

睿智ETL(UDIS-ETL)使用

北京睿智欣泰科技有限公司

www.udis.com.cn

主要内容

- 睿智ETL组成
- 睿智ETL管理界面总布局
- 数据库配置
- 变量配置
- 规则表达式配置
- 主要的一些ETL服务
 - SQL服务
 - 文件-表的转换服务
 - 表复制
 - 表-表转换服务
 - 表并发路由服务
 - 事实表处理服务
 - 前端报表软件触发
 - Shell脚本调用
 - 定时服务
 - Ftp服务
- 流程
- 应用工程项目创建
- 系统运行监控
- 二次开发
 - 如何引入特殊格式文件数据？
 - 如何扩展特殊数据转换/清洗处理？
 - 如何增加转换变量？
 - 如何增加服务？
 - 如何OEM？

睿智ETL组成

主要由睿智ETL管理工具、元数据服务器、ETL服务器三部分组成

The image displays the '睿智ETL管理工具' (Wise ETL Management Tool) interface. The top part shows a workflow diagram with components like 'Task_1', 'Task_2', 'Task_3', 'TableWriter', 'TableReader', and 'Table'. Below the diagram is a log window titled '睿智ETL管理工具' showing the execution results of a test job. The log includes the following entries:

```
2006-11-25 15:17:31 [Service/ExtractSingleFileTableService/0] Single FileTable Wq Service Begin!
2006-11-25 15:17:31 [Service/ExtractTableFullService/0] process begin.
2006-11-25 15:17:36 [Service/ExtractSingleFileTableService/0] finished: 14 lines of sql: excel(<XC336
2006-11-25 15:17:36 [Service/ExtractSingleFileTableService/0] finished: 14 lines of sql: excel(<XC336223
2006-11-25 15:17:36 [Service/ExtractSingleFileTableService/0] Single FileTable Wq Service finished!
2006-11-25 15:17:36 [Service/TableMapService/tablemap/0] Trans begin!
2006-11-25 15:17:36 [Service/TableMapService/tablemap/0] Extract and Translated: 3 lines
2006-11-25 15:17:36 [Service/TableMapService/tablemap/0] Trans finish!
2006-11-25 15:17:36 [Service/TableMonitorService/tablemonitor/0] TableMonitor begin!
2006-11-25 15:17:36 [Service/TableMonitorService/tablemonitor/0] Extract: 3 lines
2006-11-25 15:17:36 [Service/TableMonitorService/tablemonitor/0] TableMonitor finish!
2006-11-25 15:17:36 [Service/TableMonitorService/tablemonitor/0] Target Table: userprintinfidivl_wqrole_source Trans and Insert: 1 Lines
2006-11-25 15:17:36 [Service/TableMonitorService/tablemonitor/0] Target Table: userprintinfidivl_wqrole_target Trans and Insert: 1 Lines
```

The right side of the image shows a command prompt window displaying the Tomcat startup logs, including the following entries:

```
2006-11-25 15:53:35 org.apache.catalina.startup.Catalina load
信息: Initialization processed in 703 ms
2006-11-25 15:53:35 org.apache.catalina.core.StandardService start
信息: Starting service Catalina
2006-11-25 15:53:35 org.apache.catalina.core.StandardEngine start
信息: Starting Servlet Engine: Apache Tomcat/5.5.17
2006-11-25 15:53:35 org.apache.catalina.core.StandardHost start
信息: XML validation disabled
2006-11-25 15:53:35 org.apache.catalina.startup.HostConfig deployWAR
信息: Deploying web application archive etlmanager.war
ManagerProxyService start success
2006-11-25 15:53:36 org.apache.coyote.http11.Http11BaseProtocol start
信息: Starting Coyote HTTP/1.1 on http-7001
2006-11-25 15:53:37 org.apache.jk.common.ChannelSocket init
信息: JK: ajp13 listening on /0.0.0.0:8009
2006-11-25 15:53:37 org.apache.jk.server.JkMain start
信息: Jk running ID=0 time=0/15 config=null
2006-11-25 15:53:37 org.apache.catalina.storeconfig.StoreLoader load
信息: Find registry server-registry.xml at classpath resource
2006-11-25 15:53:37 org.apache.catalina.startup.Catalina start
信息: Server startup in 1937 ms
ManagerServlet (2006-11-25 15:54:21): initializing...
ManagerServlet (2006-11-25 15:54:21): ProcessType = Service.ServiceType Requ
type = Query
className = unb.manager.server.Service.ServiceTypeProcess
ServiceType...
ManagerServlet (2006-11-25 15:54:21): ProcessType = Process.ProcessType Requ
type = Query
className = unb.manager.server.Process.ProcessTypeProcess
ProcessType...
ManagerServlet (2006-11-25 15:54:21): DBResource Requ
type = Query
className = unb.man
ManagerServlet (2006-11-25 15:54:21): ServiceType = FTPClientService Process
= Service.Service RequestType = Query
className = unb.manager.server.Service.ServiceProcess
ManagerServlet (2006-11-25 15:54:22): ServiceType = FTPServerService Process
= Service.Service RequestType = Query
className = unb.manager.server.Service.ServiceProcess
ManagerServlet (2006-11-25 15:54:22): ServiceType = SocketClientService Proc
type = Service.Service RequestType = Query
className = unb.manager.server.Service.ServiceProcess
ManagerServlet (2006-11-25 15:54:22): ServiceType = SocketServerService Proc
type = Service.Service RequestType = Query
className = unb.manager.server.Service.ServiceProcess
ManagerServlet (2006-11-25 15:54:22): ServiceType = TimerService ProcessType
type = Query
className = unb.manager.server.TimerService.TimerServiceProcess
TimerService init ServiceID: LoadData1
LoadDataService init ServiceID: LoadData1 End
LoadDataService init end!
TimerServices init begin
TimerServices init ServiceID: ETLTimeService
2006-11-25 15:53:26 org.quartz.simpl.SimpleThreadPool initialize
信息: Job execution threads will use class loader of thread: main
2006-11-25 15:53:27 org.quartz.simpl.RAMJobStore initialize
信息: RAMJobStore initialized.
2006-11-25 15:53:27 org.quartz.impl.StdSchedulerFactory instantiate
信息: Quartz scheduler 'DefaultQuartzScheduler' initialized from default resourc
e file in Quartz package: 'quartz.properties'
2006-11-25 15:53:27 org.quartz.impl.StdSchedulerFactory instantiate
信息: Quartz scheduler version: 1.4.5
```

睿智ETL管理界面总布局

睿智数据集成管理

系统 资源 基础服务 业务服务 流程 高级 帮助

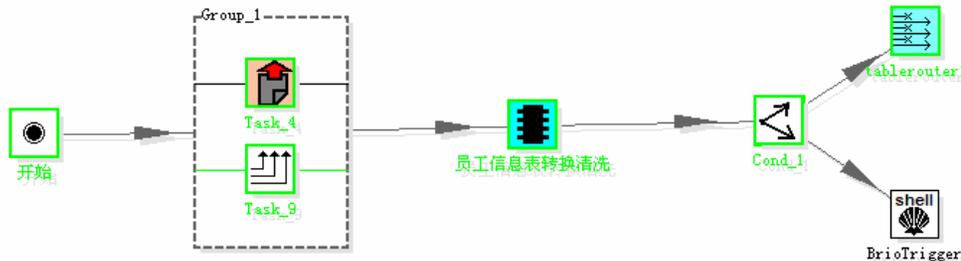
基础配置

- 数据库连接
- 变量
- 规则
- 服务
 - 基础服务
 - 业务服务
 - 动态SQL服务1
 - 动态SQL函数服务
 - 文件-表转换服务
 - testservice
 - 表复制(增量)
 - 表复制2
 - 甘肃省全量抽取
 - bbb
 - 表-表转换服务
 - 员工信息表转换清洗
 - 表并发路由
 - tablerouter1
 - 事实表处理
 - 申报业务事实表加工
 - 数据抽取到文本文件
 - 从文本文件装载数据
 - 数据抽取到DBF文件
 - 从DBF文件装载数据
 - 基于内存的表复制
 - 变化数据复制
 - Brio触发服务
 - BrioTrigger

