以进行消息转发策略文件的添加，点击列表中的策略名称可以对策略文件进行修改，添加或修改完毕之后，

点击下图中的“**保存转发策略**”将转发策略保存到 EI 服务器上：



配置消息转发策略时，从“匹配条件”和“转发目的地”两部分进行配置，配置结束后点击“保存”按钮可以将消息转发策略文件保存到 EI 服务器上；只有消息满足一个消息转发策略的“匹配条件”时，消息才会被自动转发到该策略中定义的转发目的地。

配置“匹配条件”，点击上图中的“添加”按钮进行添加，或点击某个匹配条件

名称进行修改，匹配条件分为四种：业务对象类型、内容、源地址、目的地地址，在下图中可以选择“匹配条件类型”下拉框中的类型分别进行配置：



配置“转发目的地”，如下图，切换到“转发目的地”标签，点击“添加”按钮进行添加，或点击某个 EI 服务器地址进行修改。



“地址类型”下拉框：有“包含”和“排除”两个选择，“包含”是指要转发的目的地地址，“排除”是指不转发到的目的地地址。

“EI 服务器地址类型”下拉框：有“服务器别名”和“路由名称”两种。选择“服务器别名”是指“EI 服务器地址”输入框中填的值是服务器的别名；选择“路由名称”是指“EI 服务器地址”输入框中填的值是服务器的路由名称。

“EI 服务器确切地址”复选框：指“EI 服务器地址”输入框中填的值是否是确切的地址。勾选指填的值是精确地地址，不勾选则指填的值是模糊的地址。

“EI 服务器地址”输入框：填写 EI 服务器的路由名称或别名。

连接器地址及消息处理器相关的三个控件含义同 EI 服务器类似。

2、复制

在转发策略列表页面中，选中一个转发策略，然后点击“复制”按钮，复制此转发策略的内容到新的转发策略中，可以对新的策略进行修改。

消息转发策略列表

当前EI服务器上所配置的消息转发策略列表

选择	策略名称	最后修改时间	状态
<input checked="" type="checkbox"/>	政策变化转发策略	2010-09-15 05:43:16	停用
<input type="checkbox"/>	政策变化转发策略2	2010-09-15 05:31:06	编辑

3、删除

在转发策略列表页面中，选中一个或多个转发策略，然后点击“删除”按钮，对转发策略进行删除。

消息转发策略列表

当前EI服务器上所配置的消息转发策略列表

选择	策略名称	最后修改时间	状态
<input checked="" type="checkbox"/>	政策变化转发策略	2010-09-15 05:43:16	停用
<input type="checkbox"/>	政策变化转发策略2	2010-09-15 05:31:06	编辑

4、启用

在转发策略列表页面中，选中一个或多个处于停用状态的转发策略，然后点击“启用”按钮，启用转发策略。

消息转发策略列表

当前EI服务器上所配置的消息转发策略列表

选择	策略名称	最后修改时间	状态
<input checked="" type="checkbox"/>	政策变化转发策略	2010-09-15 05:43:16	停用
<input type="checkbox"/>	政策变化转发策略2	2010-09-15 05:31:06	编辑

5、停用

在转发策略列表页面中，选中一个或多个处于启用状态的转发策略，然后点击“停用”按钮，停用转发策略。



6、导出

在转发策略列表页面中，选中一个或多个处于转发策略，然后点击“导出”按钮，会出现文件下载对话框，可以点击保存，将转发策略以 zip 包的形式保存。



7、导入

在转发策略列表页面中，点击“导入”按钮，导入 zip 包中的转发策略文件。



点击“下一步”：

导入消息转发策略，第二步：

zip包中消息转发策略如下表所示，请根据需要选择下一步操作。

转发策略名称	可执行性	后续操作
zll.xml	选择后续操作	<input checked="" type="radio"/> 覆盖(导入这些转发策略) <input type="radio"/> 忽略(不导入这些转发策略)

[上一步](#) [确定](#) [取消](#)

选择“覆盖”或“忽略”，点击确定：

导入消息转发策略，第三步：

您所选zip包中的消息转发策略导入结果如下表所示：

转发策略名称	操作结果
zll.xml	导入成功

[返回](#)

3.3.4.9.5 异常通知

异常通知是指当消息在交换过程中出现异常时，由 EI 发送通知给相关的管理人员的功能。目前提供两种形式的异常通知：邮件和 Webservice，前者通过发送邮件到指定邮箱来通知异常，后者可以调用自定义实现的异常通知服务来通知异常。在管理控制台中配置异常通知时需要的信息，如用于发送邮件的邮箱、收件人地址，或者是调用的服务信息等。异常通知配置要点如下：

1、在服务器端进行异常通知配置，当消息在该服务器出现异常时发送通知给服务器相关人员。

2、在客户端进行异常通知配置，当该客户端发送的消息出现异常时通知给该客户端相关的人员。

服务器端异常通知配置

服务器端异常通知配置入口为：管控->左侧导航树->EI 服务器下面的“参数配置管理”->“异常通知”，如下图：



点击后进入异常通知配置列表，如下图：

The screenshot shows the 'Exception Notification Configuration Management' list page. The table header includes columns for Selection, Name, Type, Status, and Description. The data rows show the following configurations:

选择	名称	类型	状态	描述
<input type="checkbox"/>	通知配置11	邮件	停用	desc11
<input type="checkbox"/>	A	邮件	启用	
<input type="checkbox"/>	贵阳市异常通知配置	邮件	启用	贵阳市异常通知配置
<input type="checkbox"/>	贵州省异常通知配置	服务	停用	贵州省异常通知配置
<input type="checkbox"/>	服务通知	服务	停用	

点击“添加”按钮添加配置，如下图：

The screenshot shows the 'Add Exception Notification Configuration, Step 1' page. The form fields are:

- 异常通知类型: (Exception Notification Type): 邮件 (Email) selected. Help text: 选择发送异常通知的方式.
- 名称: (Name): . Help text: 最大长度64.
- 描述: (Description): . Help text: 最大长度256.

Buttons at the bottom: 下一步 (Next), 上一步 (Previous), 确定 (Confirm), 取消 (Cancel), 测试 (Test).

异常通知类型：选择“邮件”或者“服务”

名称：填写异常通知配置的名字

描述：异常通知配置的其他信息

点击“下一步”

一、邮件通知配置

如果通知类型选择的为“邮件”则页面如下所示：

新建异常通知配置
请填写异常通知配置基本信息

账户类型：	POP3	*	目前只支持POP3方式
发送邮件服务器(SMTP)：			输入SMTP服务器地址
SMTP端口：	25	*	输入SMTP服务器端口
接收邮件服务器(POP3)：			输入POP3服务器地址
POP3端口：	110	*	输入POP3服务器端口
电子邮件地址：			输入用于发送的电子邮件地址
用户名：			输入用于发送的电子邮件用户名
密码：			输入用于发送的电子邮件密码
<input checked="" type="checkbox"/> 要求使用安全密码验证(SPA)登录			
通知接收地址：			填写异常通知接收方的邮件地址，如多个用;分隔
<input type="button" value="下一步"/> <input type="button" value="上一步"/> <input type="button" value="确定"/> <input type="button" value="取消"/> <input type="button" value="测试"/>			

账户类型：目前只支持 POP3

SMTP 服务器/SMTP 端口/POP3 服务器/POP3 端口：填写用于发送和接收邮件的服务器及端口

电子邮件地址：填写用于发送的电子邮件地址

用户名/密码：填写用于发送的电子邮件用户名和密码

通知接收地址：填写异常通知接收方的地址，可以填写多个，用; 分隔

二、服务通知配置

如果异常通知类型选择的为“服务”，则点击“下一步”后页面如下所示：

新建异常通知配置
请填写异常通知配置基本信息

服务地址：	<input type="text"/>	...	填写服务访问地址
服务名：	<input type="text"/>	...	填写服务名
操作名：	<input type="text"/>	...	填写服务中操作名称
命名空间：	<input type="text"/>	...	填写服务命名空间
SoapAction：	<input type="text"/>	...	填写服务消息动作
<input type="button" value="下一步"/> <input type="button" value="上一步"/> <input type="button" value="确定"/> <input type="button" value="取消"/> <input type="button" value="测试"/>			

点击各输入框后面的...可以查看对应项的帮助信息，如下图：

新建异常通知配置
请填写异常通知配置基本信息

服务地址：	<input type="text"/>	填写服务访问地址
服务名：	<input type="text"/>	填写服务名
操作名：	<input type="text"/>	填写服务中操作名称
命名空间：	<input type="text"/>	填写服务命名空间
SoapAction：	<input type="text"/>	填写服务消息动作

帮助
地址格式为：`http://{主机地址}:{端口}/{服务上下文根}/服务名,可从wsdl中获得,如图:`
`<wsdl:port name="TestServiceSOAP11port_http" binding="axis2:TestServiceSOAP11Bi<soap:address location="http://192.9.107.18:9080/rode/services/TestService" /></wsdl:port>`

[下一步](#) [上一步](#) [确定](#) [取消](#) [测试](#)

服务地址：填写异常通知服务的访问地址，如不清楚可查看服务相应的 wsdl

服务名：填写异常通知服务的服务名，如不清楚可查看服务相应的 wsdl

操作名：填写异常通知服务中的方法名，如不清楚可查看服务相应的 wsdl

命名空间：填写消息格式命名空间，如不清楚可查看服务相应的 wsdl

SoapAction：填写 SoapAction，如不清楚可查看服务相应的 wsdl

注：当选择异常通知类型为“服务”时，用户可调用自己开发的 WEB 服务进行异常通知，对服务名、操作名、命名空间不做限制，用户可以在管控中根据实际情况进行配置；但是，服务要求没有返回值，并且对服务中操作的参数有要求，参数如下表所示：

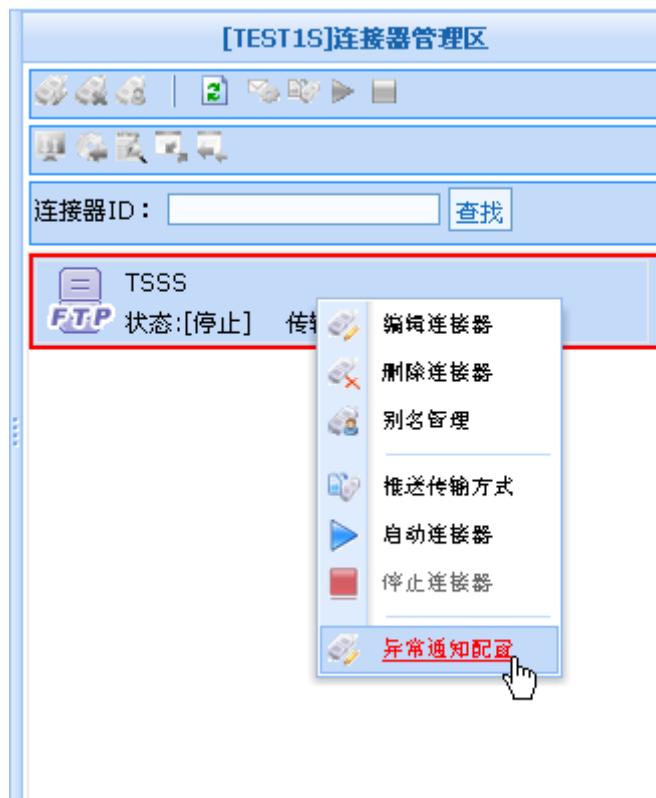
参数名	类型	描述
relationKey	String	消息在传输过程中唯一标识
messageID	String	消息 ID
schemaID	String	业务类型
title	String	消息标题
channel	String	传输通道
source	String	消息发送源地址 (s=,c=)
dest	String	如果是多目的地模糊地址，不返回值，而是用提示信息替换
errNode	String	发生错误节点
errTime	long	错误发生时间

errMsg	String	
extend	String	后续扩展用，如果需要其他信息，以 XML 方式存储

通知配置填写好后点击“确定”按钮。成功后回到异常通知配置列表页面。可点击某异常通知配置的名称进行修改；点击“启用”，“停用”按钮对异常通知进行启停操作，只有处于“启动”状态的异常通知配置才生效。

客户端异常通知配置

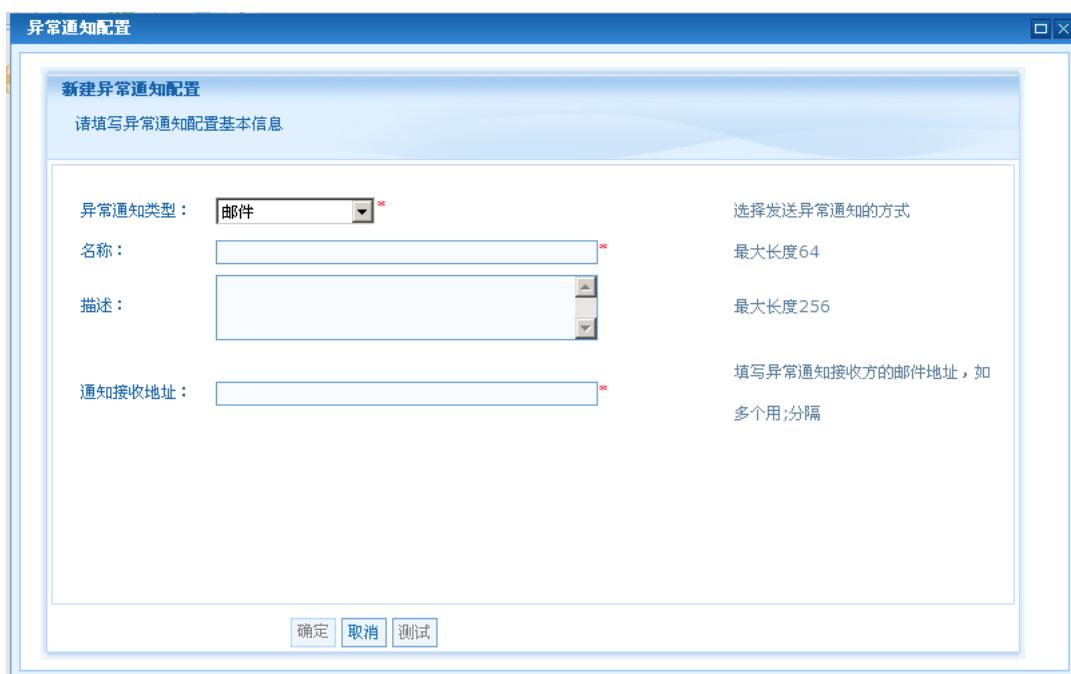
客户端异常通知配置入口为：管控->总线工作区->右侧连接器管理区->右键选择某连接器，选择“异常通知配置”，如下图：