- 流程
- ETLProcess
- 应用1流程
 - ExtractToTargetETLProcess
- 应用配置
- 应用项目例子1
- 测试
 - 测试例子1
 - 表映射例子
 - 路由由例子
 - 文件映射例子
- 运行监控
- 正在运行
- 成功完成
 - 测试/测试例子1/顺序执行流程例子
 - 测试/测试例子1/意外处理例子
 - 测试/测试例子1/并行流程例子
- 停止的
- 意外的

运行监控 查看流程 查看表转换处理 查看表路由处理 编辑流程 查看流程 查看流程

测试/测试例子1/并行流程例子 流程运行监控图(绿色表示运行成功,黄色表示正在运行,红色表示被终止)



菜单测试/测试例子1/并行流程例子的运行结果:
开始时间: 2006-11-22 11:35:48

```
2006-11-22 11:35:48 {Service/ExtractSingleFileToTable/testservice/0} Single File-Table Map Service Begin!
2006-11-22 11:35:48 {Service/ExtractToTargetFullService/bbb/0} process begin
2006-11-22 11:35:48 {Service/ExtractToTargetFullService/bbb/0} process Table:CSBGTTT@local_oracle_db
2006-11-22 11:35:48 {Service/ExtractSingleFileToTable/testservice/0} finished: 3 lines of map: csv<->csvtest
2006-11-22 11:35:49 {Service/MemoryCopyTableService/MemoryCopyTableService/0} Extract and Insert: 1 lines
2006-11-22 11:35:49 {Service/ExtractToTargetFullService/bbb/0} finished process Table: CSBGTTT@local_oracle_db
2006-11-22 11:35:49 {Service/ExtractToTargetFullService/bbb/0} process end
2006-11-22 11:35:49 {Service/ExtractSingleFileToTable/testservice/0} finished: 14 lines of map: excel<->CSBG
2006-11-22 11:35:49 {Service/ExtractSingleFileToTable/testservice/0} finished: 14 lines of map: excel<->CSBGBBB
2006-11-22 11:35:49 {Service/ExtractSingleFileToTable/testservice/0} Single File-Table Map Service finished!
2006-11-22 11:35:49 {Service/TableMapService/tablemap1/0} Trans begin!
2006-11-22 11:35:49 {Service/TableMapService/tablemap1/0} Extract and Translated: 3 lines
2006-11-22 11:35:49 {Service/TableMapService/tablemap1/0} Trans finished!
2006-11-22 11:35:49 {Service/TableRouterService/tablerouter1/0} TableRouter begin!
2006-11-22 11:35:49 {Service/TableRouterService/tablerouter1/0} Extract : 3 lines
2006-11-22 11:35:49 {Service/TableRouterService/tablerouter1/0} TableRouter finished!
2006-11-22 11:35:49 {Service/TableRouterService/tablerouter1/0} Target Table: userprintinfo@local_oracle_source Trans and Insert: 1 Lines
2006-11-22 11:35:49 {Service/TableRouterService/tablerouter1/0} Target Table: userprintinfo@local_oracle_target Trans and Insert: 1 Lines
```

数据库配置(1)

选择数据库的类型和版本号

输入库名、用户名/密码等

输入连接池最大连接数

选择数据库字符集

The screenshot shows a dialog box titled "数据库连接修改" (Database Connection Modification). It contains several input fields and dropdown menus. Annotations with arrows point from text boxes on the left to specific fields in the dialog:

- "选择数据库的类型和版本号" points to the "类型" (Type) dropdown menu, which is set to "DBCconnect", and the "版本号" (Version) text field, which contains "9.2".
- "输入库名、用户名/密码等" points to the "URL" text field, which contains "jdbc:oracle:thin:@localhost:1521:etldb", and the "用户名" (Username) text field, which contains "etl".
- "输入连接池最大连接数" points to the "连接最大数" (Maximum Connections) text field, which contains "10".
- "选择数据库字符集" points to the "字符集" (Character Set) dropdown menu, which is set to "GBK".

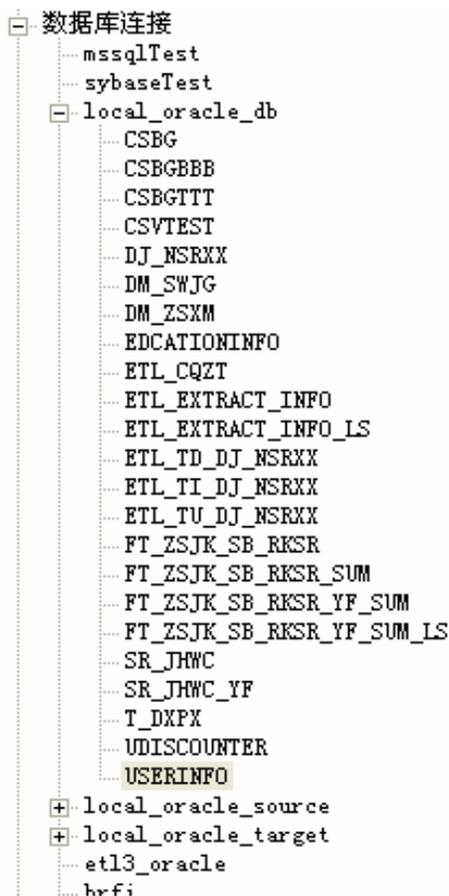
Other fields in the dialog include:

- "数据库标识" (Database Identifier): local_oracle_db
- "驱动类型" (Driver Type): Oracle
- "密码" (Password): ***
- "工作区用户?" (Work Area User?): false
- "工作区用户" (Work Area User): etl

At the bottom of the dialog are two buttons: "保存" (Save) and "退出" (Exit).

数据库配置(2)

完成数据库配置后，能看到该库下的表列表、相应表的结构信息



A screenshot of a table structure window titled 'USERINFO@local_oracle_db'. The window displays a table with the following columns: 字段名 (Field Name), 类型 (Type), 是否主键? (Is Primary Key?), 允许为空? (Allow Null?), and 大小 (Size). The table contains the following data:

字段名	类型	是否主键?	允许为空?	大小
USERID	VARCHAR2	false	NOT NULL	50
NAME	VARCHAR2	false	NOT NULL	50
SEX	CHAR	false	NOT NULL	2
JIGUAN	VARCHAR2	false	NOT NULL	50
BIRTHDDAY	DATE	false	NOT NULL	7
PROVINCE	VARCHAR2	false	NOT NULL	50
CITY	VARCHAR2	false	NOT NULL	50
DIQU	VARCHAR2	false	NOT NULL	50
ADDR	VARCHAR2	false	NOT NULL	150
EMAIL	VARCHAR2	false	NOT NULL	150
TEL	VARCHAR2	false	NOT NULL	50

At the bottom of the window, there is a button labeled '关闭' (Close).

变量配置(1)

基础变量：用于定义变量类的组成及各组成成员的类型等信息。

主要用于描述流程全局输入变量，给组合变量和转换变量提供输入。

基础变量配置窗口截图：

- 变量标识：ExtractDataProcessInput
- 变量类型：DefaultInput
- 变量描述：base var
- 成员变量列表：
 - 成员变量1 (选中)
 - 成员变量2
 - 成员变量3
 - 成员变量4
 - 成员变量5
 - 成员变量6
 - 成员变量7
 - 成员变量8
 - 成员变量9
- 参数序号：1
- 参数类型：java.lang.String
- 参数名称：exportFileName
- 默认值：aaa
- 操作按钮：添加、修改、删除、保存、退出

变量配置(2)

组合变量：主要根据已有的变量组合形成新的变量，包含变量的组成、各组成成员的类型、来源等信息。主要给流程处理节点的提供输入。

其成员来源可以是基础变量的类型、上一个或上几个流程节点的输出变量类型、转换变量类型、组合变量类型等。

组合变量

变量标识: SQLTransInput 变量类型: VarInput

变量描述: String

成员变量1
成员变量2
成员变量3
成员变量4

添加

修改

删除

参数序号: 1

参数类型: java.lang.String

参数名称: sDB

默认值: String

值是从其它变量中来?