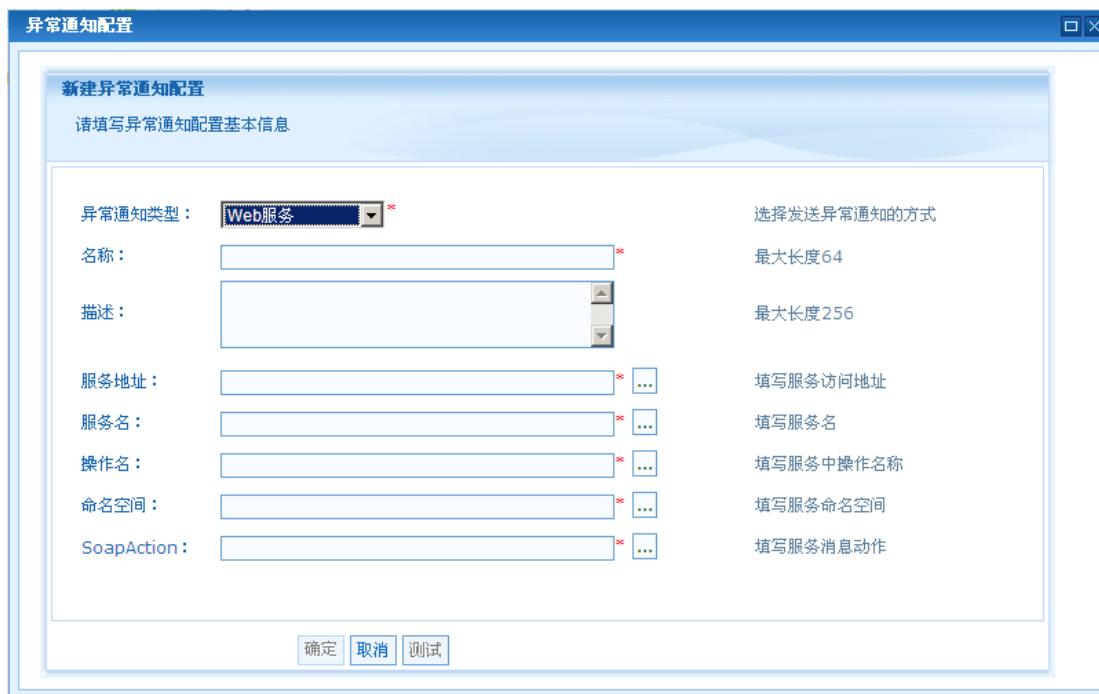
点击后进入异常通知配置列表，如下图所示：



添加、修改、删除与启用，停用等操作与服务器端大致相同，需要注意的是，在客户端，如果是“邮件”类型的异常通知，则不用填写发送邮件地址等相关信息，只要填写接收邮件地址即可，因为邮件是从服务器端发出，使用的是服务器端配置的发件地址。也就是说，当消息发送的过程中，经过一个或者多个服务器节点，如果在某一个节点出现了问题，则使用在该节点上配置的邮箱向客户端发送异常通知邮件，如下图所示：



如果异常通知类型为“服务”，则填写项与服务器端相同，如下图：



3.3.4.9.6 传输选项

当前 RCloud 云 EI 服务服务器上的传输参数如消息池侦听间隔、FTP 侦听间隔、附件最大值等参数可实时更新，不用重启当前服务器，即时生效。参数可以在传输选项中配置。

入口：



传输选项
配置消息传输相关参数

保存 重置 返回

配置项名称	配置项值	配置项描述
消息池侦听间隔	30 秒	对消息池进行侦听的时间间隔(单位:秒)
MQ队列侦听间隔	30 秒	对MQ队列进行侦听的时间间隔(单位:秒)
FTP侦听间隔	30 秒	对FTP侦听目录进行侦听的时间间隔(单位:秒)
MQ消息最大值	4096 K	MQ传输允许的消息最大值(单位:K)
HTTP连接超时时间	120 秒	HTTP建立连接超时时间(单位:秒)
附件最大值	4096 K	设置所携带附件的最大值(消息带附件传递时使用)
允许消息切片	是	是否允许对大消息进行切片
切片大小	512 K	消息体超过所设定值时, 对消息进行切片, 每片大小为所设定的值
MQ连接管理通道	RONEDE.SVRCONN	MQ管理服务器连接通道名称
是否需要异常反馈	否	消息交换异常时是否需要异常反馈
URL最大连接数	256	指定URL最大连接数

消息池侦听间隔：对消息池进行侦听的时间间隔(单位: 秒)

MQ 队列侦听间隔：对 MQ 队列进行侦听的时间间隔(单位: 秒)

FTP 侦听间隔：对 FTP 侦听目录进行侦听的时间间隔(单位: 秒)

MQ 消息最大值：MQ 允许的消息最大值(单位: K)

HTTP 连接超时时间：HTTP 建立连接超时时间(单位: 秒)

附件最大值：设置所携带附件的最大值(消息带附件传递时使用)

允许消息切片：是否允许对大消息进行切片

切片大小：消息体超过所设定值时，对消息进行切片，每片大小为所设定的值

MQ 连接管理通道：MQ 管理服务连接通道名称

是否需要异常反馈：消息交换异常时是否需要异常反馈

URL 最大连接数：指定 URL 最大连接数

3.3.4.9.7 消息自动重试

当前 RCloud 云 EI 服务服务器上的传输的消息出现异常时，如果这些消息是可重试类型消息，可以选择自动重试这些异常消息。重试参数可以在消息自动重试中配置。

入口：

参数配置管理

R1 EI 参数配置管理

消息转发策略 配置当前EI服务器的消息转发策略

异常通知 配置服务器端用来发送和接收异常通知的方式和具体信息

传输选项 配置消息传输相关参数

消息自动重试 配置消息传输发生异常时，消息自动重试参数

系统参数 配置EI服务器运行时的参数

统一资源库 配置当前EI服务器节点使用的R1统一资源库地址，一个EI服务器节点只能配置一个资源库地址

审计日志 配置在消息传输过程中，哪些操作需要被记入审计日志中

日志配置 配置控制台日志、文件日志参数

消息自动重试

配置消息传输发生异常时，消息自动重试参数

保存	重置	返回
配置项名称	配置项值	配置项描述
是否启动自动重试	<input checked="" type="checkbox"/> 是	是则启动重试，否则关闭
短重试最大次数	5 次	短重试达到次数上限时会进行长重试
长重试最大次数	5 次	长重试达到次数上限时状态变为手动重试
短重试间隔	5 分钟	单位：分钟；短重试时间间隔，默认为5分钟
长重试间隔	120 分钟	单位：分钟；长重试时间间隔，默认为 $2*60=120$ 分钟

是否启动自动重试：选择“是”将开启自动重试功能

短重试最大消息数：短重试达到次数上限时会进行长重试

长重试最大次数：长重试达到次数上限时状态变为手动重试

短重试间隔：(单位：分钟)；短重试时间间隔，默认为 5 分钟

长重试间隔：(单位：分钟)；长重试时间间隔，默认为 $2*60=120$ 分钟

3.3.4.9.8 日志配置

当前 RCloud 云 EI 服务服务器上的日志配置如日志级别、日志格式、日志文件位置、日志文件数量等参数可实时更新，不用重启当前服务器，即时生效。这些参数可以在日志配置页面中配置。

入口：

参数配置管理

R1 EI 参数配置管理

消息转发策略 配置当前EI服务器的消息转发策略

异常通知 配置服务器端用来发送和接收异常通知的方式和具体信息

传输选项 配置消息传输相关参数

消息自动重试 配置消息传输发生异常时，消息自动重试参数

系统参数 配置EI服务器运行时的参数

统一资源库 配置当前EI服务器节点使用的R1统一资源库地址，一个EI服务器节点只能配置一个资源库地址

审计日志 配置在消息传输过程中，哪些操作需要被记入审计日志中

日志配置 配置控制台日志、文件日志参数

日志配置

配置控制台日志、文件日志参数

保存 重置 返回

配置项名称	配置项值	配置项描述
控制台		
日志级别	info	设置控制台中日志信息的级别
日志格式	%d %p - %m%n	设置控制台中日志信息的格式
文件		
日志级别	info	设置文件日志信息的级别
日志位置	/ResourceOneHome/logs/DE-I	设置文件日志目录
日志文件最大值	1024 K	设置文件日志最大值
循环日志文件个数	10	循环日志文件个数
日志格式	%d %p - %m%n	设置文件日志信息的格式

日志级别： DEBUG、INFO、WARN、ERROR 和 FATAL

日志格式：可以按照日志标准进行日期、对齐方式、日志级别、换行设置日志

日志位置：日志文件产生的位置

日志最大值：单个日志文件的大小 (单位：K)

循环文件个数：日志位置文件夹中日志文件循环的个数。如 设置为 10，则第 11 个文件将覆盖第一个文件

3.3.4.10 MQ 对象管理

管理当前 RCloud 云 EI 服务服务器所使用到的 IBM Websphere MQ 队列和通道。

3.3.4.10.1 队列管理

管理 IBM Websphere MQ 消息队列，可在指定的队列管理器中进行队列的创建、修改、删除和查询操作。可管理的队列类型有：本地队列、传输队列和远程队列。要使用该模块功能，首先需要存在 RCloud 云 EI 服务专用的 MQ 队列管理器服务器连接通道（RONEDE.SVRCONN），另外，这里创建的队列只设置了基本的队列类型和主要参数定义，对队列和通道的详细配置和管理参数设置，需要参考 IBM-MQ 提供的管理文档在 MQserver 管理控制台设置。例如：队列触发器设置、队列监听间隔时间、进程定义等。

3.3.4.10.2 查询队列

从队列管理器下拉列表中选择一个队列管理器，点击“查询”按钮，就会在下面列出当前选中的队列管理器下所有的队列。

MQ队列管理			
管理IBM Websphere MQ消息队列，可在指定的队列管理器中进行队列的创建、修改、删除和查询操作。			
序号	队列名称	队列类型	队列深度
<input type="checkbox"/>	LQ_18_1	本地队列	0
<input type="checkbox"/>	LQ_18_2	本地队列	0
<input type="checkbox"/>	MsgCutOut	本地队列	0
<input type="checkbox"/>	Qout_wanggang	本地队列	47
<input type="checkbox"/>	RQ_18_55_1	远程队列	
<input type="checkbox"/>	RQ_18_55_2	远程队列	
<input type="checkbox"/>	RQ_18_57_1	远程队列	
<input type="checkbox"/>	RQ_18_57_2	远程队列	
<input type="checkbox"/>	TQ_18_55_1	传输队列	0
<input type="checkbox"/>	TQ_18_55_2	传输队列	0

3.3.4.10.2.1 添加队列

如果当前 MQ 队列管理器下没有所需要的队列，可以通过“添加”按钮添加一个新的队列。

需要填写的项有

队列管理器名：将队列创建到哪个 MQ 队列管理器下面。

本地队列名称：需要创建的队列名称

主机地址：队列管理器 IP

侦听端口：队列管理器访问端口。

队列类型：选择将创建何种类型的队列（本地队列或远程队列）。

消息具有持久性：选择该队列是否需要将经过该队列的消息持久化。

用法：指定该队列的用途。（正常或者传输）



3.3.4.10.2.2 修改和删除队列

修改：打开队列信息框，修改队列信息，点击“更新”保存。

删除：将需要删除的队列前打上“√”，点击上面的“删除”按钮删除。



(修改队列信息)



(删除队列)

3.3.4.10.3 通道管理

管理 IBM Websphere MQ 通道，可在指定的队列管理器中进行各种通道的创建、修改、删除、启动、停止、PING 和查询操作。可管理的通道类型有：发送方通道、接收方通道、请求通道、服务器通道和服务器连接通道。

3.3.4.10.3.1 查询通道

从队列管理器下拉列表中选择一个队列管理器，点击“查询”按钮，就会在下面列出当前选中的队列管理器下所有的通道。点击某一通道的通道名称，显示该通道的详细信息。

通道管理						
管理IBM Websphere MQ通道，可在指定的队列管理器中进行各种通道的创建、修改、删除、启动、停止、PING和查询操作。						
序号	通道名称	通道类型	连接名称	传输队列名称	状态	管理
□	CH_18_55_1	发送方	1417	TQ_18_55_1	正在运行	<input type="button" value="stop"/>
□	CH_18_55_2	发送方	1417	TQ_18_55_2	正在运行	<input type="button" value="stop"/>
□	CH_18_57_1	发送方	1415	TQ_18_57_1	正在运行	<input type="button" value="stop"/>
□	CH_18_57_2	发送方	1415	TQ_18_57_2	正在运行	<input type="button" value="stop"/>
□	CH_55_18_1	接收方			正在运行	<input type="button" value="stop"/>
□	CH_55_18_2	接收方			正在运行	<input type="button" value="stop"/>
□	CH_57_18_1	接收方			正在运行	<input type="button" value="stop"/>
□	CH_57_18_2	接收方			正在运行	<input type="button" value="stop"/>
□	RONEDE.SVRCONN	服务器连接			正在运行	

3.3.4.10.3.2 添加通道

点击“添加”按钮，打开添加通道窗口，填写必要信息，添加一个新的 IBM-MQ 通道。

需要填写的信息如下：

队列管理器名： 指定将通道创建到哪个 MQ 队列管理器下面。

通道名称： 需要创建的通道名称

主机地址： 队列管理器 IP

侦听端口： 队列管理器访问端口。

通道类型： 选择是建发送方通道还是接收方通道。如果通道类型是选择发送方通道，还需要填写以下信息：

连接地址： 远程队列所在 MQ 队列管理器 IP

连接端口： 远程队列所在 MQ 队列管理器访问端口。

传输队列名称：必须和该通道的远程队列名称一致。



3.3.4.10.3.3 修改、删除通道

操作和队列管理相似。



3.3.4.10.3.4 启动或停止通道

提供通道启动和停止操作，如果当前通道是运行状态，可以在管控中停止该通道，同样，如果状态是停止的，也可以在管控中启动通道。

注意：

由于队列和通道的管理是通过 IBM MQ 管理 API 实现的，所以，必须手动创建服务器传输通道后，才能使用 MQ 管理功能模块。

3.3.4.11 消息跟踪

3.3.4.11.1 普通消息跟踪

可以跟踪查看每一个经过当前总线传输的消息的执行情况（成功或异常），便于对消息的监控。

首先，点击总线下的“消息跟踪”（下图 1），会列出当前总线下的注册的 RCloud 云 EI 服务服务器列表，点击每个 RCloud 云 EI 服务服务器，会列出该 RCloud 云 EI 服务服务器下所有注册的连接器和代理服务，每个代理服务下都有两个子目录：发送的消息和接收的消息（如图 2），点击进入就会列出当前连接器（或代理服务）下发送（或接收）的消息列表（如图 3）。点击消息列表中每个消息还可以查看更详细的信息，如消息的传输路径、消息在每个交换节点上的执行情况等。



消息标识	关键标识	消息交换模式	消息类型	业务消息类型	时间
a06d88-11bf3ea2003-b16...	rk024	双向	请求		2008-08...
c48772-11bf3cf5f9f-b167e...	rk023	双向	请求		2008-08...
1f36af6-11bf390bac2-b16...	rk013	双向	请求		2008-08...
1f36af6-11bf37e1c95-b16...	rk012	双向	请求		2008-08...
1f36af6-11bf36b41ad-b16...	rk00008	单向	请求		2008-08...
1f36af6-11bf369cd4d-b16...	rk00007	单向	请求		2008-08...
1f36af6-11bf06ffa72-b167...	rk06	单向	请求		2008-08...
1f36af6-11bf05fdf3-b167...	rk05	单向	请求		2008-08...
1f36af6-11bf00700f9-b16...	rk002	双向	请求		2008-08...
1f36af6-11bef25a93a-b16...	rk001	双向	请求		2008-08...