来源的变量类型: BaseVar

来源的变量ID: ExtractDataProcessInput

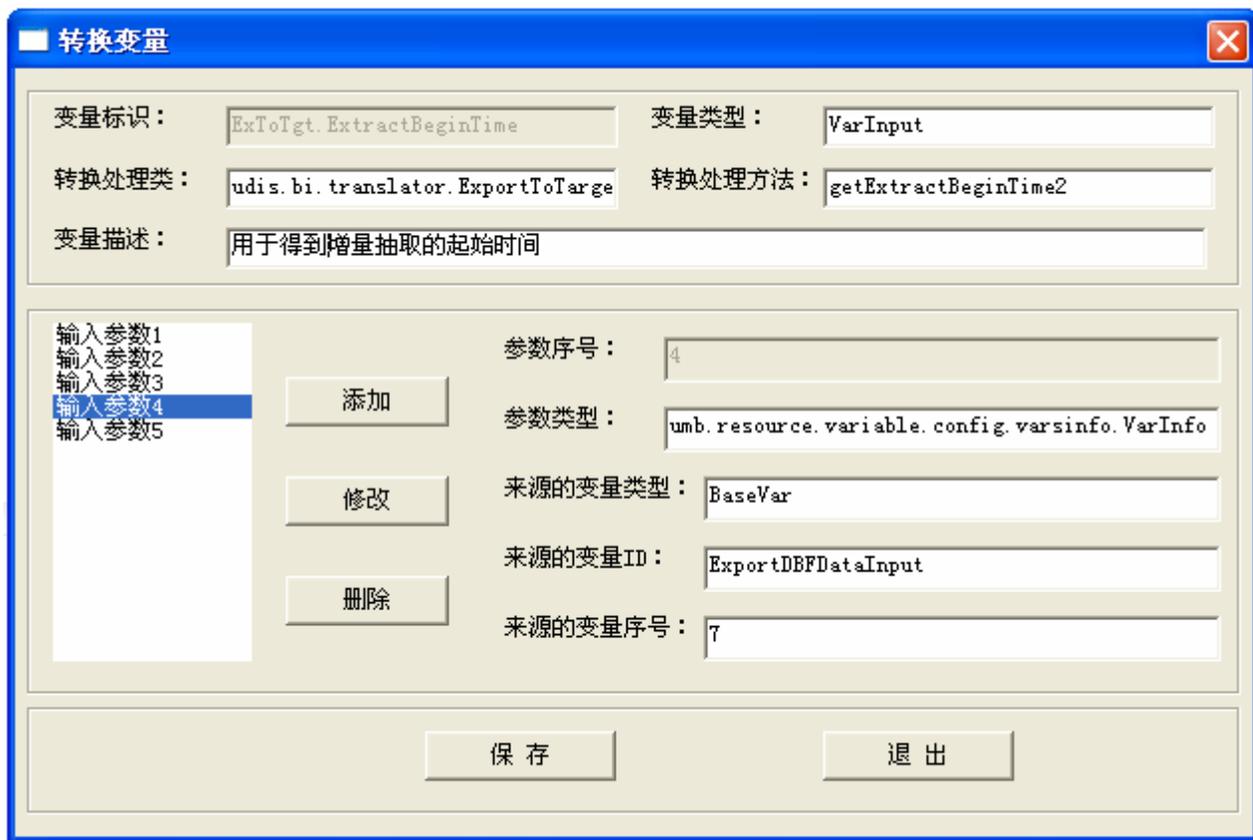
来源的变量序号: 3

保存 退出

变量配置(3)

转换变量：通过转换处理类的方法来实现变量的转换处理。输入的变量经过加工处理后，返回一个新的结果。

主要作为规则表达式中的变量或组合变量的成员等。



转换变量

变量标识： 变量类型：

转换处理类： 转换处理方法：

变量描述：

输入参数1
输入参数2
输入参数3
输入参数4
输入参数5

添加
修改
删除

参数序号：

参数类型：

来源的变量类型：

来源的变量ID：

来源的变量序号：

保存 退出

规则表达式配置

规则表达式是指含有变量的表达式，变量用“#”表示的，其中的变量一般是转换变量。规则表达式在运行时由规则引擎动态处理得到相应的结果值。

规则表达式一般用于动态条件语句、动态SQL语句等场合。

变量规则定义

变量标识: 变量类型:

变量描述:

变量表达式:

变量1
变量2
变量3
变量4
变量5

参数序号:

参数类型:

参数ID:

默认值:

添加 修改 删除

保存 退出

SQL服务(1)

SQL服务：调用静态或动态的数据库SQL语句、数据库SQL函数、数据库存储过程等的服务。

下图是动态SQL语句调用，该SQL语句中含有用“#”表示的变量

SQL服务

服务标识： createETLTMP 服务名称： createETLTMP

数据库标识： local_oracle_db SQL服务类型： SQLSQL

SQL处理： create table #_ETL as select * from #Q# where (LRRQ

输入参数1
输入参数2
输入参数3
输入参数4
输入参数5

序号： 1

类型： VarTrans

参数名称： SimpleTrans.getTargetTableName

默认值：

IO类型： In

添加 修改 删除

保存 退出

输入SQL语句（若含变量用“#”表示）

选择SQLSQL类型

若含有变量，选择相应变量的类型和变量的标示

SQL服务(2)

SQL服务：调用静态或动态的数据库SQL语句、数据库SQL函数、数据库存储过程等的服务。

下图是数据库SQL函数调用，选择相应的返回值/输入/输出/输入输出类型

服务标识: beijing_CP_RUN_ALL_C

数据库标识: local_oracle_db

SQL处理: beijing.CP_RUN_ALL_C

服务名称: CP_RUN_ALL_C

SQL服务类型: SQLFunc

是否有输入参数?

返回值
输入参数1

序号: 0

类型: INTEGER

参数名称: ret

默认值:

IO类型: Return

添加

保存

退出

输入数据库函数或存储过程名称

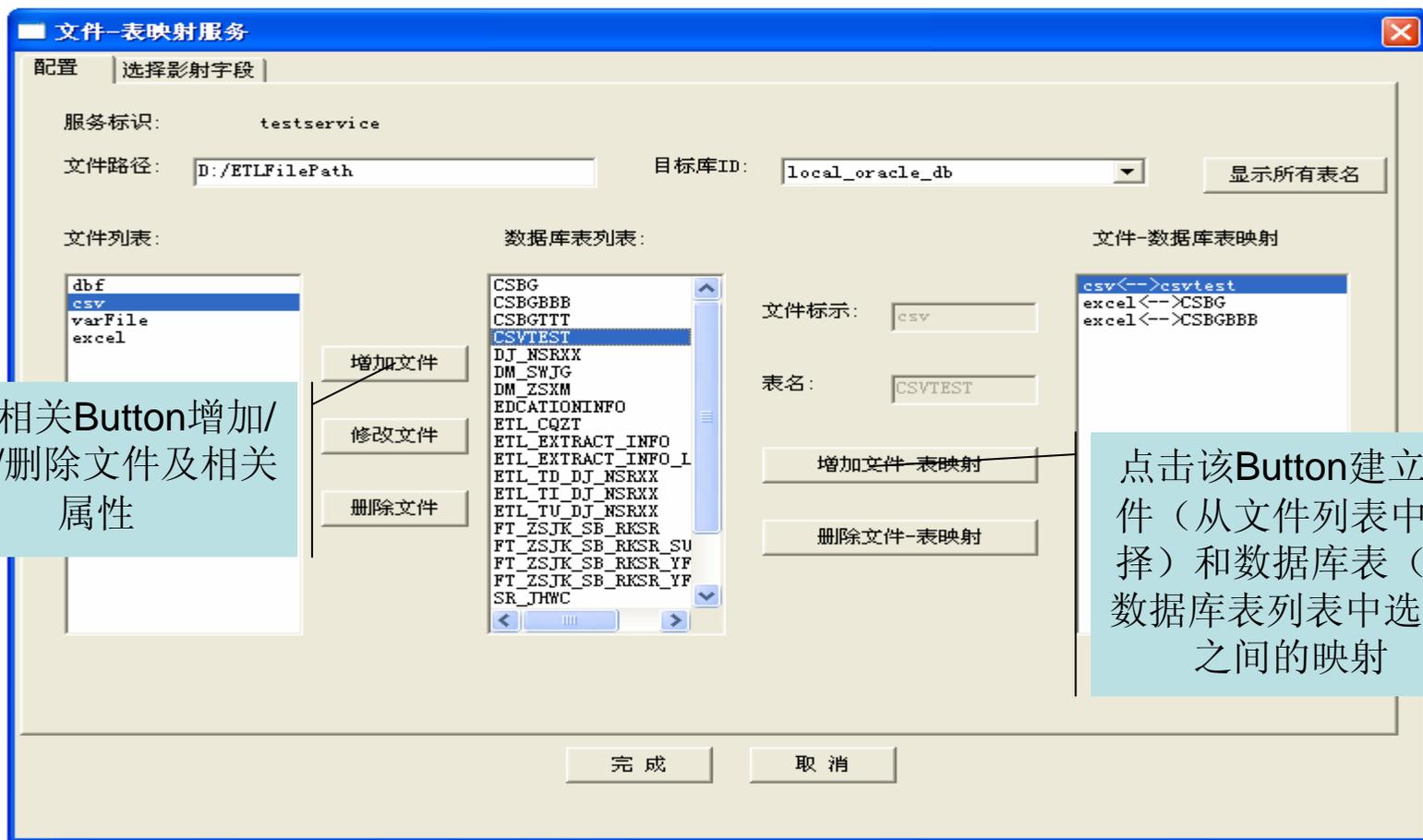
选择SQLFunc类型

选择相应的IO类型，分别为输入、输入输出、输出、返回值

文件-表的转换服务(1)

文件-表转换服务：实现多种格式文件的抽取、并实现和数据库目标表的转换。

下图是选择的文件和选择的数据库表之间的映射配置



点击相关Button增加/修改/删除文件及相关属性

点击该Button建立文件（从文件列表中选择）和数据库表（从数据库表列表中选择）之间的映射

文件-表的转换服务(2)

目前主要支持dbf文件、有分隔符的文本文件、Excel文件等，其他的特殊格式的文件可以通过实现IFileReader接口来引入（细节见二次开发部分）。

下图是文件-表转换服务的文件及相关属性的定义

文件信息定义

文件标示: csv

文件类型: 含有分隔符的文本文件

文件名称: addresses.csv

文件名是否有变量?

来源类型:

任务/变量ID:

返回值/变量成员序:

控制文件名称:

分隔符: , 引用符: "

放弃行数: 0

确认 放弃

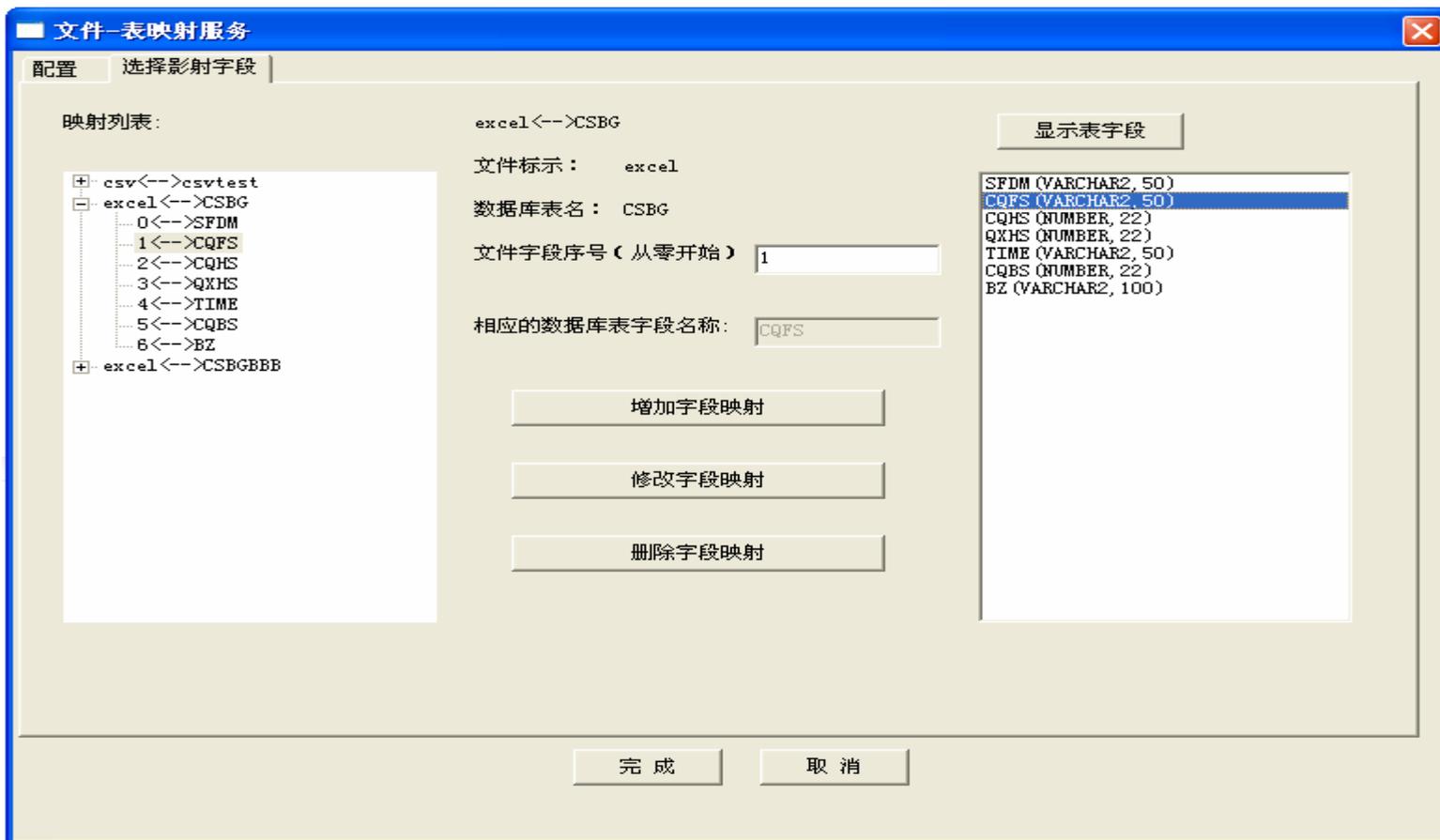
选择相应的文件类型

文件名中可以有变量，也可以直接给定文件名

也可以用控制文件作为其控制信息，控制文件的格式参见二次开发部分

文件-表的转换服务(3)