第 1 - 10 条 共 10 条

(3)

3.3.4.11.2 转发消息跟踪

根据转发策略进行转发的消息的跟踪入口如下：

(1) 打开发送方符合转发策略条件的消息：

时间	消息标识	关键标识	消息交换...	消息类型	业务对...
2009-06-19 10:09:13.281	15529ff-12...	r1	单向	请求	tlam
2009-06-19 09:50:08.593	15529ff-12...	r1	单向	请求	tlam
2009-06-18 18:09:30.906	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 17:48:55.812	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 17:06:31.593	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 16:28:47.5	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 16:21:41.375	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 16:18:48.296	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 16:11:19.921	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 16:09:10.078	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 16:06:29.406	15529ff-12...	relationKey_zll_2	单向	请求	tlam
2009-06-18 15:58:10.968	15529ff-12...	relationKey_zll_1	单向	请求	

第 1 - 12 条 共 12 条

(2) 打开使用转发策略的交换节点：

消息交换路由路径信息列表			
消息交换路径列表，点击某个DE - I 服务器，查看消息在这个DE - I 服务器上的传输处理审计日志。			
返回	消息发送端	交换节点名称	消息接收端
	tlam_gjj_zll ->	国家局 S_18 -> 四川S_16 ->	tlam_sc_zll

(3) 在审计日志列表中打开“构造转发消息”：

消息审计日志列表						
步骤	消息标识	消息...	路由...	日志...	时间	
接收消息	15529ff-121f26710d0-b167e71512e...	请求	接入	正常	2009-06-18 16...	
发送消息	15529ff-121f26710d0-b167e71512e...	请求	接出	正常	2009-06-18 16...	
接收消息	15529ff-121f269849e-b167e71512e...	请求	接入	正常	2009-06-18 16...	
发送消息	15529ff-121f269849e-b167e71512e...	请求	接出	正常	2009-06-18 16...	
接收消息	15529ff-121f26b7fc3-b167e71512e1...	请求	接入	正常	2009-06-18 16...	
构造转发消息	59e659e6-121f26ab399-0f34e3c0d8...	请求	接入	正常	2009-06-18 16...	
发送消息	15529ff-121f26b7fc3-b167e71512e1...	请求	接出	正常	2009-06-18 16...	
发送消息	59e659e6-121f26ab474-0f34e3c0d8...	请求	接出	正常	2009-06-18 16...	
发送消息	59e659e6-121f26ab464-0f34e3c0d8...	请求	接出	正常	2009-06-18 16...	
接收消息	15529ff-121f272573a-b167e71512e...	请求	接入	正常	2009-06-18 16...	
发送消息	15529ff-121f272573a-b167e71512e...	请求	接出	正常	2009-06-18 16...	
接收消息	15529ff-121f274fb5f-b167e71512e1...	请求	接入	正常	2009-06-18 16...	
发送消息	15529ff-121f274fb5f-b167e71512e1...	请求	接出	正常	2009-06-18 16...	
接收消息	15529ff-121f27b7bec-b167e71512e1...	请求	接入	正常	2009-06-18 16...	
发送消息	15529ff-121f27b7bec-b167e71512e1...	请求	接出	正常	2009-06-18 16...	
接收消息	15529ff-121f29e080a-b167e71512e...	请求	接入	正常	2009-06-18 17...	
发送消息	15529ff-121f29e080a-b167e71512e...	请求	接出	正常	2009-06-18 17...	
接收消息	15529ff-121f2c4da55-b167e71512e1...	请求	接入	正常	2009-06-18 17...	
发送消息	15529ff-121f2c4da55-b167e71512e1...	请求	接出	正常	2009-06-18 17...	
接收消息	15529ff-121f2d7b2fb-b167e71512e1...	请求	接入	正常	2009-06-18 18...	
发送消息	15529ff-121f2d7b2fb-b167e71512e1...	请求	接出	正常	2009-06-18 18...	

(4) 在审计日志详细信息中点击“跟踪转发消息”按钮：

消息审计日志详细信息	
返回	跟踪转发消息
事件属性	值
步骤	构造转发消息
关联键标识	relationKey_zll_2
消息标识	59e659e6-121f26ab399-0f34e3c0d894a7fb1ca6354eec48cb51
消息大小	23
消息交换模式	单向
消息类型	请求
事件源	com.icss.ro.de.connector.TransportEvent
日志级别	正常
是否人工干预过	未人工干预
重试次数	0
前一节点路由名称	
前一节点类型	1
下一节点路由名称	Sichuan.gjj
下一节点类型	1
消息池名称	
路由引擎名称	
URL	192.9.107.16\$QM_0018\$1428\$RQ_18.16
回调处理类型	
状态	
源DE-I 服务器路由名称	gjj

(5) 在新弹出的页面中进行转发消息的跟踪:

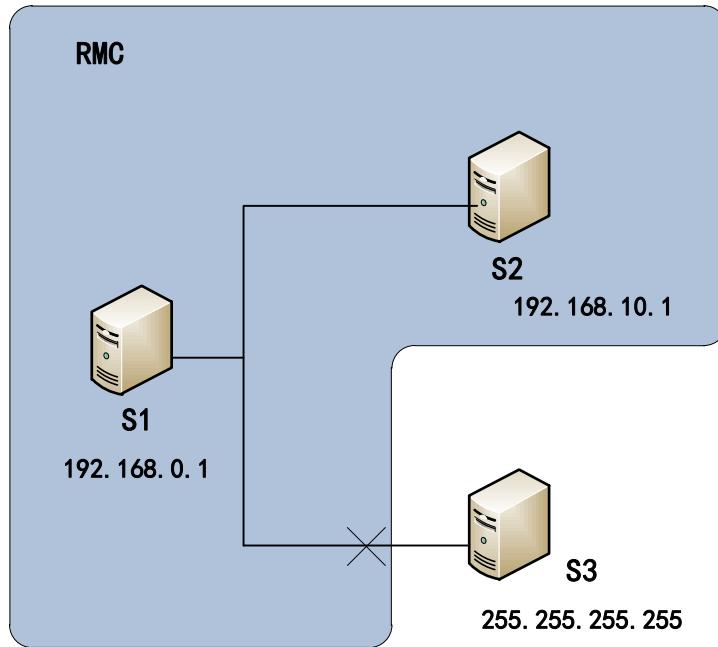
消息交换路由路径信息列表		
消息交换路径列表，点击某个DE - I 服务器，查看消息在这个DE - I 服务器上的传输处理审计日志。		
消息发送端	交换节点名称	消息接收端
交换服务节点[四川S_16] ->		
四川S_16 -> 成都S_10 ->		tiam_cd_zll
四川S_16 -> 乐山S_50 ->		tiam_ls_zll

3.3.5 多管控配置

从 V4.2 开始，RCloud 云 EI 服务中增加了配置多个管控管理同一条总线的能力。目前 RCloud 云 EI 服务提供了两种配置多管控的方式：从文件导入配置和从服务器加载配置。

3.3.5.1 从文件导入配置

从文件导入配置功能多用于原有的管理控制台无法直接连通路由表中某台 EI 服务器的场景。如下图所示：



操作步骤如下：

第一步：从原有的管控中导出总线配置。

第二步：在能够连通该 EI 服务器的物理主机上新部署配置一个管理控制台

(参见本文档 2.5 节中管理控制台安装相关内容)。

第三步：从文件导入总线配置。

3.3.5.1.1 导出配置

进入 EI 总线管理界面，选择一条需要导出的总线，点击“导出配置”按钮，保存弹出的 zip 包即可将所选的总线配置导出。如需导出多条总线配置，请重复该步骤。



导出总线配置

3.3.5.1.2 从文件导入配置

登录新部署配置好的管理控制台，进入 EI 总线管理界面，点击“从文件导入配置”进入配置导入向导。



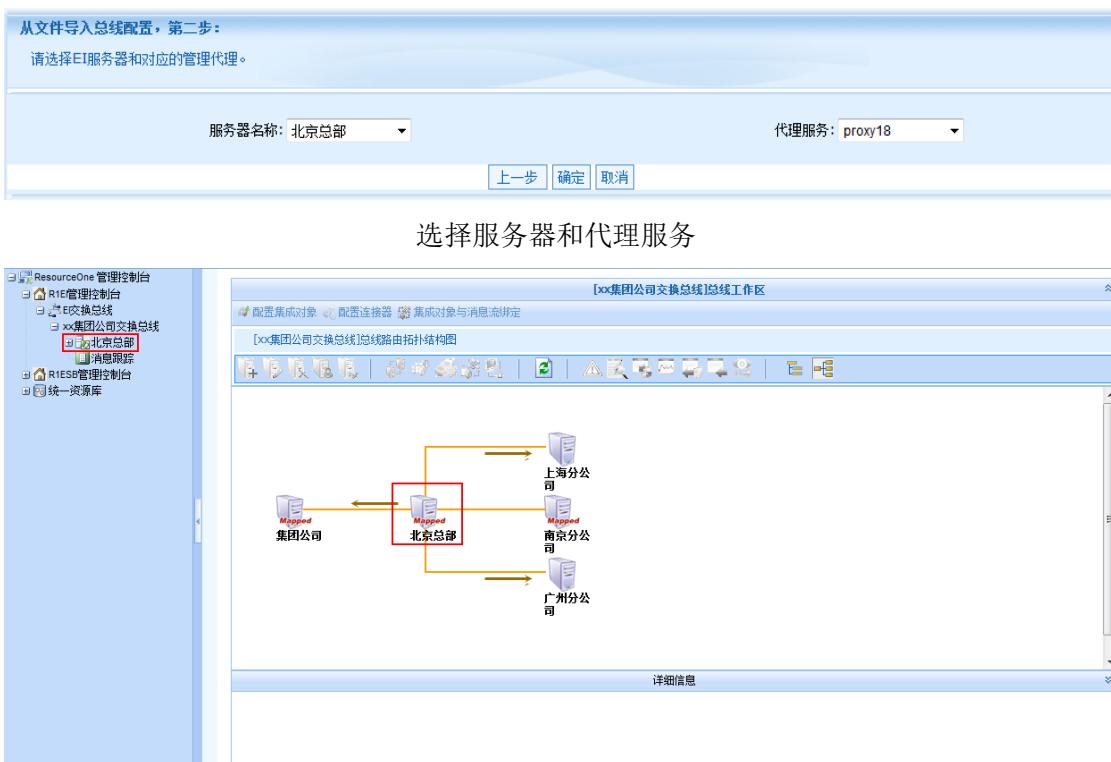
进入从文件导入配置向导

第一步，选择从原管控中导出的配置文件。



选择配置文件

第二步，选择路由表中需要管理的 EI 服务器和管理该 EI 服务器所对应的代理服务，点击“确定”导入总线配置信息，并为所管理的 EI 服务器生成左侧导航菜单。



从文件导入配置效果示意

注：如在第二步中不选择 EI 服务器和代理服务，则只导入总线配置信息，并只生成总线的导航菜单，而不生成 EI 服务器的导航菜单。如需生成 EI 服务器导航菜单，请从路由表中右击该 EI 服务器，选择“管理 EI 服务器”。

3.3.5.2 从服务器加载配置

从服务器加载配置功能多用于原有管理控制台部署配置了整个总线环境之后，增加一个管理控制台专门管理其中的某一台或者某几台服务器的场景。例如，某集团公司要求总部可以管理整条总线，各分公司都有自己的管理控制台，而且只能管理本公司的 EI 服务器。

操作步骤如下：

第一步：在能够连通该 EI 服务器的物理主机上新部署配置一个管理控制台（参见本文档 2.5 中管理控制台安装相关内容）。

第二步：登录新部署配置好的管理控制台，进入 EI 总线管理界面，点击“从服务器加载配置”进入向导。

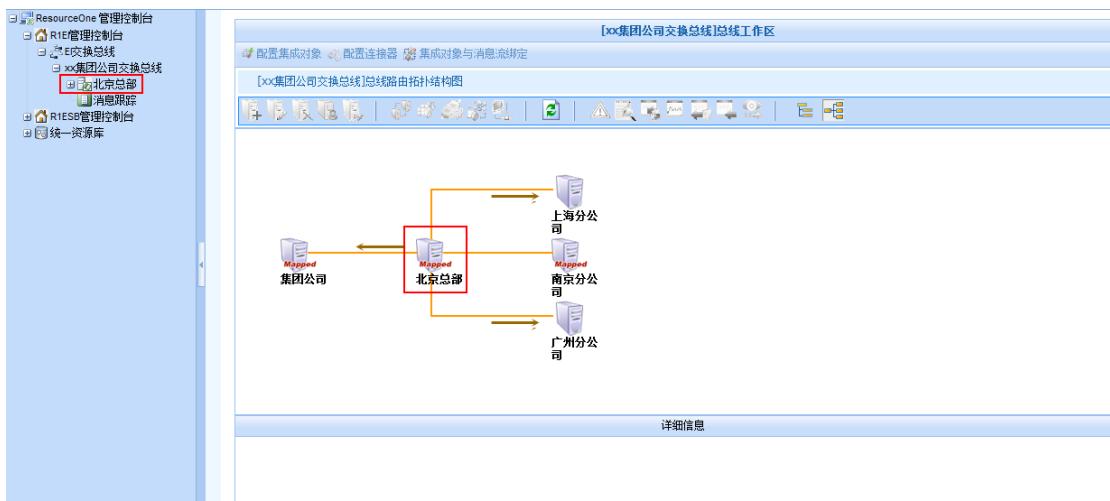


进入从服务器加载配置向导

第三步：选择需要管理的 EI 服务器对应的代理服务，点击“确定”加载总线配置信息，并为所管理的 EI 服务器生成左侧导航菜单。



选择代理服务

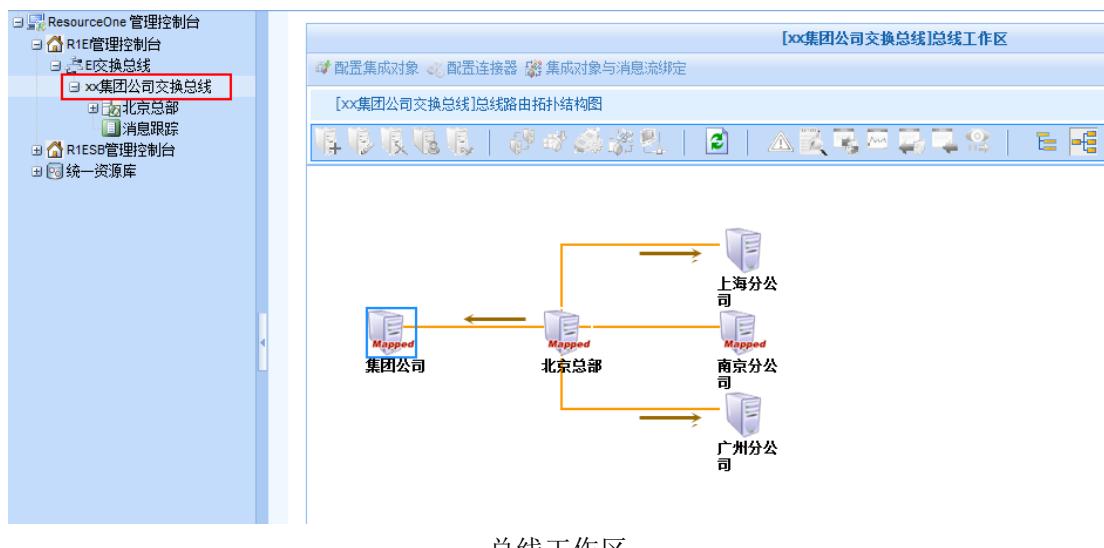


从服务器加载配置效果示意

3.3.5.3 管理 EI 服务器

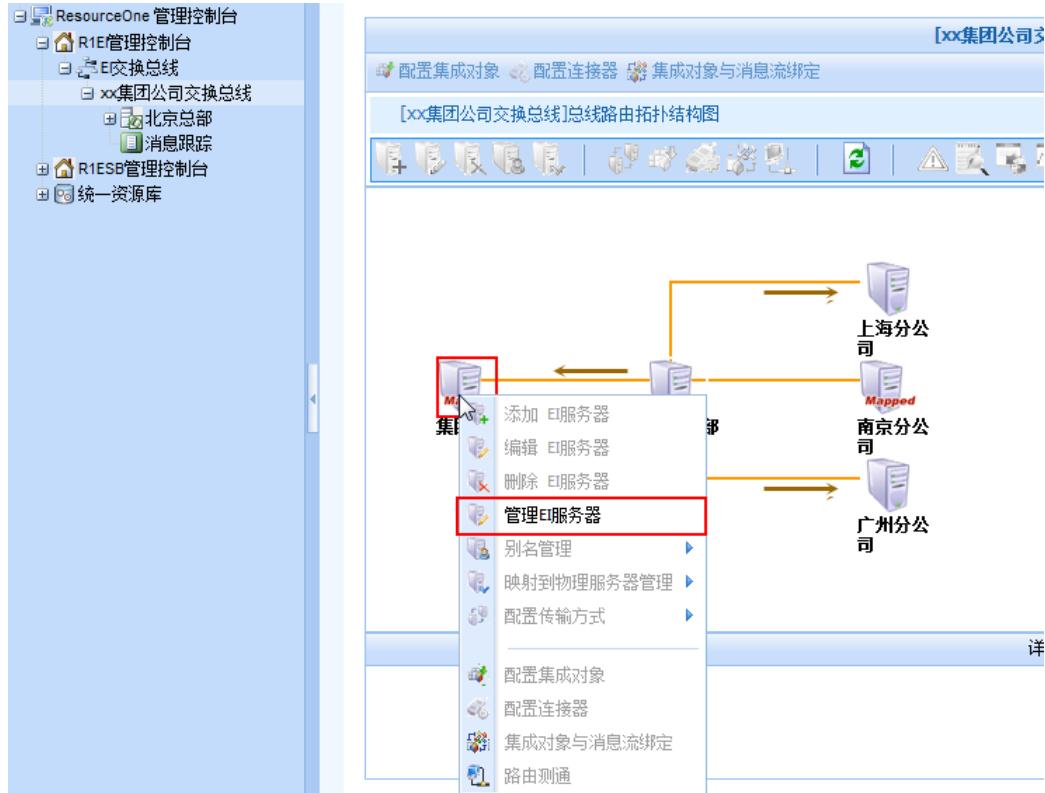
对于使用从文件导入或者从服务器加载方式生成的总线，往往只能管理其中某一台 EI 服务器，使用“管理 EI 服务器”功能则可管理总线中其他的服务器。

第一步，点击左侧导航菜单中的总线，进入总线工作区。



总线工作区

第二步，右击所需要管理的 EI 服务器，选择“管理 EI 服务器”。



管理 EI 服务器

第三步，选择该 EI 服务器对应的管理代理，点击“确定”，为所选 EI 服务
器生成左侧导航菜单。



第四章 注意事项及优化指南

4.1 总线部署使用时的注意事项

1、总线中每一个 EI 服务器路由信息的变化，都将引起整个总线的重新映射，即如果某个服务器的路由信息改变后，需要重新广播映射到其他 EI 服务器。

2、配置 DEAgent 连接器，文件连接器，FTPAgent 连接器时，应该保证 Agent 客户端应用正常运行并且是可访问的；配置完后，一定要执行推送操作，而且推送后，推荐用户最好要重启客户端应用；尽量避免消息流引擎的停止，以免引起消息交换错误，消息流异常恢复时，优先选择从当前环节恢复。

3、特别注意：通过管控，配置总线时，在未映射到物理服务器成功前，不

许进行其他操作，比如启用别名、配置链路等。

4、如果使用内部数据源，需要修改\RCloudHome\ DataExchange\RODEConfig.xml 中数据库连接配置，将<ManagerType>改为“2”，
配置文件默认为外部数据源。

如下图所示

```
<DBConfig>
    <ManagerType>1</ManagerType> <!-- 数据库连接管理类型：
Container:1,DE-I:2 -->
</DBConfig>
```

4.2 RCloud 云 EI 服务部署时的注意事项

4.2.1 RCloud 云 EI 服务服务注册

RCloud 云 EI 服务的 SOAP/WS 连接器中，接收消息的 Web 服务、长周期服务调用节点接收反馈消息的 Web 服务，需要在部署时注册到统一资源库中，服务名为 WSConnectionService。

部署方法为： RCloud 云 EI 服务 Server 部署完成后，通过访问 RCloud 云 EI 服务应用 <http://ip:port/context>，（注： ip 不能是 localhost/127.0.0.1），查找到该服务的 WSDL，并保存服务名为 WSConnectionService.wsdl。打开统一资源库的管理控制台，使用该 wsdl 发布一个资源，类型为“Web 服务”，资源名称和 ID 为 WSConnectionService，版本为 1.0。

4.2.2 DE Agent 客户端部署注意事项

EI 会提供一个 RCloudAgent 部署压缩包 RCloudDEAgent.zip，将压缩包里的 RCloudHome 文件夹拷贝到当前应用服务器所在跟目录下，需要注意一下几点：

1、压缩包里的 lib 包下是应用基于 RCloud 云 EI 服务开发的 SDK 包，需要

加载到客户端应用的编译环境。

2、压缩包里的 3rdlib 下是应用客户端需要的第三方 jar 包，需要加载到客户端应用的编译环境。

3、压缩包里的 wsdd 下是应用客户端需要的 webservice 部署描述符文件。需要放置在在客户端应用 WAR 包中的\WEB-INF 目录下。

4、压缩包里的 classes 目录下 rode.properties 文件，需要复制到 DEAgent 所在客户端应用的/web-inf/classes 目录下。

在 Agent 客户端的 web.xml 中注册 AxisServlet、SOAPMonitorService、DETServlet、ClientInitServlet 和 AxisHTTPSessionListener，详细参照部署手册中 Agent 客户端部署部分。当以上 4 步都完成后，将客户端应用部署到 WEB 容器中（如 websphere），访问 URL: http://[ip][:port]/[context]/services 如果能正常看到 webservice 服务列表页面，表示客户端部署成功。需要确认一下服务列表页面中列出的服务和 webservice 部署描述符文件中的服务是否相同。

4.2.3 RCloud 云 EI 服务服务器部署注意事项

1. 当使用 DB2 创建 RCloud 云 EI 服务服务器数据库时，需要先在 RCloudEI4.5_CreateTaleSpace.sql 脚本中扩大表空间的大小，然后再执行建表语句，否则有些表会创建不成功。
2. 如果应用服务器使用的是 weblogic，那么数据源的 JNDI 名字要用 RODE.DataSource。
3. 如果应用服务器使用的是 weblogic，一定要将 WAR 包解压后再部署，并且应用的上下文根在 weblogic.xml 中指定。
4. Websphere下部署需要将类加载顺序改为parent_last，即优先加载应用程序的类。

4.3 RCloud 云 EI 服务服务器配置建议

如果要交换的消息的大小超过 1.5M、并且会连续的、不间断的发送超过 100

个消息，不管有多少个流程和节点，那么建议把应用服务器的 jvm 最大占用内存堆调高至 512M – 1G。目前，Websphere 的 jvm 占用内存最大堆默认为 256M。



JAVA 虚拟机内存大小修改界面

附录一 RCloud 云 EI 服务 Web Service 访问安全配置

5.1 概述

RCloud 云 EI 服务中 Web 服务安全实现，支持基于 WS-Policy 的安全策略。通过 AXIS2 中提供的安全模块 Rampart 的使用，服务提供者可以对提供的服务设置口令、时间戳、数字签名和加密。

WS-Policy 可以在以下领域（domain）中使用。

- 安全
- 传输控制
- 应用程序优先权
- 用户帐户优先权

5.2 安装安全模块

首先，到 apache.org 上下载最新的 Rampart 包（目前最新的为 1.3），将 rampart 目录下 rampart-x.x.mar 和 rahas-x.x.mar 复制到 RCloud 云 EI 服务 Server 的 Web-inf/module 下。用文本编辑器打开 WEB-INF\conf\axis2.xml 并在<module ref="addressing"/>元素后添加<module ref="rampart"/><module ref="rahas"/>。至此已将基于 Apache wss4j 的 axis2 安全模块配置到服务器中了。

同样过程配置到客户端中。

5.3 配置服务安全策略

5.3.1 身份验证配置

WS-Security 支持用户名/密码、通过 X.509 证书的 PKI 、Kerberos 三种身份验证。用户安全性令牌（用户名/密码）是用于端到端之间传输时互相检验用户的访问安全口令，它包括客户端到服务器端和服务器端到客户端。下面以此

来说明配置过程（AXIS2 环境）：

5.3.1.1 密码生成摘要发送

1、提供口令处理类（服务器和客户端都需提供）

```
public class PWCBHandler implements CallbackHandler {  
  
    public void handle(Callback[] callbacks) throws  
    IOException, UnsupportedCallbackException {  
  
        for (int i = 0; i < callbacks.length; i++) {  
  
            WSPasswordCallback pwcb =  
            (WSPasswordCallback) callbacks[i];  
  
            String id = pwcb.getIdentifier();  
  
            if ("bob".equals(id)) {  
  
                pwcb.setPassword("bobPW"); //  
            }  
        }  
    }  
}
```

2、客户端 axis2.xml 配置

```
<module ref="rampart" /><!-- rampart 模块注册-->  
  
<parameter name="OutflowSecurity">  
  
    <action>  
  
        <items>UsernameToken </items>  
  
    <!--访问服务设置身份验证-->  
  
        <user>bob</user>  
  
    <!--指定发送消息的身份，需和 PWCBHandler 中一致-->  
  
        <passwordCallbackClass>  
  
    <!--口令提供类路径-->  
  
        org.apache.rampart.samples.sample02.PWCBHandler
```

```
</passwordCallbackClass>  
  
</action>  
  
</parameter>
```

3、服务器端 services.xml 配置

```
<service>  
  
    <operation name="echo">  
  
        <messageReceiver  
        class="org.apache.axis2.rpc.receivers.RPCMessageRec  
eiver"/>  
  
    </operation>  
  
    <parameter name="ServiceClass" locked="false">  
        org.apache.rampart.samples.sample02.SimpleService  
    </parameter>  
  
  
    <module ref="rampart" />  
  
    <!--对该服务使用 rampart，如果在 axis2.xml 中注册了  
    rampart 模块，这里就不需要再配置-->  
  
    <parameter name="InflowSecurity">  
  
        <action>  
  
            <items>  
  
                <!--访问该服务需要身份验证-->  
  
                UsernameToken  
  
            </items>  
  
            <passwordCallbackClass>  
  
                <!--口令提供类路径-->  
  
                org.apache.rampart.samples.sample02.PWCBHandler  
  
            </passwordCallbackClass>
```

```
</action>  
</parameter>  
</service>
```

5.3.1.2 密码明文发送

发送的 SOAP 消息中包含密码

1、提供口令处理类（服务器和客户端都需提供）

```
public class PWCBHandler implements CallbackHandler {  
  
    public void handle(Callback[] callbacks) throws  
        IOException, UnsupportedCallbackException {  
  
        for (int i = 0; i < callbacks.length; i++) {  
  
            //When the server side need to authenticate  
            the user  
  
            WSPasswordCallback pwcb =  
                (WSPasswordCallback) callbacks[i];  
  
            if (pwcb.getUsage() ==  
                WSPasswordCallback.USERNAME_TOKEN_UNKNOWN) {  
  
                if (pwcb.getIdentifier().equals("bob")  
                    && pwcb.getPassword().equals("bobPW")) {  
  
                    //If authentication successful,  
                    simply return  
  
                    return;  
  
                } else {  
  
                    throw new  
                        UnsupportedCallbackException(callbacks[i], "check  
                        failed");  
                }  
            }  
        }  
    }  
}
```

```
//When the client requests for the password  
to be added in to the  
//UT element  
pwcb.setPassword("bobPW");  
}  
}
```

2、客户端 axis2.xml 配置

```
<axisconfig name="AxisJava2.0">  
.....  
<module ref="rampart" />  
  
<parameter name="OutflowSecurity">  
    <action>  
        <items>UsernameToken</items>  
        <user>bob</user>  
        <passwordCallbackClass>  
            org.apache.rampart.samples.sample03.PWCBHandler  
        </passwordCallbackClass>  
        <passwordType>PasswordText  
    </passwordType>  
    </action>  
    </parameter>  
.....  
</axisconfig>
```

3、服务器端 services.xml 配置

```
<service>
```

```
<operation name="echo">

    <messageReceiver
        class="org.apache.axis2.rpc.receivers.RPCMessageRec
        eiver"/>

    </operation>

    <parameter name="ServiceClass" locked="false">
        org.apache.rampart.samples.sample02.SimpleService
    </parameter>

    <!--对服务使用 rampart，如果在 axis2.xml 中注册了 rampart
        模块，这里就不再需要再配置-->

    <parameter name="InflowSecurity">

        <action>

            <items>

                <!--访问该服务需要身份验证-->

                UsernameToken

            </items>

            <passwordCallbackClass>

                <!--口令提供类路径-->

                org.apache.rampart.samples.sample03.PWCBHandler

            </passwordCallbackClass>

        </action>

    </parameter>

</service>
```

5.3.2 消息数字签名

使用签名，SOAP 消息的接收方可以知道已签名的元素在路由中未发生改变。首先我们用密钥和证书管理工具 keytool 生成 x.509 公钥和私钥证书。对 JKS 文件，service.jks 和 client.jks 说明如下：

service.jks 存放了 service 的私钥和 client 的公钥，service.jks 的私钥和公钥 keystore 的密码是 apache。

client.jks 存放了 client 的私钥和 service 的公钥，client.jks 的私钥和公钥 keystore 的密码是 apache。

1、口令（服务器端和客户端都需要部署）

```
public class PWCBHandler implements CallbackHandler {  
    public void handle(Callback[] callbacks) throws  
    IOException, UnsupportedCallbackException {  
        for (int i = 0; i < callbacks.length; i++) {  
            WSPasswordCallback pwcb =  
                (WSPasswordCallback) callbacks[i];  
  
            String id = pwcb.getIdentifier();  
            if("client".equals(id)) {  
                pwcb.setPassword("apache");  
            } else if("service".equals(id)) {  
                pwcb.setPassword("apache");  
            }  
        }  
    }  
}
```