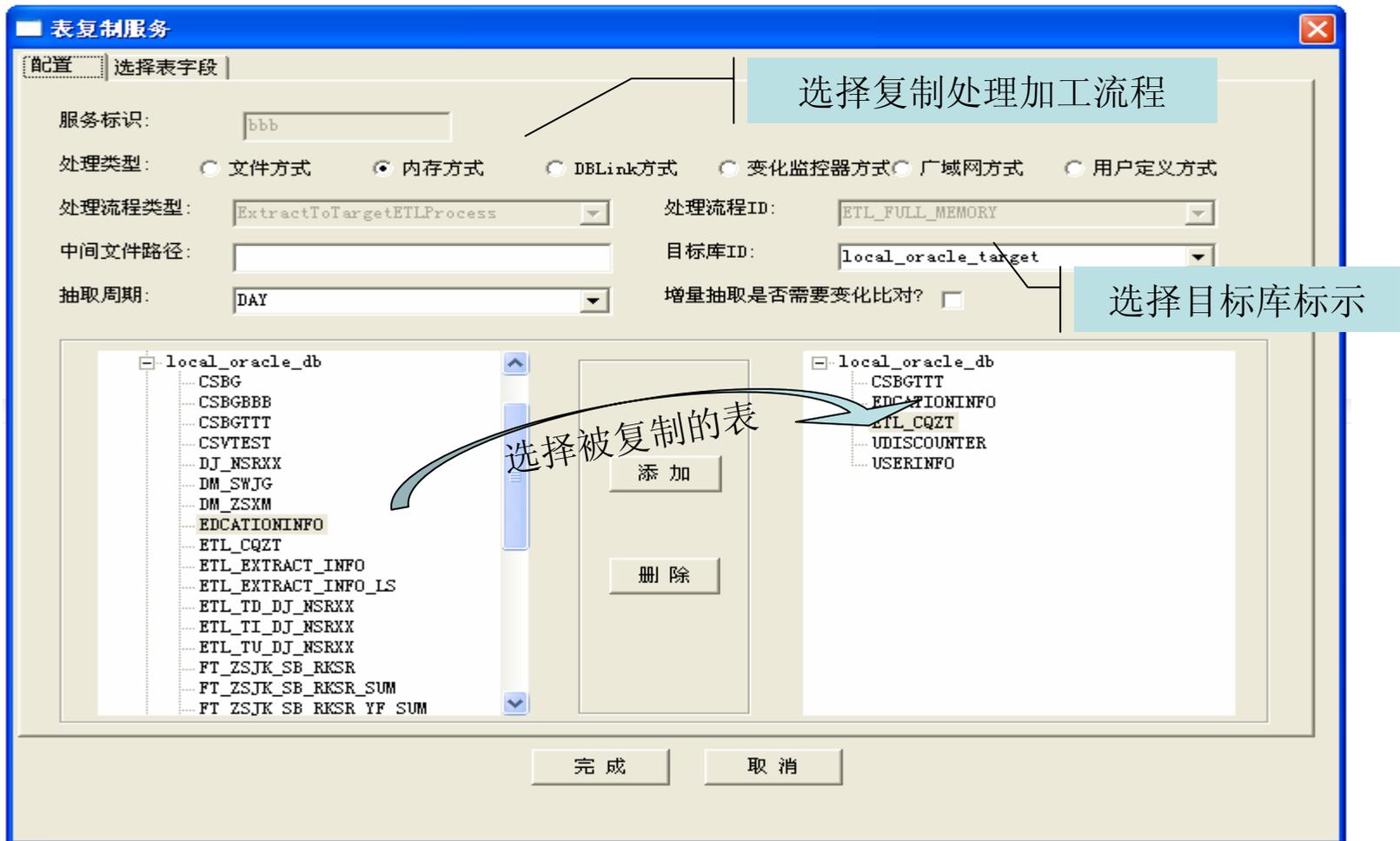
下图是文件-表转换服务的文件中的字段和数据库表中的字段之间的映射定义，系统会自动识别相应的结构



表复制(1)

表复制：用于实现将各业务系统中的数据库中的数据复制到目标库中。

下图是表复制服务的源表、加工处理的流程等信息的选择



表复制(2)

下图是表复制服务的表字段、目标表、过滤条件等信息的输入



表-表转换服务(1)

表-表转换服务：用于实现表的联合、转换/清洗等处理。

通过虚拟列，实现表字段（间）的加工生成新的“虚拟字段”

下图是通过拖拉表而生成源表和目标的字段元数据

The screenshot shows a software interface for data integration management. On the left is a tree view of database connections and tables. The main area displays the metadata for two tables: 'EDUCATIONINFO@local_oracle_db' and 'USERPRINTINFO@local_oracle_target'. The 'EDUCATIONINFO' table has fields: USERID, FIRSTSCHOOL, FIRSTTIME, SECONDSCHOOL, SECONDTIME, and URGENT. The 'USERPRINTINFO' table has fields: USERID, NAME, SEX, JIGUAN, BIRTHDDAY, PROVINCE, and CITY. A '选择的列:' (Selected Columns) table is empty. A '映射' (Mapping) table shows the mapping between the source and target fields. Handwritten annotations in Chinese indicate that the source table metadata is generated by dragging the source table, and the target table metadata is generated by dragging the target table.

字段名	类型
USERID	VARCHAR2 (50)
FIRSTSCHOOL	VARCHAR2 (50)
FIRSTTIME	VARCHAR2 (50)
SECONDSCHOOL	VARCHAR2 (50)
SECONDTIME	VARCHAR2 (50)
URGENT	VARCHAR2 (50)

字段名	类型
USERID	VARCHAR2 (50)
NAME	VARCHAR2 (50)
SEX	CHAR (2)
JIGUAN	VARCHAR2 (50)
BIRTHDDAY	DATE (7)
PROVINCE	VARCHAR2 (50)
CITY	VARCHAR2 (50)

字段名	类型	映射
USERID	VARCHAR2 (50)	
NAME	VARCHAR2 (50)	
FIRSTSCHOOL	VARCHAR2 (50)	
FIRSTTIME	VARCHAR2 (50)	
SECONDSCHOOL	VARCHAR2 (50)	
SECONDTIME	VARCHAR2 (50)	
LASTSCHOOL	VARCHAR2 (50)	

通过拖拉生成源表字段元数据

通过拖拉生成目标表字段元数据

表-表转换服务(2)

从源表中选择的字段拖拉生成选择列，在虚拟列中通过浮动菜单单出创建虚拟列创建/修改界面；
系统自动根据源表和目标表的元数据自动完成格式的转换；

对于数据的加工处理，可以通过虚拟列的定义完成，已定义的虚拟列可以作为新定义的虚拟列的输入。虚拟列的特殊加工处理方法可以扩展，详细参见二次开发部分

The screenshot displays the interface for creating virtual columns. On the left, two source tables are shown: 'userinfo@local_oracle_db' and 'educationInfo@local_oracle_db'. The 'userinfo@local_oracle_db' table has columns: USERID, NAME, SEX, JIGUAN, BIRTHDDAY, PROVINCE, CITY. The 'educationInfo@local_oracle_db' table has columns: USERID, FIRSTSCHOOL, FIRSTTIME, SECONDSCHOOL, SECONDTIME, LASTSCHOOL, LASTTIME, DEGREE. In the center, a '选择的列:' (Selected Columns) list contains columns from both tables. A '虚拟列:' (Virtual Columns) list at the bottom shows 'address' (StringCat), 'tttt' (test), 'subaddr', and 'counter'. A context menu is open over 'tttt' with options: '移去虚拟列', '添加虚拟列', and '编辑虚拟列'. On the right, the '虚拟列设置' (Virtual Column Settings) dialog is open for the 'address' column. It shows '处理方法' (StringCat) and '参数定义' (t.userinfo.PROVINCE, t.userinfo.CITY, t.userinfo.DIQU, t.userinfo.ADDR). The '参数来源' (Parameter Source) is set to '选择的列' (Selected Column), and '选择列名' (Selected Column Name) is 'userinfo.CITY'. The '其他' (Other) field is set to '整形' (Integer). Buttons for '退出' (Exit) and '保存' (Save) are at the bottom.

通过拖拉生成选择列

通过拖拉生成选择列

字段名	类型
USERID	VARCHAR2 (50)
NAME	VARCHAR2 (50)
SEX	CHAR (2)
JIGUAN	VARCHAR2 (50)
BIRTHDDAY	DATE (7)
PROVINCE	VARCHAR2 (50)
CITY	VARCHAR2 (50)

字段名	类型
USERID	VARCHAR2 (50)
FIRSTSCHOOL	VARCHAR2 (50)
FIRSTTIME	VARCHAR2 (50)
SECONDSCHOOL	VARCHAR2 (50)
SECONDTIME	VARCHAR2 (50)
LASTSCHOOL	VARCHAR2 (50)
LASTTIME	VARCHAR2 (50)
DEGREE	VARCHAR2 (50)

字段名	类
userinfo.USERID	VARCH
userinfo.NAME	VARCH
userinfo.SEX	CHAR (
userinfo.JIGUAN	VARCH
userinfo.BIRTHDDAY	DATE (
userinfo.PROVINCE	VARCH
userinfo.CITY	VARCH
userinfo.DIQU	VARCH
userinfo.ADDR	VARCH
userinfo.EMAIL	VARCH
userinfo.TEL	VARCH
educationInfo.USERID	VARCH
educationInfo.FIRSTS...	VARCH
educationInfo.FIRSTTIME	VARCH
educationInfo.SECOND...	VARCH
educationInfo.SECOND...	VARCH

字段名	类型
address	StringCat (t...
tttt	test (t.user...
subaddr	
counter	

虚拟列设置

虚拟列名字: address 处理方法: StringCat

虚拟列参数: t.userinfo.PROVINCE, t.userinfo.CITY, t.userinfo.DIQU, t.userinfo.ADDR

参数定义: 上移, 下移, 新增, 修改, 删除

参数来源: 选择的列 虚拟列 其他

选择列名: userinfo.CITY

虚拟列名:

其他: 类型: 整形

退出 保存

表-表转换服务(3)

The screenshot displays a database tool interface for table conversion. It is divided into three main sections: '修改联合条件' (Modify Joint Conditions), '保存且退出' (Save and Exit), and '退出' (Exit).