2、客户端 axis2.xml 配置

```
<axisconfig name="AxisJava2.0">

    .....

    <module ref="rampart" />

    <parameter name="OutflowSecurity"><!--发送消息安全
配置-->

        <action>

            <items> Signature</items>

            <user>client</user>

            <signaturePropFile>client.properties</signature
PropFile>

            <passwordCallbackClass>

                org.apache.rampart.samples.sample04.PWCBHandler

            </passwordCallbackClass>

        </action>

    </parameter>

    <parameter name="InflowSecurity"><!--接收消息安全配
置-->

        <action>

            <items> Signature</items>

            <signaturePropFile>client.properties</signature
PropFile>

        </action>

    </parameter>
```

```
PropFile>
    </action>
    </parameter>
    .....
</axisconfig>
```

client.properties 文件属性设置

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.com-
ponents.crypto.Merlin
org.apache.ws.security.crypto.merlin.keystore.type=jks
org.apache.ws.security.crypto.merlin.keystore.password=apacheorg.
apache.ws.security.crypto.merlin.file=client.jks
```

3、服务器端 services.xml 配置

```
<service>
    .....
    <parameter name="InflowSecurity"><!--请求消息安全配置-->
        <action>
            <items> Signature</items><!--数字签名验证-->
            <signaturePropFile>service.properties</signaturePropFile><!--数
字签名属性文件-->
        </action>
    </parameter>

    <parameter name="OutflowSecurity"><!--响应消息安全配置-->
        <action>
            <items> Signature</items>
            <user>service</user>
        </action>
    </parameter>
```

```
<passwordCallbackClass>  
    org.apache.rampart.samples.sample04.PWCBHandler  
</passwordCallbackClass>  
  
<signaturePropFile>service.properties</signaturePropFile><!--数  
字签名属性文件-->  
  
<signatureKeyIdentifier>DirectReference</signatureKeyIdentifie  
r>  
  
</action>  
  
</parameter>  
  
.....  
</service>
```

service.properties 文件中配置的属性

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.com  
ponents.crypto.Merlin  
org.apache.ws.security.crypto.merlin.keystore.type=jks  
org.apache.ws.security.crypto.merlin.keystore.password=apache  
org.apache.ws.security.crypto.merlin.file=service.jks
```

5.3.3 消息加密配置

配置对消息的加密和解密

1、口令验证类（服务器端和客户端都需提供）

```
public class PWCBHandler implements CallbackHandler  
{
```

```
public void handle(Callback[] callbacks) throws
    IOException, UnsupportedCallbackException {
    for (int i = 0; i < callbacks.length; i++) {
        WSPasswordCallback pwcb =
            (WSPasswordCallback) callbacks[i];
        String id = pwcb.getIdentifer();
        if("client".equals(id)) {
            pwcb.setPassword("apache");
        } else if("service".equals(id)) {
            pwcb.setPassword("apache");
        }
    }
}
```

2、客户端 axis2.xml 配置

```
<axisconfig name="AxisJava2.0">
    .....
    <module ref="rampart" />

    <parameter name="OutflowSecurity">
        <action>
            <items>Encrypt</items><!--消息加密-->
            <encryptionUser>service</encryptionUser><!--加
密者-->
            <encryptionPropFile>client.properties</encryp
```

```
tionPropFile><!--加密属性文件-->

    </action>

    </parameter>

<parameter name="InflowSecurity">

    <action>

        <items>Encrypt</items>

        <passwordCallbackClass>

            org.apache.rampart.samples.sample05.PWCBHandler

        </passwordCallbackClass>

    </action>

    <decryptionPropFile>client.properties</decryptionPropFile>

    </action>

    </parameter>

    .....

</axisconfig>
```

client.properties 文件属性设置：

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.components.crypto.Merlin
org.apache.ws.security.crypto.merlin.keystore.type=jks
org.apache.ws.security.crypto.merlin.keystore.password=apache
org.apache.ws.security.crypto.merlin.file=client.jks
```

3、服务器端 service.xml 中配置

```
<service>

    .....

    <parameter name="InflowSecurity">

        <action>

            <items>Encrypt</items><!-对请求消息解密-->

            <passwordCallbackClass>

                org.apache.rampart.samples.sample05.PWCBHandler

            </passwordCallbackClass>




            <decryptionPropFile>service.properties</decryptionPropFile>

        </action>

    </parameter>




    <parameter name="OutflowSecurity">

        <action>

            <items>Encrypt</items><!-对响应消息加密-->

            <encryptionUser>client</encryptionUser>




            <encryptionPropFile>service.properties</encryptionPropFile>

        </action>

    </parameter>

    .....

</service>
```

service.properties 文件中配置的属性：

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.com  
ponents.crypto.Merlin
```

```
org.apache.ws.security.crypto.merlin.keystore.type=jks
org.apache.ws.security.crypto.merlin.keystore.password=apache
org.apache.ws.security.crypto.merlin.file=service.jks
```

5.3.4 组合使用配置

可以配置对消息同时使用时间戳、数字签名和加密。下面的配置为对 client 和 server 端的请求消息先使用时间戳、数字签名，再对消息进行发送。对响应消息先接解密，再验证签名和时间戳。

5.3.4.1 先使用时间戳、数字签名，再对消息进行发送

1、口令验证类

```
public class PWCBHandler implements CallbackHandler {
    public void handle(Callback[] callbacks) throws
    IOException, UnsupportedCallbackException {
        for (int i = 0; i < callbacks.length; i++) {
            WSPasswordCallback pwcb =
(WSPasswordCallback) callbacks[i];
            String id = pwcb.getIdentifier();
            if("client".equals(id)) {
                pwcb.setPassword("apache");
            } else if("service".equals(id)) {
                pwcb.setPassword("apache");
            }
        }
    }
}
```

```
}
```

```
}
```

2、客户端 axis2.xml 配置

```
<axisconfig name="AxisJava2.0">

    .....

    <module ref="rampart" />

<parameter name="OutflowSecurity">

    <action>

        <items>Timestamp Signature Encrypt</items>

        <user>client</user>

        <passwordCallbackClass>

            org.apache.rampart.samples.sample06.PWCBHandler

        </passwordCallbackClass>

    </action>

<signaturePropFile>client.properties</signatureProp
File>

<signatureKeyIdentifier>DirectReference</signatureK
eyIdentifier>

<encryptionKeyIdentifier>SKIKeyIdentifier</encrypti
onKeyIdentifier>

        <encryptionUser>service</encryptionUser>

    </action>

</parameter>
```

```
<parameter name="InflowSecurity">  
    <action>  
        <items>Timestamp Signature Encrypt</items>  
        <passwordCallbackClass>  
            org.apache.rampart.samples.sample06.PWCBHandler  
        </passwordCallbackClass>  
  
        <signaturePropFile>client.properties</signatureProp  
        File>  
    </action>  
</parameter>  
.....  
</axisconfig>
```

client.properties 文件属性设置：

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.  
components.crypto.Merlin  
org.apache.ws.security.crypto.merlin.keystore.type=jks  
org.apache.ws.security.crypto.merlin.keystore.password=apache  
org.apache.ws.security.crypto.merlin.file=client.jks
```

3、服务器端 services.xml 配置

```
<service>  
    .....  
    <parameter name="InflowSecurity">  
        <action>
```

```
<items>Timestamp Signature
Encrypt</items>

<passwordCallbackClass>
    org.apache.rampart.samples.sample06.PWCBHandler
</passwordCallbackClass>

<signaturePropFile>service.properties</signaturePropFile>
</action>
</parameter>

<parameter name="OutflowSecurity">
    <action>
        <items>Timestamp Signature
Encrypt</items>
        <user>service</user>
        <passwordCallbackClass>
            org.apache.rampart.samples.sample06.PWCBHandler
        </passwordCallbackClass>
        <signaturePropFile>service.properties</signaturePropFile>
        <signatureKeyIdentifier>DirectReference</sign
```

```
atureKeyIdentifier>

<encryptionKeyIdentifier>SKIKeyIdentifier</en
cryptionKeyIdentifier>

<encryptionUser>useReqSigCert</encryptionUser
>

</action>

</parameter>

.....
</service>
```

service.properties 文件中配置的属性:

```
org.apache.ws.security.crypto.provider=org.apache.ws.security.
components.crypto.Merlin
org.apache.ws.security.crypto.merlin.keystore.type=jks
org.apache.ws.security.crypto.merlin.keystore.password=apache
org.apache.ws.security.crypto.merlin.file=service.jks
```

5.3.4.2 先加密再签名

组合使用可以位置互换，例如，上面使用的是先签名在加密，也可以先加密再签名，配置如下。

客户端 axis2.xml 配置

```
.....
<items>Timestamp Encrypt Signature</items>
.....
```

服务器端 services.xml 配置

```
.....  
<items>Timestamp Encrypt Signature</items>  
.....
```

5.3.4.3 消息签名两次

客户端 axis2.xml 配置

```
.....  
  
<parameter name="OutflowSecurity">  
  
    <action>  
  
        <items>Timestamp Signature NoSerialization</items>  
  
        <user>client</user>  
  
        <passwordCallbackClass>  
  
            org.apache.rampart.samples.sample08.PWCBHandler  
  
        </passwordCallbackClass>  
  
  
<signatureKeyIdentifier>DirectReference</signatureKeyIdentifie  
r>  
  
  
<signatureParts>{Element} {http://docs.oasis-open.org/wss/2004/  
01/oasis-200401-wss-wssecurity-utility-1.0.xsd}Timestamp</sign  
atureParts>  
  
  
<signaturePropFile>client.properties</signaturePropFile>  
  
    </action>  
  
  
    <action>  
  
        <items>Signature</items>
```

```
<user>client</user>

<passwordCallbackClass>
    org.apache.rampart.samples.sample08.PWCBHandler
</passwordCallbackClass>

<signaturePropFile>client.properties</signaturePropFile>

</action>

</parameter>

....
```

服务器端 services.xml 配置

```
.....
<items>Timestamp Signature Signature</items>
.....
```

5.3.5 发送消息时动态指定安全策略

客户端发送消息时，动态指定安全策略，而不是在 axis2.xml 中配置。

```
public class Client {

    public static void main(String[] args) throws
Exception {

    if(args.length != 2) {

        System.out.println("Usage: $java Client
endpoint_address client_repo_path");
    }
}
```

```
ConfigurationContext ctx =
ConfigurationContextFactory.createConfigurationContextFromFileSystem(args[1], args[1] +
"/conf/axis2.xml");

ServiceClient client = new ServiceClient(ctx,
null);

Options options = new Options();
options.setAction("urn:echo");
options.setTo(new
EndpointReference(args[0]));

//Set the rampart parameters

options.setProperty(WSSHandlerConstants.OUTFLOW_SECURITY,
getOutflowConfiguration());

options.setProperty(WSSHandlerConstants.INFLOW_SECURITY,
getInflowConfiguration());

client.setOptions(options);

//Engage rampart

client.engageModule(new QName("rampart"));

OMElement response =
client.sendReceive(getPayload("Hello world"));
```

```
        System.out.println(response);

    }

    private static OMElement getPayload(String value)
    {
        OMFactory factory =
        OMAbstractFactory.getOMFactory();

        OMNamespace ns =
        factory.createOMNamespace("http://sample11.samples.
rampart.apache.org/xsd", "ns1");

        OMElement elem =
        factory.createOMELEMENT("echo", ns);

        OMElement childElem =
        factory.createOMELEMENT("param0", null);

        childElem.setText(value);

        elem.addChild(childElem);

        return elem;
    }

    private static Parameter
getOutflowConfiguration() {
    OutflowConfiguration ofc = new
    OutflowConfiguration();

    ofc.setActionItems("Timestamp Signature
Encrypt");

    ofc.setUser("client");
}
```

```
ofc.setPasswordCallbackClass("org.apache.rampart.sa
mples.sample11.PWCBHandler");

ofc.setSignaturePropFile("client.properties");

ofc.setSignatureKeyIdentifier(WSSHandlerConstants.B
ST_DIRECT_REFERENCE);

ofc.setEncryptionKeyIdentifier(WSSHandlerConstants.
ISSUER_SERIAL);

ofc.setEncryptionUser("service");

return ofc.getProperty();

}

private static Parameter getInflowConfiguration()
{
    InflowConfiguration ifc = new
InflowConfiguration();

    ifc.setActionItems("Timestamp Signature
Encrypt");

    ifc.setPasswordCallbackClass("org.apache.rampart.sa
mples.sample11.PWCBHandler");

    ifc.setSignaturePropFile("client.properties");
}
```

```
    return ifc.getProperty();  
  
}  
  
}
```

5.3.6 使用策略描述文件

发布服务时，将策略描述文件加到 `services.xml` 中，客户端获取策略描述内容保存到本地，发送消息时加载这个策略文件。

`policy.xml` 内容

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<!--  
  
!  
  
! Copyright 2006 The Apache Software Foundation.  
  
!  
  
! Licensed under the Apache License, Version 2.0 (the "License");  
!  
! you may not use this file except in compliance with the License.  
  
! You may obtain a copy of the License at  
  
!  
  
!     http://www.apache.org/licenses/LICENSE-2.0  
  
!  
  
! Unless required by applicable law or agreed to in writing, software  
!  
! distributed under the License is distributed on an "AS IS" BASIS,  
!  
! WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or  
implied.  
  
! See the License for the specific language governing permissions  
and  
  
! limitations under the License.
```

```
!-->

<wsp:Policy wsu:Id="UTOverTransport"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-ws-
  s-wssecurity-utility-1.0.xsd"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">

  <wsp:ExactlyOne>
    <wsp:All>
      <sp:TransportBinding
        xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">

        <wsp:Policy>
          <sp:TransportToken>
            <wsp:Policy>
              <sp:HttpsToken RequireClientCertificate="false"/>
            </wsp:Policy>
          </sp:TransportToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:Basic256/>
            </wsp:Policy>
          </sp:AlgorithmSuite>
          <sp:Layout>
            <wsp:Policy>
              <sp:Lax/>
            </wsp:Policy>
          </sp:Layout>
          <sp:IncludeTimestamp/>
        </wsp:Policy>
      </sp:TransportBinding>
```

```
<sp:SignedSupportingTokens
  xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
  <wsp:Policy>
    <sp:UsernameToken
      sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient" />
  </wsp:Policy>
</sp:SignedSupportingTokens>

<ramp:RampartConfig
  xmlns:ramp="http://ws.apache.org/rampart/policy">
  <ramp:user>alice</ramp:user>
  <ramp:passwordCallbackClass>org.apache.rampart.samples.policy.sample01.PWCBHandler</ramp:passwordCallbackClass>
</ramp:RampartConfig>

</wsp:All>
</wsp:ExactlyOne>
</wsp:Policy>
```

PWCBHandler.java

```
public class PWCBHandler implements CallbackHandler {
  public void handle(Callback[] callbacks) throws
    IOException,
    UnsupportedCallbackException {
    for (int i = 0; i < callbacks.length; i++) {
```

```
//When the server side need to authenticate the
user

WSPasswordCallback pwcb =
(WSPasswordCallback) callbacks[i];

if (pwcb.getUsage() ==
WSPasswordCallback.USERNAME_TOKEN_UNKNOWN) {

    if(pwcb.getIdentifer().equals("alice")
&& pwcb.getPassword().equals("bobPW")) {

        return;

    } else {

        throw new
UnsupportedCallbackException(callbacks[i], "check
failed");

    }

}

//When the client requests for the password to
be added in to the

//UT element

pwcb.setPassword("bobPW");

}

}

}
```

1) 服务器段 services.xml 中增加安全策略描述

```
<?xml version="1.0" encoding="UTF-8"?>

<!--

!

! Copyright 2006 The Apache Software Foundation.

!

! Licensed under the Apache License, Version 2.0 (the "License");
! you may not use this file except in compliance with the License.
! You may obtain a copy of the License at

!

!     http://www.apache.org/licenses/LICENSE-2.0

!

! Unless required by applicable law or agreed to in writing, software
! distributed under the License is distributed on an "AS IS" BASIS,
! WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.

! See the License for the specific language governing permissions
and

! limitations under the License.

!-->

<!-- services.xml of sample-1 : UsernameToken-->

<service>

    <operation name="echo">

        <messageReceiver
class="org.apache.axis2.rpc.receivers.RPCMessageReceiver"/>

    </operation>

    <parameter name="ServiceClass"
```

```
locked="false">org.apache.rampart.samples.policy.sample01.SimpleService</parameter>

<module ref="rampart" />
<module ref="addressing" />

<wsp:Policy wsu:Id="UTOverTransport"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">

  <wsp:ExactlyOne>
    <wsp:All>
      <sp:TransportBinding
        xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">
        <wsp:Policy>
          <sp:TransportToken>
            <wsp:Policy>
              <sp:HttpsToken RequireClientCertificate="false"/>
            </wsp:Policy>
          </sp:TransportToken>
          <sp:AlgorithmSuite>
            <wsp:Policy>
              <sp:Basic256/>
            </wsp:Policy>
          </sp:AlgorithmSuite>
          <sp:Layout>
            <wsp:Policy>
              <sp:Lax/>
            </wsp:Policy>
          </sp:Layout>
        </wsp:Policy>
      </sp:TransportBinding>
    </wsp:All>
  </wsp:ExactlyOne>
</wsp:Policy>
```

```
</wsp:Policy>

</sp:Layout>

<sp:IncludeTimestamp/>

</wsp:Policy>

</sp:TransportBinding>

<sp:SignedSupportingTokens

xmlns:sp="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy">

<wsp:Policy>

<sp:UsernameToken

sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypo
licy/IncludeToken/AlwaysToRecipient" />

</wsp:Policy>

</sp:SignedSupportingTokens>

<ramp:RampartConfig

xmlns:ramp="http://ws.apache.org/rampart/policy">

<ramp:passwordCallbackClass>org.apache.rampart.samples.policy.
sample01.PWCBHandler</ramp:passwordCallbackClass>

</ramp:RampartConfig>

</wsp:All>

</wsp:ExactlyOne>

</wsp:Policy>

</service>
```

2) 客户端发送

客户端首先应该获取到访问服务所需的 policy.xml 文件，然后在发送消息时设置

```
public class Client {  
  
    public static void main(String[] args) throws  
        Exception {  
  
        if(args.length != 3) {  
  
            System.out.println("Usage: $java Client  
                endpoint_address client_repo_path policy_xml_path");  
        }  
  
        ConfigurationContext ctx =  
        ConfigurationContextFactory.createConfigurationCont  
        extFromFileSystem(args[1], null);  
  
        ServiceClient client = new ServiceClient(ctx,  
        null);  
  
        Options options = new Options();  
        options.setAction("urn:echo");  
        options.setTo(new  
        EndpointReference(args[0]));  
  
        options.setProperty(RampartMessageData.KEY_RAMPART_  
        POLICY, loadPolicy(args[2]));  
        client.setOptions(options);  
    }  
}
```

```
        client.engageModule(new QName("addressing"));

        client.engageModule(new QName("rampart"));

        OMElement response =
client.sendReceive(getPayload("Hello world"));

        System.out.println(response);

    }

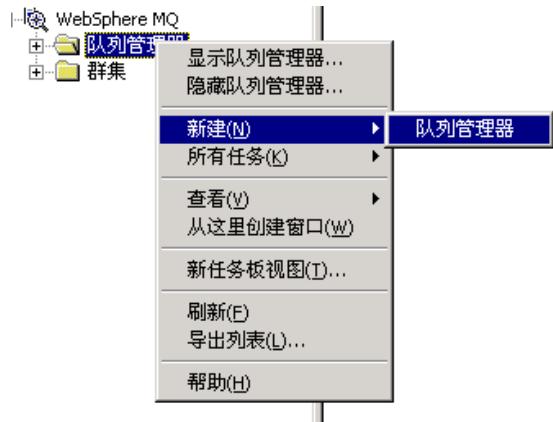
private static Policy loadPolicy(String xmlPath)
throws Exception {
    StAXOMBuilder builder = new
StAXOMBuilder(xmlPath);
    return
PolicyEngine.getPolicy(builder.getDocumentElement());
}

private static OMElement getPayload(String value)
{
    .....
}
```

附录二 通过 MQ (V5.3) 资源管理器使用

6.1. 队列管理器的创建

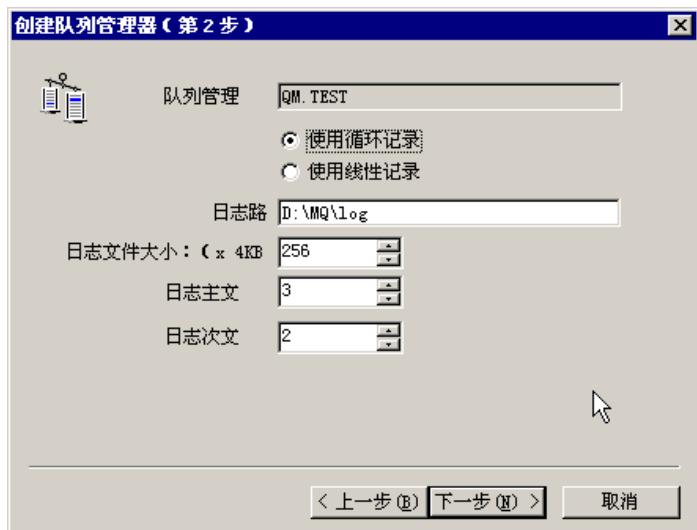
打开 WebSphere MQ 资源管理器，右键点击“队列管理器”，在菜单中选择“新建”->“队列管理器”



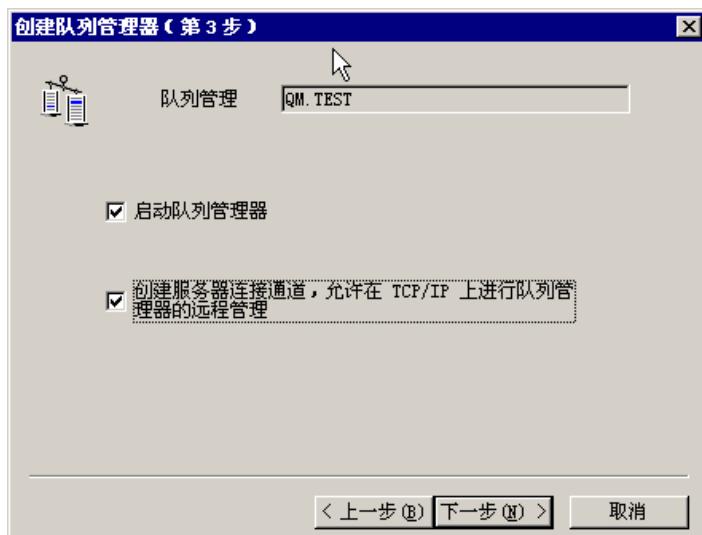
弹出下面的窗口



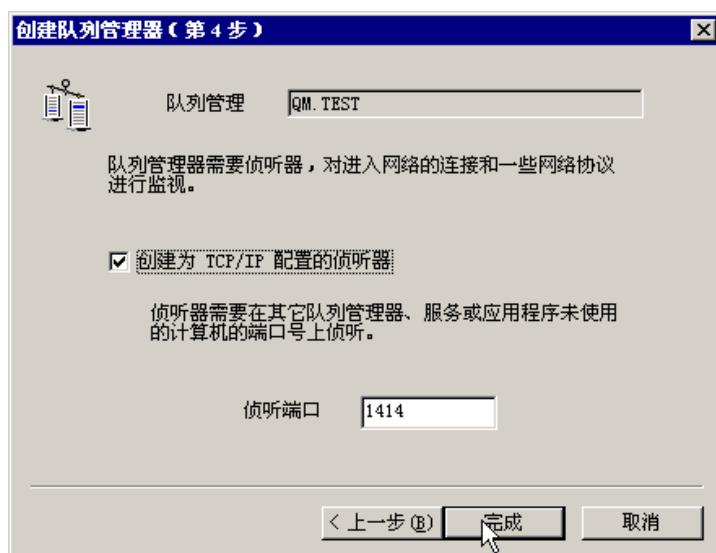
在“队列管理”填写队列管理器的名称，点击“下一步”



点击“下一步”



点击“下一步”：

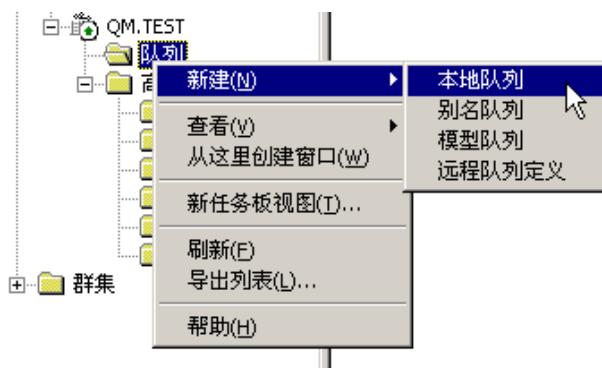


“侦听端口”默认为 1414，可根据实际需要做修改，然后点击“完成”。队列管理器创建完成。

6.1.1.队列的创建

6.1.1.1.本地队列的创建

打开要配置的队列管理器，右键点击“队列”，在弹出的菜单中选择“新建”→“本地队列”



弹出队列创建窗口：



在“队列名称”中填写本地队列的名称，“确定”完成。

6.1.1.2. 传输队列的创建

重复创建本地队列的步骤



注意“用法”选“传输”。传输队列要和远程队列成对出现的，一个队列管理器有几个远程队列就有几个传输队列。

6.1.1.3. 远程队列的创建

一个远程队列对应另外一个节点 MQ 上的本地队列，本地节点上要创建对应每个与其链接的节点上本地队列的远程队列。

右键点击“队列”，在弹出的菜单中选择“新建”→“远程队列定义”





“远程队列名称”是链接远程节点上 MQ 上本地队列的名称。

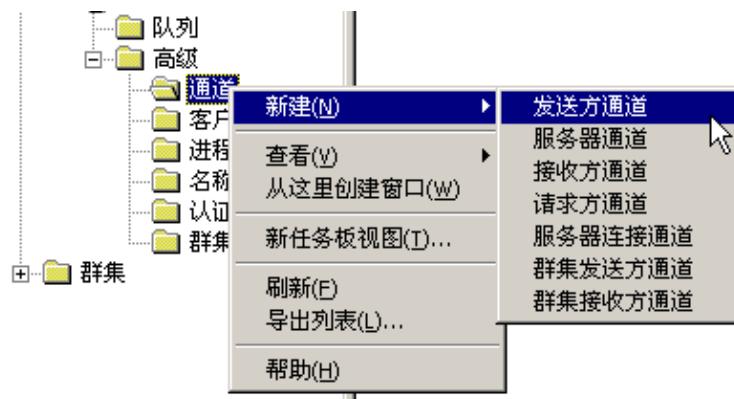
“远程队列管理器名称”是链接远程节点上 MQ 队列管理器的名称。

“传输队列名称”选择没有被其他远程队列使用的传输队列，一个传输队列只能与一个远程队列对应。

6.1.2.通道的创建

6.1.2.1.发送方通道

通过“队列管理器->高级->通道”，右键点击“通道”，选择“新建->”发送方通道”



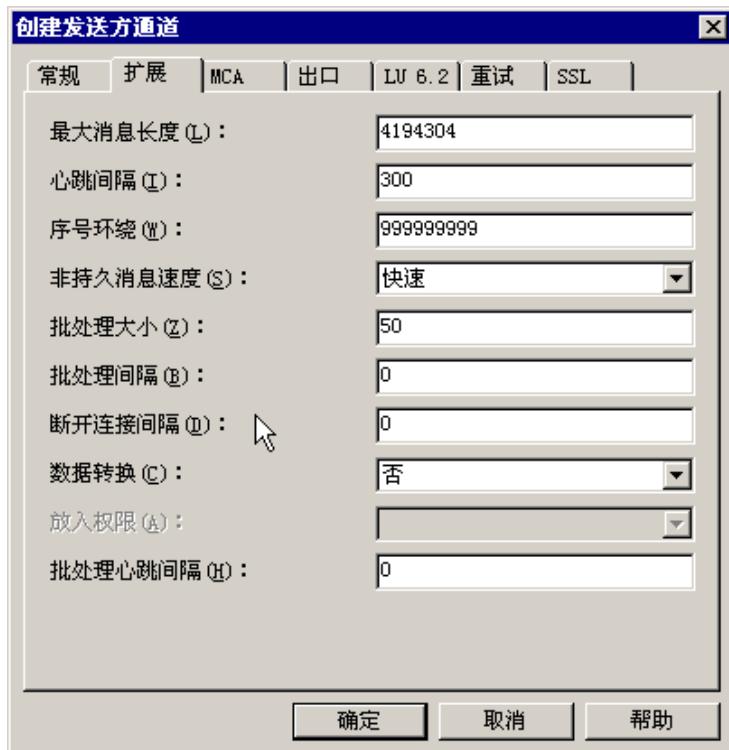
弹出发送方通道的新建窗口



“传输协议”选默认设置。

“连接名称”的格式为 IP(侦听端口)”, 这里的 IP 和侦听端口都是指对方(远程节点)的。例如要连接 IP 为 192.9.100.13 的节点, 该节点的侦听端口为 1414, “连接名称”为 192.9.100.13(1414)。“传输队列”要选择和连接远程节点的远程队列一样的传输队列, 即连接远程的节点的远程队列和发送方通道要使用同一传输队列。

点击上面的“扩展”



将“断开连接间隔”改为 0,为确定”完成。

6.1.2.2. 接收方通道的创建

通过”队列管理器->高级->通道”，右键点击“通道”，选择“新建”->”接收方通道”



接收方通道只需填写“通道名称”即可，名称要和与其连接的远程节点上的发送方通道名称相同。

6.1.2.3.通道的启动

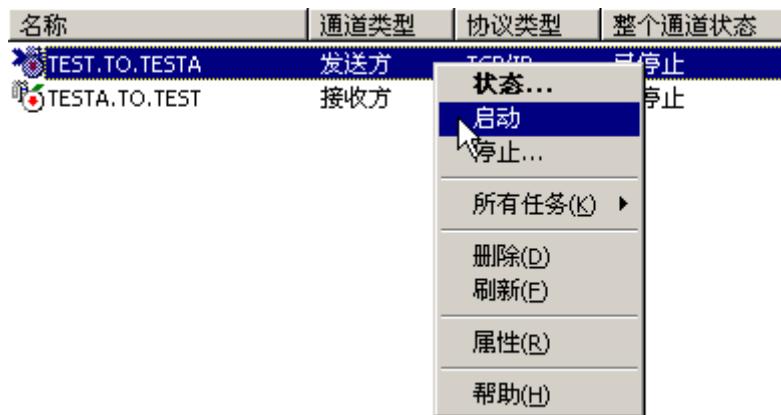
通过“队列管理器->高级->列通道”，点击“通道”，



右侧窗口显示已经创建的通道

名称	通道类型	协议类型	整个通道状态
TEST.TO.TESTA	发送方	TCP/IP	已停止
TESTA.TO.TEST	接收方	TCP/IP	已停止

若“整个通道状态”显示不是“正在运行”就要启动通道，右键点击要启动的通道

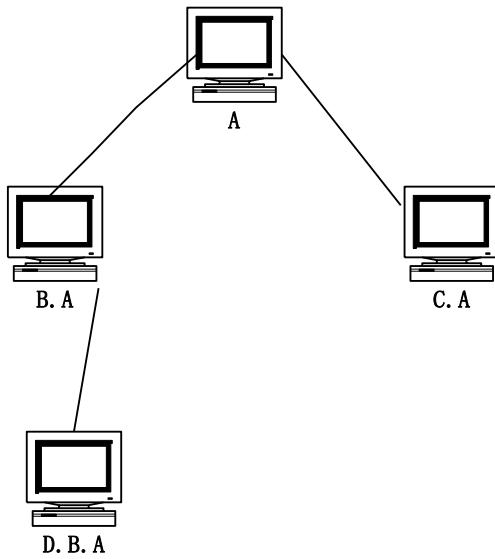


点击菜单中的“启动”。通道的启动要在两个节点之间进行，先启动 A 节点的发送方通道 A.TO.B，然后在 B 节点上启动与 A 节点上已经启动的发送方通道同名的接收方通道 A.TO.B。

名称	通道类型	协议类型	整个通道状态
TESTA.TO.TESTA	发送方	TCP/IP	正在运行
TESTA.TO.TEST	接收方	TCP/IP	正在运行

启动成功后的通道状态显示为“正在运行”。

附录三 MQ 传输配置示例



节点树有 4 个节点，A 有子节点：B.A 和 C.A，B.A 有子节点 D.B.A。

7.1. 节点信息

节点名称	IP	MQ 侦听端口
A	192.9.100.1	1414
B.A	192.9.100.2	1414
C.A	192.9.100.3	1414
D.B.A	192.9.100.4	1414

7.2. 节点 A 的配置:

7.2.1. MQ

- 队列管理器: QM.A
- 队列
 - ◆ 本地队列: APP.LOCAL.INBOX
 - ◆ 传输队列: APP.TRAN.B, APP.TRAN.C
 - ◆ 远程队列:
 - ◊ APP.REMOTE.B, 对应 B.A 的本地队列

配置信息:

远程队列名称: APP.LOCAL.INBOX(B.A 上 MQ 的本地队列)

远程队列管理器名称: QM.B

传输队列: APP.TRAN.B

◊ APP.REMOTE.C, 对应 C.A 的本地队列

配置信息:

远程队列名称: APP.LOCAL.INBOX(C.A 上 MQ 的本地队列)

远程队列管理器名称: QM.C

传输队列: APP.TRAN.C

- 通道

- ◆ 发送方通道:

◊ A.TO.B, 与 B.A 链接的通道

配置信息:

连接名称: 192.9.100.2(1414)

传输队列: APP.TRAN.B

本地通讯地址: 192.9.100.1

- ◆ A.TO.C,与 C.A 链接的通道

配置信息:

连接名称: 192.9.100.3(1414)

传输队列: APP.TRAN.C

本地通讯地址: 192.9.100.1

◆ 接收方通道:

- ◆ B.TO.A, 与 B.A 链接的通道
- ◆ C.TO.A, 与 C.A 链接的通道

7.3. 节点 B.A 的配置:

7.3.1.MQ

■ 队列管理器: QM.B**■ 队列****◆ 本地队列: APP.LOCAL.INBOX****◆ 传输队列: APP.TRAN.A, APP.TRAN.D****◆ 远程队列:**

- ◆ APP.REMOTE.A, 对应 A 的本地队列

配置信息:

远程队列名称: APP.LOCAL.INBOX(A 上 MQ 的本地队列)

远程队列管理器名称: QM.A

传输队列: APP.TRAN.A

- ◆ APP.REMOTE.D, 对应 D.B.A 的本地队列

配置信息:

远程队列名称: APP.LOCAL.INBOX(D.B.A 上 MQ 的本地队列)

远程队列管理器名称: QM.D

传输队列: APP.TRAN.D

■ 通道

◆ 发送方通道:

◇ B.TO.A,与 A 链接的通道

配置信息:

连接名称: 192.9.100.1(1414)

传输队列: APP.TRAN.A

本地通讯地址: 192.9.100.2

◇ B.TO.D,与 D.B.A 链接的通道

配置信息:

连接名称: 192.9.100.4(1414)

传输队列: APP.TRAN.D

本地通讯地址: 192.9.100.2

◆ 接收方通道:

◇ A.TO.B, 与 A 链接的通道

◇ D.TO.B, 与 D.B.A 链接的通道

7.3.2.WebSphere

■ 队列连接工厂: dc_factory

配置信息:

JNDI: jms/dc_factory

QUEUE MANAGER: QM.B

■ JMS 队列目的地

◆ 接收消息的队列目的地: b_inbox

配置信息:

JNDI 名: jms/b_inbox

基本队列名: APP.LOCAL.INBOX

基本 Queue Manager 名: QM.B

- ◆ 发送消息的队列目的地

向 B.A 发送消息的队列目的地: a_inbox

配置信息:

JNDI 名: jms/a_inbox

基本队列名: APP.REMOTE.A

基本 Queue Manager 名: QM.B

- ◇ 向 D.B.A 发送消息的队列目的地: d_inbox

配置信息:

JNDI 名: jms/d_inbox

基本队列名: APP.REMOTE.D

基本 Queue Manager 名: QM.B

- 倾听器端口: JmsListenerPort

配置信息:

连接工厂 JNDI 名: jms/dc_factory

目的地 JNDI 名: jms/b_inbox

7.4. 节点 C.A 的配置:

7.4.1. MQ

- 队列管理器: QM.C

- 队列

- ◆ 本地队列: APP.LOCAL.INBOX

- ◆ 传输队列: APP.TRAN.A

- ◆ 远程队列:

- ◇ APP.REMOTE.A, 对应 A 的本地队列

配置信息:

远程队列名称: APP.LOCAL.INBOX(A 上 MQ 的本地队列)

远程队列管理器名称: QM.A

传输队列: APP.TRAN.A

■ 通道

◆ 发送方通道:

◇ C.TO.A, 与 A 链接的通道

配置信息:

连接名称: 192.9.100.1(1414)

传输队列: APP.TRAN.A

本地通讯地址: 192.9.100.3

◆ 接收方通道:

◇ A.TO.C, 与 A 链接的通道

7.5. 节点 D.B.A 的配置:

7.5.1. MQ

■ 队列管理器: QM.D

■ 队列

◆ 本地队列: APP.LOCAL.INBOX

◆ 传输队列: APP.TRAN.B

◆ 远程队列:

◇ APP.REMOTE.B, 对应 B.A 的本地队列

配置信息:

远程队列名称: APP.LOCAL.INBOX(B.A 上 MQ 的本地队列)

远程队列管理器名称: QM.B

传输队列: APP.TRAN.B

■ 通道

◆ 发送方通道:

◆ D.TO.B,与 B.A 链接的通道

配置信息:

连接名称: 192.9.100.2(1414)

传输队列: APP.TRAN.B

本地通讯地址: 192.9.100.4

◆ 接收方通道:

◆ B.TO.D, 与 B.A 链接的通道

附录四 TongLINK/Q7.1 传输配置示例

8.1 TongLINK /Q7.1 使用概述

本附录简述 TongLINK /Q7.1 (以下简称 TLQ) 在配置各个节点之间传输实例，主要简述使用 TLQ 传输的相关内容，涉及具体技术细节请参看 TLQ 使用手册。

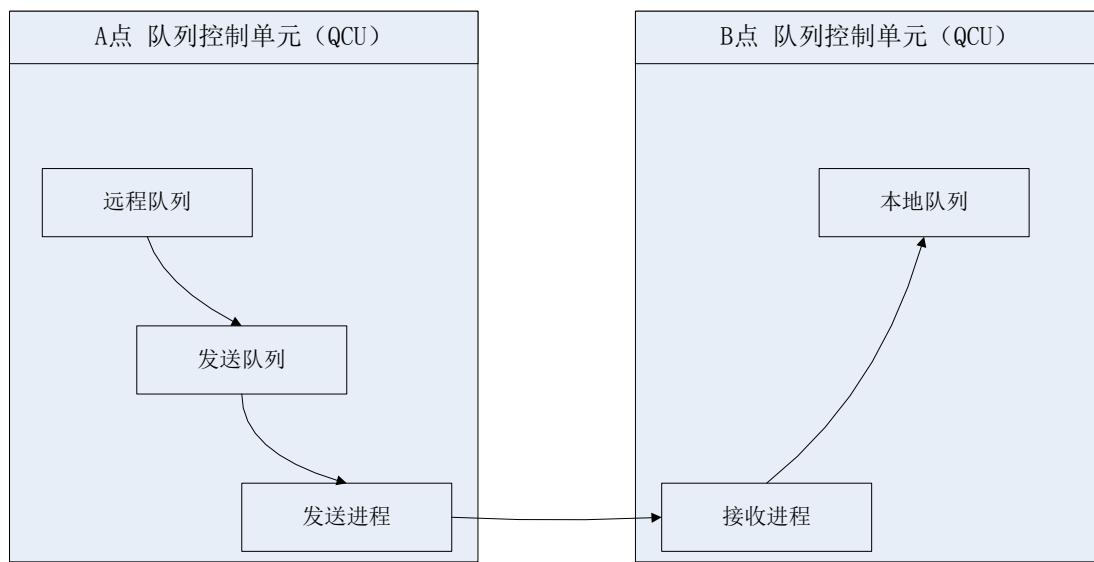
8.1.1 TLQ 连接模式

连接 TLQ 服务器有 2 种模式，分别为绑定 TongLINKQ 服务器模式以及客户端连接模式，前者需要 EI 服务器（或者 DEAgent 应用）和 TLQ 服务器在同一物理主机上，主机上安装 TLQ 服务器软件即可；后者 EI 服务器（或者 DEAgent 应用）不需要和 TLQ 服务器在同一台主机上，不过 EI 服务器（或者 DEAgent 应用）所在的主机上必须安装 TLQ 客户端软件。

8.1.2 TLQ 消息传输简述

如下图所示，消息从 A 点 TLQ 传输到 B 点 TLQ，需要在 A 点 QCU，也就是队列控制单元中，配置 1 条远程队列，此远程队列由 1 条发送队列以及对方（B 点）TLQ 的本地队列构成，而这个发送队列和本地的 1 个发送进程相关联，同时本地

的一个发送进程和远程目的地（B 点）TLQ 的接收进程相关联。



8.2 TLQ 传输配置示例

以 A 点和 B 点为例，一个为发送消息，另外一个为接收消息，假设：

A 点 IP 为：192.9.100.1

B 点 IP 为：192.9.100.2

以下以 A 点到 B 点消息传输配置为例，简单说明如何配置 TLQ 以达到传输消息的目的，示例仅为 A 到 B 点的单向传输，如果多点双向传输，在此基础上增加配置即可。

8.2.1 A 点（发送端）etc 配置文件

假定队列控制单元名称为 qcu1，那么配置文件为 tlqcu_qcu1.conf，在此文件中配置队列以及发送接收进程即可，下面只列出主要的配置项，其他项默认即可（注：etc 配置文件位置在 TLQ 安装目录下，比如 Tongtech\TLQ71\etc）

(1) 远程队列

```
#####
[RemoteQue]      # 远程队列单元小节
#
```

```
[RemoteQueRecord]      #
RemoteQueName = rq_a2b      # 远程队列名
SendQueName = sq_a2b        # 对应发送队列名
DestQueName = lq_b_1        # 目的队列名
#
#####
#####
```

(2) 发送队列

```
#####
#[SendQue]      # 发送队列单元小节
#
#[SendQueRecord]      #
SendQueName = sq_a2b        # 发送队列名称
MsgNum = 100      # 队列存放最大消息数
MsgSize = 1048576      # 单个消息最大长度, 单位 byte 注意需要设置大一些
SendQDataBuff = 10000      # 发送队列数据存储区记录数
DestQCUName = getqcu        # 目的队列控制单元名
DefDestQueName =          # 缺省目的队列名
#
#####
#####
```

(3) 发送进程

```
#####
#[SendProcess]      # 发送进程小节
#
#[SendProcessRecord]      #
SendProcID = 1      # 发送进程唯一标识
SendProcStatus = 1      # 发送进程状态, 0 禁用, 1 正常
#####
#####
```

```
#  
  
[SendConnRecord]      # 发送连接小节  
  
ConnName = conn       # 连接名称  
  
SendQueName = sq_a2b    # 连接对应的发送队列名称  
  
ConnStatus = 1         # 发送连接状态, 0 禁用, 1 正常  
  
HostName = 192.9.100.2    # 被连接节点的 IP 地址  
  
ConnPort = 10004        # 被连接节点的端口号  
  
ConnType = 0            # 连接类型, 0 为常连接, 1 为按需连接  
  
LineType = LINE10M      # 线路类型  
  
DiscInterval = 60        # 线路维持时间  
  
BeatInterval = 20        # 线路检测时间(只适用于常连接)  
  
SecExitFlag =           # 出口标志 Net|Transport|ByRecv  
  
SecExitNetData =         # 网络层出口用户数据  
  
SecExitTransportData =   # 传输层出口用户数据  
  
#  
  
#####
```

(4) 瘦客户代理, TLQ 客户端远程访问 TLQ 服务器时候, 需要配置此项, 默认端口为 10261, 不用修改即可

```
#####  
  
[ClientBroker]          # 瘦客户代理小节  
  
#  
  
[ClientBrokerRecord]     # 瘦客户代理  
  
CliBrokerID = 2          # 代理进程编号  
  
CliBrokerStatus = 1        # 启动节点时代理的启动状态, 0:不启动, 1:启动  
  
ListenPort = 10261        # 代理监听的端口号  
  
HisRecMaxNum = 10         # 历史文件中历史记录的最大数(支持断点续传)  
  
#
```

8.2.2 B 点（接收端）etc 配置文件

(1) 接收进程

```
#####
[RcvProcess]      # 接收进程小节
#
[RcvProcessRecord]      #
RcvProcID = 1      # 接收进程编号
RcvProcStatus = 1      # 接收进程状态, 0 禁用, 1 正常
ListenPort = 10004      # 监听端口 发送端连接需要指定这个
LocalAddr =      # 本地通信地址,如果有此项, 则接收进程绑定
#
#####
```

(2) 本地队列

```
#####
#
[LocalQueRecord]      #
LocalQueName = 1q_b_1      # 本地队列名 被发送端远程队列指定
MsgNum = 100      # 队列存放最大消息数
MsgSize = 1048576      # 单个消息最大长度, 单位 byte   设置大一些
LocalQueDataBuff = 100      # 本地队列数据存储区记录数
MsgArrangeMode = 0      # 消息组织模式, 0 为先进先出, 1 为优先级
UsageType = 0      # 队列的使用类型, 0: 普通本地队列, 1: 发布订阅队列
#
#####
```

(3) 瘦客户代理，TLQ 客户端远程访问 TLQ 服务器时候，需要配置此项，默认端口为 10261，不用修改即可

```
#####
[ClientBroker]          # 瘦客户代理小节
#
[ClientBrokerRecord]      # 瘦客户代理
CliBrokerID = 2           # 代理进程编号
CliBrokerStatus = 1        # 启动节点时代理的启动状态,0:不启动,1:启动
ListenPort = 10261         # 代理监听的端口号
HisRecMaxNum = 10          # 历史文件中历史记录的最大数(支持断点续传)
#
#
```

8.2.3 启动 TLQ 以及调试诊断

配置好 TLQ 后，启动 TLQ 服务器即可（注：启动程序在 TLQ 安装程序的 Tongtech\TLQ71\start.bat 中），启动如果正常，会提示如图信息：

```
Contract Number -1
Product Name      =[TongLINK/Q]
Max QCUs          =[100]
Max Applications=[100]
Max Thin Clients=[100]
Max Connections To Send =[100]
Max Connections To Recv =[100]
      [Version Information]:
Product Version =[7.1.0.1]
Product Name      =[TongLINK/Q]
TLQ Create IPC resource OK!
Starting now, maybe take a few minutes....
TLQ Start process tl_moni OK!
-----
QCUI[qcui1] start OK!
TLQ start OK!

d:\Program Files\Tongtech\TLQ71>pause
请按任意键继续... ■
```

如果启动失败，可能是某些配置项配置错误导致，可以参考 TLQ 的日志，位置为 TLQ 安装目录下 Tongtech\TLQ71\log\TongLink.sys 中，具体错误排查请参考 TLQ 使用手册，这里不再赘述。

8.2.4 测试 TLQ 物理链路是否畅通

配置好 TLQ，并且启动正常，最好还需要验证 2 点之间的物理链路是否畅通，可以利用 TLQ 命令行输入命令进行测试，也可以使用代码进行发送、接收数据进行测试，我们推荐使用第二种方法，因为第一种方法中 TLQ 不同版本之间，命令或者方法可能不同。

使用代码发送接收消息，需要在机器装有 TLQ 客户端，这样可以远程访问 TLQ 服务器，测试也就不再需要在各个物理主机上运行代码。下面基于 A 和 B 节点之间的配置，展示代码示例：

(1) 向远程队列或者本地队列发送消息（标红为需要注意的地方）

```
public static void main(String[] args) {
    sendMsg();
}

private static void sendMsg() {
    System.out.println(">>in sendMsg");
    TlqConnection tlqConnection = null;
    TlqQCU tlqQcu = null;
    try {
        TlqConnContext context = new TlqConnContext();
        context.BrokerId=-1;
        context.HostName="192.9.100.1";
        context.ListenPort=10261;//这里的端口为TongLink/Q的客户端代理，在TLQ服务器的QCU配置文件的‘瘦客户代理小节’中配置
        System.out.println("new Connection");
        tlqConnection = new TlqConnection(context);
        System.out.println("end get connection");
    } catch (Exception e1) {
        e1.printStackTrace();
    }
    try {
        System.out.println("begin open qcu");
        tlqQcu = tlqConnection.openQCU(myQcuName); //打开队列管理器
        System.out.println("end open qcu");
    } catch (TlqException e1) {
        e1.printStackTrace();
    }
    TlqMessage message = new TlqMessage();
```

```
TlqMsgOpt opt = new TlqMsgOpt();
TlqMessage msgInfo = new TlqMessage();
TlqMsgOpt msgOpt = new TlqMsgOpt(); //msg操作选项类
msgInfo.MsgType = TlqMessage.BUF_MSG; //TlqMessage.BUF_MSG;
String msgStr = "TESTMSG!!"; //消息内容
msgInfo.MsgSize = msgStr.getBytes().length; //消息大小
byte[] msgContent = msgStr.getBytes();
msgInfo.setMsgData(msgContent);
msgInfo.Persistence = TlqMessage.TLQPER_Y; //持久性
msgInfo.Priority = TlqMessage.TLQPRI_NORMAL; //优先级
msgOpt.QueName = "rq_a2b"; //队列名
msgOpt.RemoveFileFlag = TlqMsgOpt.NOTREMOVEFILE;
try {
    System.out.println("put msg");
    tlqQcu.putMessage(msgInfo, msgOpt); //发送消息、使用队列管理单元发送消息
    System.out.println("put msg ok : "+msgStr);
} catch (TlqException e) {
    e.printStackTrace();
} finally{
    try {
        tlqQcu.close();
    } catch (TlqException e) {
        e.printStackTrace();
    }
    try {
        tlqConnection.close();
    } catch (TlqException e) {
        e.printStackTrace();
    }
}
}
```

(2) 从远程队列或者本地队列获取消息 (标红为需要注意的地方)

```
public static void main(String[] args) {
    revMsg();
}

private static void revMsg() throws Exception{
    int MyMsgCount = 0;
    int myWaitInterval=0;
```

```
TlqConnection tlqConnection = null;
TlqQCU tlqQcu = null;

int msgCount = 0;
try {
    TlqConnContext context =new TlqConnContext();
    context.BrokerId=-1;
    context.HostName="192.9.100.2";
    context.ListenPort=10261;
    System.out.println("new Connection");
    tlqConnection = new TlqConnection(context);
    System.out.println("end get connection");
    tlqQcu = tlqConnection.openQCU(myQcuName);
    while (true) {
        TlqMessage msgInfo = new TlqMessage();
        TlqMsgOpt msgOpt = new TlqMsgOpt();
        msgOpt.QueName = "1q_b_1";//本地队列
        msgOpt.WaitInterval = myWaitInterval;
        msgOpt.OperateType = TlqMsgOpt.TLQOT_GET;
        tlqQcu.txBegin();
        tlqQcu.getMessage(msgInfo, msgOpt);
        msgCount = msgCount + 1;
        if (msgInfo.GroupID.length() > 0) { //是否组消息
            System.out.println(
                "-----This is a Group Message,GroupId is " +
                msgInfo.GroupID);
            int child = 1;
            while (msgInfo.GroupStatus !=
TlqMessage.TLQGROUP_LAST) { //是否组消息的最后一条
                System.out.println("-----GroupSeq is " +
                    msgInfo.GroupSeq);
                tlqQcu.getMessage(msgInfo, msgOpt);
                child++;
                msgCount++;
            }
            System.out.println("-----GroupSeq is " +
msgInfo.GroupSeq);
            System.out.println("-----receive group msg
over!----- ");
        } else {
            int _ii = printMsgInfo(msgInfo);
            if(_ii== -1) {
```

```
        tlqQcu.txRollback();
    } else{
        tlqQcu.txCommit();
    }
}

if (msgOpt.AckMode == TlqMsgOpt.TLQACK_USER) {
    int acktype = TlqMsgOpt.TLQACK_COMMIT;
    tlqQcu.ackMessage(msgInfo, msgOpt, acktype);
}

}

} catch (TlqException e) {
    try {
        tlqQcu.close();
        tlqConnection.close();
    } catch (TlqException e1) {
    }
} finally {
    MyMsgCount = msgCount;
System.out.println("=====msgCount=====:"+msgCount);
    try {
        tlqQcu.close();
        tlqConnection.close();
    } catch (TlqException e) {
        e.printStackTrace();
    }
}
}

}

private static int printMsgInfo(TlqMessage msgInfo) {
    if ((int) msgInfo.MsgType == 1) {
        System.out.println("Received a File Msg");
        System.out.print("msgInfo.MsgId=" + new
String(msgInfo.MsgId));
        System.out.println("msgInfo.MsgSize=" + (int)
msgInfo.MsgSize);
        System.out.println("msgInfo:" + new
String(msgInfo.getMsgData()));
    } else {
        System.out.println("Received a Buffer Msg");
        System.out.print("msgInfo.MsgId=" + new
String(msgInfo.MsgId));
    }
}
```

```
System.out.println(" msgInfo.MsgSize=" + (int)
msgInfo.MsgSize);
System.out.println("msgInfo:" +new
String(msgInfo.getMsgData()));
String path = "d:\\temp"+File.separator+"tlqfile.txt";
File file = new File(path);
if(!file.exists()){
try {
file.createNewFile();
} catch (IOException e) {
e.printStackTrace();
}
}
FileOutputStream os = null;
try {
os= new FileOutputStream(file);
os.write(msgInfo.getMsgData());

} catch (FileNotFoundException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}finally{
try {
os.close();
} catch (IOException e) {
e.printStackTrace();
}
}
System.out.println("ok");
}
return 1;
}
```

附录五 数据源配置

RCloud 云 EI 服 务 数 据 源 配 置 文 件 在
\RCloudHome\DataExchange\RODEConfig.xml 中设置：

```
<?xml version="1.0" encoding="UTF-8"?>

<DEConfig>

    <DBConfig>

        <ManagerType>2</ManagerType> <!-- 数据库连接管理类型: Container:1,DE-I:2
-->

        <Parameter> <!-- ManagerType=2 时有效 -->

            <Mode>2</Mode> <!-- 1:数据库独立运行方式,2:内嵌 Derby 方式 , 3: 内嵌
derby 方式且完全在内存中运行 DB -->

            <User>rode</User>

            <Password>rode</Password>

            <Driver>org.apache.derby.jdbc.EmbeddedDriver</Driver>           <!--
org.apache.derby.jdbc.ClientDriver -->

            <ConnectUrl></ConnectUrl>      <!--      Mode=2      时      有      效
jdbc:derby:net://ip:port/dbname -->

            <MaxPoolSize>15</MaxPoolSize>

            <MinPoolSize>3</MinPoolSize>

            <MaxIdleTime>1800</MaxIdleTime><!-- 单位: 秒 -->

        </Parameter>

    </DBConfig>

</DEConfig>
```

容器管理数据源

在应用服务器中配置数据源（带事务控制），由应用服务器负责数据库连接的创建和释放，（例如 weblogic, websphere 等）。

第一步：在应用服务器中配置数据源

第二步：rodeconfig.xml 中数据库配置如下

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<DEConfig>
    <DBConfig>
        <ManagerType>1</ManagerType> <!-- 数据库连接管理类型: Container:1,DE-I:2
-->

        <Parameter> <!-- ManagerType=2 时有效 -->
            <Mode></Mode> <!-- 1:数据库独立运行方式,2:内嵌 Derby 方式 , 3: 内嵌
derby 方式且完全在内存中运行 DB -->
            <User> </User>
            <Password> </Password>
            <Driver> </Driver> <!-- org.apache.derby.jdbc.ClientDriver -->
            <ConnectUrl></ConnectUrl>      <!-- Mode=2      时      有      效
jdbc:derby:net://ip:port/dbname -->
            <MaxPoolSize></MaxPoolSize>
            <MinPoolSize></MinPoolSize>
            <MaxIdleTime></MaxIdleTime><!-- 单位: 秒 -->
        </Parameter>
    </DBConfig>
</DEConfig>
```

RCloud 云 EI 服务管理数据源（使用外部数据库）

这种模式下数据库连接由 RCloud 云 EI 服务服务器控制，数据库使用外部数据库，不需要在应用服务器里面配置数据源。

rodeconfig.xml 中数据库配置如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<DEConfig>
    <DBConfig>
        <ManagerType>2</ManagerType> <!-- 数据库连接管理类型: Container:1,DE-I:2
-->
```

```
-->

<Parameter> <!-- ManagerType=2 时有效 -->

<Mode>1</Mode> <!-- 1:数据库独立运行方式,2:内嵌 Derby 方式 , 3: 内嵌 derby 方式且完全在内存中运行 DB -->

<User>rode</User>

<Password>rode</Password>

<Driver>org.apache.derby.jdbc.EmbeddedDriver</Driver>           <!--
org.apache.derby.jdbc.ClientDriver -->

<ConnectUrl>jdbc:derby:net://【ip】:【port】/【dbname】</ConnectUrl> <!--
Mode=2 时有效 jdbc:derby:net://ip:port/dbname -->

<MaxPoolSize>15</MaxPoolSize>

<MinPoolSize>3</MinPoolSize>

<MaxIdleTime>1800</MaxIdleTime><!-- 单位: 秒 -->

</Parameter>

</DBConfig>

</DEConfig>
```

RCloud 云 EI 服务管理数据源（使用内嵌数据库）

这种模式下数据库连接由 RCloud 云 EI 服务服务器控制，数据库使用内嵌数据库，不需要在应用服务器里面配置数据源。

rodeconfig.xml 中数据库配置如下：

```
<?xml version="1.0" encoding="UTF-8"?>

<DEConfig>

<DBConfig>

<ManagerType>2</ManagerType> <!-- 数据库连接管理类型: Container:1,DE-I:2
-->

<Parameter> <!-- ManagerType=2 时有效 -->
```

```
<Mode>2</Mode> <!-- 1:数据库独立运行方式,2:内嵌 Derby 方式 , 3: 内嵌  
derby 方式且完全在内存中运行 DB -->  
  
<User>rode</User>  
  
<Password>rode</Password>  
  
<Driver>org.apache.derby.jdbc.EmbeddedDriver</Driver> <!--  
org.apache.derby.jdbc.ClientDriver -->  
  
<ConnectUrl>      </ConnectUrl>      <!--      Mode=2      时      有      效  
jdbc:derby:net://ip:port/dbname -->  
  
<MaxPoolSize>15</MaxPoolSize>  
  
<MinPoolSize>3</MinPoolSize>  
  
<MaxIdleTime>1800</MaxIdleTime><!-- 单位: 秒 -->  
  
</Parameter>  
  
</DBConfig>  
  
</DEConfig>
```