修改联合条件 (Modify Joint Conditions): A dialog box is open with the filter condition `userinfo.userid=educationInfo.userid`. It has '确认' (Confirm) and '放弃' (Cancel) buttons.

userinfo@local_oracle_db (Source Table):

字段名	类型
USERID	VARCHAR2 (50)
NAME	VARCHAR2 (50)
SEX	CHAR (2)
JIGUAN	VARCHAR2 (50)
BIRTHDAY	DATE (7)
USERID	VARCHAR2 (50)
FIRSTSCHOOL	VARCHAR2 (50)
FIRSTTIME	VARCHAR2 (50)
SECONDSCHOOL	VARCHAR2 (50)
SECONDTIME	VARCHAR2 (50)
LASTSCHOOL	VARCHAR2 (50)
LASTTIME	VARCHAR2 (50)
DEGREE	VARCHAR2 (50)

保存且退出 (Save and Exit): Shows the selected columns and virtual columns.

选择的列 (Selected Columns):

字段名	类型
userinfo.USERID	VARCHAR2 (50)
userinfo.NAME	VARCHAR2 (50)
userinfo.SEX	CHAR (2)
userinfo.JIGUAN	VARCHAR2 (50)
userinfo.BIR	DATE (7)
educationInfo...	VARCHAR2 (50)
educationInfo...	VARCHAR2 (50)
educationInfo...	VARCHAR2 (50)

虚拟列 (Virtual Columns):

字段名	类型
address	StringCat (t...
tttt	test (t.user...
subaddr	StringSubE (...
counter	Counter (o.u...

退出 (Exit): Shows the mapping table for the target table.

userprintinfo@local_oracle_target (Target Table):

字段名	映射字段
USERID	t.userinfo.userid
NAME	t.userinfo.name
FIRSTSCHOOL	
SECONDSCHOOL	v.counter
LASTSCHOOL	v.tttt

Handwritten annotations in Chinese provide context:

- 通过拖拉生成表字段之间的映射 (Generated by dragging to create mapping between table fields)
- 通过拖拉生成表字段之间的映射 (Generated by dragging to create mapping between table fields)

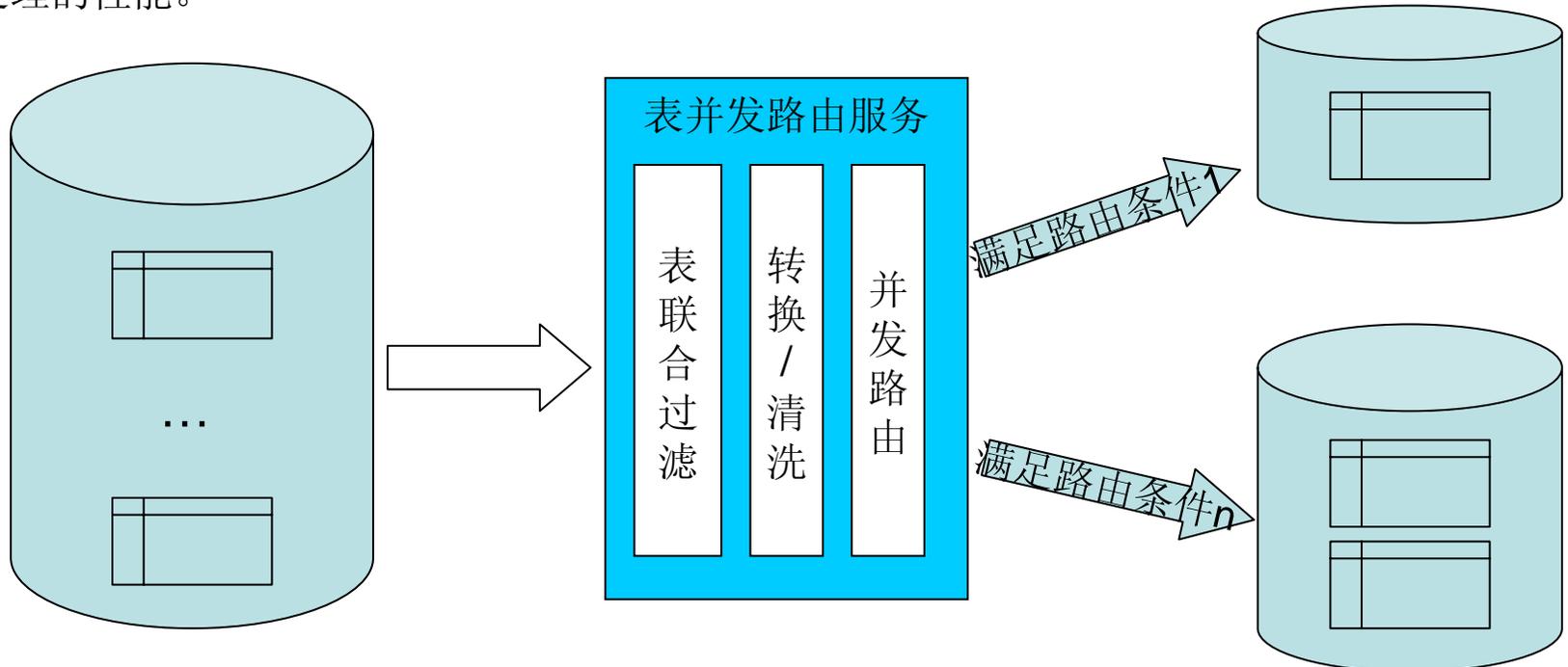
表并发路由服务(1)

表并发路由服务：主要是把源表的数据经过联合、过滤、清洗/转换等处理后，根据各目标表的路由条件，并发的将结果路由到相应的目标库表中。

该处理的转换/清洗的方法同表-表的转换处理的方法。

路由条件主要支持：“等于”，“不等于”，“小于”，“小于等于”，“大于”，“大于等于”，“规则表达式”，“是空”，“非空”，“在列表中”，“包含”，“开始字段”，“结束字段”等多种路由规则。

该路由服务，采用“一次抽取，并发路由”的机制，目的是为了减轻源数据库的压力，同时也提高处理的性能。



表并发路由服务(2)

下图是通过拖拉表而生成源表和目标表的字段元数据

睿智数据集成管理

系统 资源 基础服务 业务服务 流程 高级 帮助

基础配置 运行监控 查看表路由处理 编辑表路由处理

修改联合条件

USERINFO@local_oracle_db

字段名	类型
USERID	VARCHAR2 (50)
NAME	VARCHAR2 (50)
SEX	CHAR (2)
JIGUAN	VARCHAR2 (50)
BIRTHDAY	DATE (7)
PROVINCE	VARCHAR2 (50)
CITY	VARCHAR2 (50)

保存且退出

退出

选择的列:

字段名	类型
-----	----

虚拟列:

字段名	类型
-----	----

路由目的表:

增加路由表 修改路由表 删除路由表

USERPRINTINFO@local_oracle_target

字段名	类型	映射
USERID	VARCHAR2 (50)	
NAME	VARCHAR2 (50)	
FIRSTSCHOOL	VARCHAR2 (50)	
FIRSTTIME	VARCHAR2 (50)	
SECONDSCHOOL	VARCHAR2 (50)	
SECONDTIME	VARCHAR2 (50)	
LASTSCHOOL	VARCHAR2 (50)	
LASTTIME	VARCHAR2 (50)	
DEGREE	VARCHAR2 (50)	
ADDRESS	VARCHAR2 (150)	
PHONE	VARCHAR2 (150)	

修改路由条件

通过拖拉生成源表字段元数据

通过拖拉生成目标表字段元数据

表并发路由服务(3)

从源表中选择的字段拖拉生成选择列，在虚拟列中通过浮动菜单单出创建虚拟列创建/修改界面；
系统自动根据源表和目标表的元数据自动完成格式的转换；

对于数据的加工处理，可以通过虚拟列的定义完成，已定义的虚拟列可以作为新定义的虚拟列的输入。虚拟列的特殊加工处理方法可以扩展，详细参见二次开发部分

字段名	类型
USERID	VARCHAR2 (50)
NAME	VARCHAR2 (50)
SEX	CHAR (2)
JIGUAN	VARCHAR2 (50)
BIRTHDDAY	DATE (7)
PROVINCE	VARCHAR2 (50)
CITY	VARCHAR2 (50)

字段名	类型
USERID	VARCHAR2 (50)
FIRSTSCHOOL	VARCHAR2 (50)
FIRSTTIME	VARCHAR2 (50)
SECONDSCHOOL	VARCHAR2 (50)
SECONDTIME	VARCHAR2 (50)
LASTSCHOOL	VARCHAR2 (50)
LASTTIME	VARCHAR2 (50)
DEGREE	VARCHAR2 (50)

选择的列：	字段名	类
	userinfo.USERID	VARCH
	userinfo.NAME	VARCH
	userinfo.SEX	CHAR (
	userinfo.JIGUAN	VARCH
	userinfo.BIRTHDDAY	DATE (
	userinfo.PROVINCE	VARCH
	userinfo.CITY	VARCH
	userinfo.DIQU	VARCH
	userinfo.ADDR	VARCH
	userinfo.EMAIL	VARCH
	userinfo.TEL	VARCH
	educationInfo.USERID	VARCH
	educationInfo.FIRSTS...	VARCH
	educationInfo.FIRSTTIME	VARCH
	educationInfo.SECOND...	VARCH
	educationInfo.SECOND...	VARCH

虚拟列：	字段名	类型
	address	StringCat (t...
	tttt	test (t.user...
	subaddr	移去虚拟列
	counter	添加虚拟列
		编辑虚拟列

通过拖拉生成选择列

通过拖拉生成虚拟列

虚拟列设置

虚拟列名字: address 处理方法: StringCat

虚拟列参数:

- t.userinfo.PROVINCE
- t.userinfo.CITY
- t.userinfo.DIQU
- t.userinfo.ADDR

参数来源: 选择的列 虚拟列 其他

选择列名: userinfo.CITY

虚拟列名:

其他: 类型: 整形

退出 保存

表并发路由服务(4)

通过拖拉生成目标表和选择字段/虚拟字段的映射关系，选择的表字段显示以“t.”开头，虚拟字段以“v.”开头

目标表的路由依据字段可以是选择的表字段，也可以是虚拟字段

修改联合条件

保存且退出

退出

USERINFO@local_oracle_db

字段名	类型
BIRTHDDAY	DATE (7)
PROVINCE	VARCHAR2 (50)
CITY	VARCHAR2 (50)
DIQU	VARCHAR2 (50)
ADDR	VARCHAR2 (150)
EMAIL	VARCHAR2 (150)
TEL	VARCHAR2 (50)

选择的列：

字段名	类
USERINFO. USERID	VARCH
USERINFO. NAME	VARCH
USERINFO. SEX	CHAR (
USERINFO. PROVINCE	VARCH
USERINFO. CITY	VARCH
USERINFO. DIQU	CH
USERINFO. ADDR	VARCH
EDUCATIONINFO. DEGREE	VARCH

路由目的表：

增加路由表 修改路由表 删除路由表

EDUCATIONINFO@local_oracle_target

字段名	类型	映射
USERID	VARCHAR2 (50)	t. USE
NAME	VARCHAR2 (50)	t. USE
FIRSTSCHOOL	VARCHAR2 (50)	
FIRSTTIME	VARCHAR2 (50)	
SECONDSCHOOL	VARCHAR2 (50)	
SECONDTIME	VARCHAR2 (50)	
LASTSCHOOL	VARCHAR2 (50)	
LASTTIME	VARCHAR2 (50)	
DEGREE	VARCHAR2 (50)	t. EDC
ADDR	VARCHAR2 (150)	v. add
EMAIL	VARCHAR2 (150)	

表路由条件

路由依据字段： t.USERINFO.CITY

路由处理类型： 等于

相应的处理条件： 北京

确认 放弃

通过拖拉生成表字段之间的映射

修改路由条件

表并发路由服务(5)

修改联合条件

字段名	类型
USERID	VARCHAR2 (50)
NAME	VARCHAR2 (50)
SEX	CHAR (2)
JIGUAN	VARCHAR2 (50)
BIRTHDDAY	DATE (7)
PROVINCE	VARCHAR2 (50)
CITY	VARCHAR2 (50)

字段名	类型
USERID	VARCHAR2 (50)
FIRSTSCHOOL	VARCHAR2 (50)
FIRSTTIME	VARCHAR2 (50)
SECONDSCHOOL	VARCHAR2 (50)
SECONDTIME	VARCHAR2 (50)
LASTSCHOOL	VARCHAR2 (50)
LASTTIME	VARCHAR2 (50)
DEGREE	VARCHAR2 (50)

保存且退出

选择的列：

字段名	类
userinfo.USERID	VARCHAR2 (50)
userinfo.NAME	VARCHAR2 (50)
userinfo.SEX	CHAR (2)
userinfo.JIGUAN	VARCHAR2 (50)
userinfo.BIRTHDDAY	DATE (7)
userinfo.PROVINCE	VARCHAR2 (50)
userinfo.CITY	VARCHAR2 (50)
userinfo.DIQU	VARCHAR2 (50)
userinfo.ADDR	VARCHAR2 (50)
userinfo.EMAIL	VARCHAR2 (50)
userinfo.TEL	VARCHAR2 (50)
edcationInfo.US...	VARCHAR2 (50)
edcationInfo.FI...	VARCHAR2 (50)
edcationInfo.FI...	VARCHAR2 (50)
edcationInfo.SE...	VARCHAR2 (50)
edcationInfo.SE...	VARCHAR2 (50)

虚拟列：

字段名	类型
address	StringCat (t...
tttt	test (t. user...
subaddr	StringSubE (...
counter	Counter (o. u...

退出

路由目的表：

userprintinfo@local_oracle_source
userprintinfo@local_oracle_target

增加路由表

修改路由表

删除路由表

字段名	映射字段
USERID	t.userinfo.userid
NAME	t.userinfo.name
FIRSTSCHOOL	
FIRSTTIME	
SECONDSCHOOL	v.counter
SECONDTIME	
LASTSCHOOL	v.tttt
LASTTIME	
DEGREE	t.edcationInfo.degree
ADDRESS	v.address
EMAIL	

修改路由条件

事实表处理服务

事实表处理服务：主要是完成数据表中的根据维度的分组聚类的处理。

事实表是指用来保存结果数据的表；被加工表是指被加工的数据来源表；

加工字段是指将要保存在事实表中的维度值、一些分组聚类数值等

分组维度是指在事实表中保存的数据分组的依据。

服务修改

标示: ZJ_RKSR

服务描述: String

事实表名: FACT_SAMPLE 数据库标示: local_oracle_db

加工字段: RQ, DQ_DM, CPDL_DM, CPZL_DM, CPXL_DM, HY_DM, sum(XS),

被加工表: YW_SAMPLE

加工条件: RQ between ldt_qrq and ldt_zrq and DQ_DM like

分组维度: RQ, DQ_DM, CPDL_DM, CPZL_DM, CPXL_DM, HY_DM

看SQL语句 确认 放弃

前端报表软件触发

数据加工完成后，可通过前端报表软件触发接口，来触发前端报表软件作预加工

下图是和Brio的触发配置

The image shows a software window titled "服务修改" (Service Modification) with a close button in the top right corner. The window contains the following fields and controls:

- 服务标识:** BrioTrigger
- 服务描述:** Brio view trigger
- 触发命令1:** (Empty text area)
- 用户名称:** administrator
- 密码:** *****
- 服务器地址:** localhost
- 端口:** 1800
- 触发器:** etl1

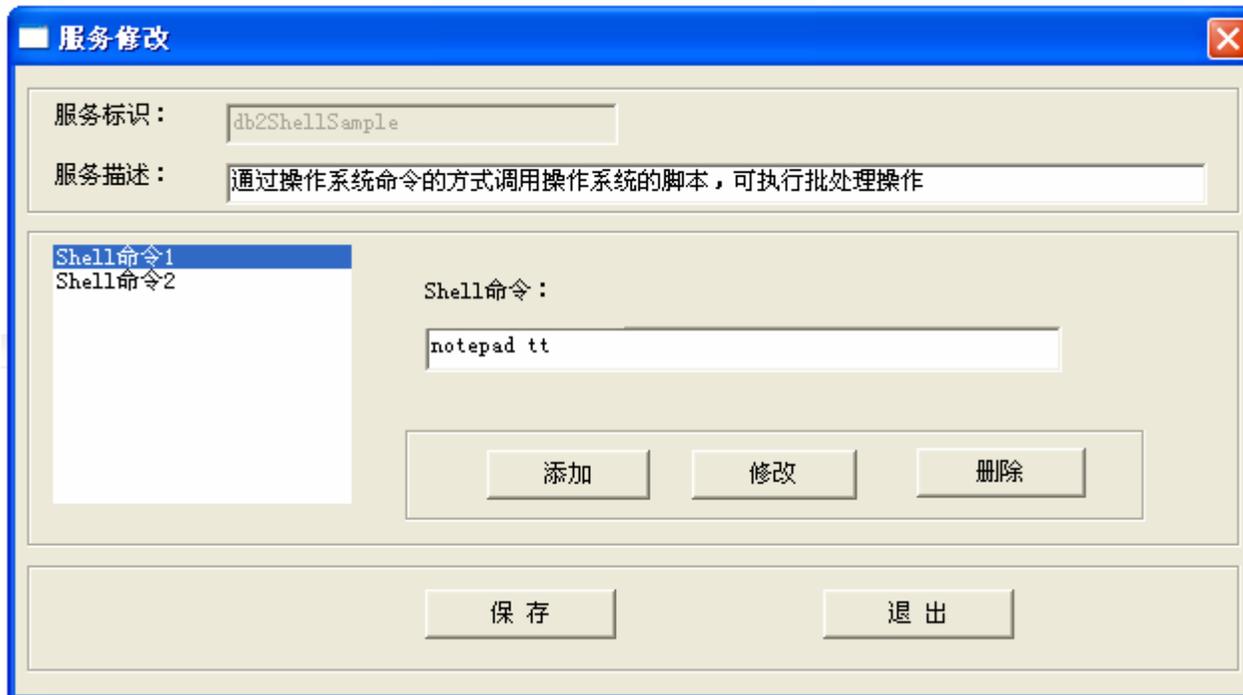
At the bottom of the window, there are two rows of buttons:

- Row 1: 添加 (Add), 修改 (Modify), 删除 (Delete)
- Row 2: 保存 (Save), 退出 (Exit)

Shell脚本服务

Shell脚本服务：通过操作系统命令的方式调用操作系统的脚本，可执行批处理操作

该方法可以用来调用操作系统的命令，通过操作系统命令实现和其他业务系统的触发调用
可以执行多个命令的批量操作。



The image shows a Windows-style dialog box titled "服务修改" (Service Modification). It contains the following fields and controls:

- 服务标识:** A text box containing "db2ShellSample".
- 服务描述:** A text box containing "通过操作系统命令的方式调用操作系统的脚本，可执行批处理操作".
- Shell命令列表:** A list box containing "Shell命令1" (selected) and "Shell命令2".
- Shell命令:** A text box containing "notepad tt".
- Buttons:** "添加" (Add), "修改" (Modify), "删除" (Delete), "保存" (Save), and "退出" (Exit).

Ftp服务

Ftp服务主要用于通过FTP服务器来传送数据的方式，一般应用于广域网的数据集成。

主要包括FTP客户端服务配置、FTP服务器端服务配置

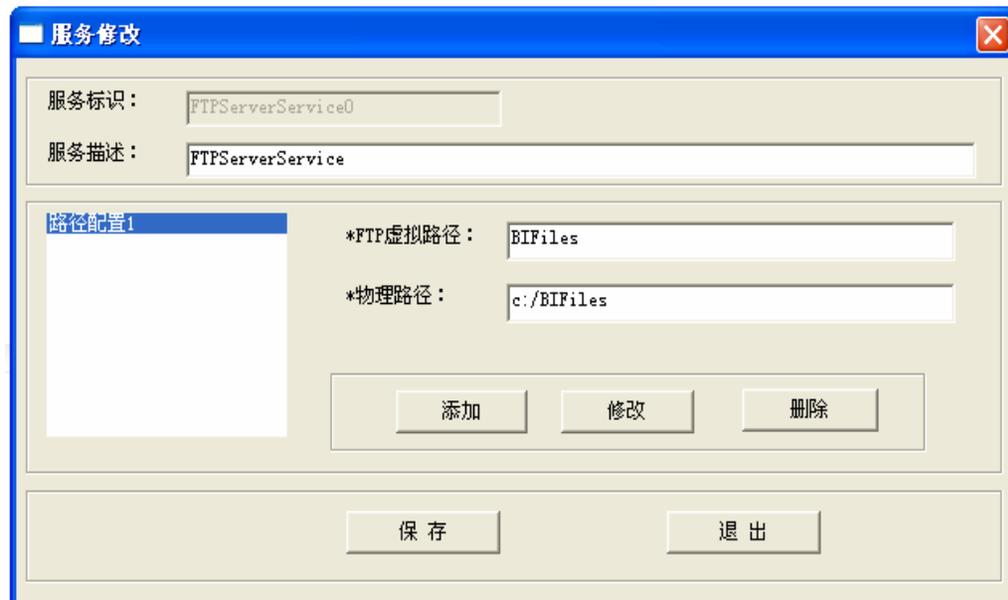


The screenshot shows a 'Service Modification' dialog box with the following fields and values:

- 标示: ftpl
- 服务描述: FTP Service
- FTP服务器地址: 10.1.151.104
- 端口号: 21
- 连接方式: PASV
- 建立连接最大延时: 20000
- 送数据块的大小: 4096
- 用户名: bi
- 密码: bi

Buttons at the bottom: 确认 (Confirm), 放弃 (Cancel).

FTP客户端服务配置



The screenshot shows a 'Service Modification' dialog box with the following fields and values:

- 服务标识: FTPServerService0
- 服务描述: FTPServerService
- 路径配置1: (empty list)
- *FTP虚拟路径: BIFiles
- *物理路径: c:/BIFiles

Buttons for path configuration: 添加 (Add), 修改 (Modify), 删除 (Delete).

Buttons at the bottom: 保存 (Save), 退出 (Exit).

FTP服务器端服务配置

流程(1)

预定义的数据加工服务，经过鼠标的拖拉作为工作流的处理节点。流程也可以作为另一个流程的处理节点。

The screenshot displays a software interface for '睿智数据集成管理' (Rui Zhi Data Integration Management). The main window shows a workflow diagram titled '客户评估数据加工处理流程图' (Customer Evaluation Data Processing Flowchart). The workflow starts with a '开始' (Start) node, leading to a '并行抽取' (Parallel Extraction) block containing '甘肃省增量抽取' (Gansu Province Incremental Extraction) and '甘肃省全量抽取' (Gansu Province Full Extraction). This is followed by '文件-表转换' (File-Table Conversion), '表转换清洗处理' (Table Conversion and Cleaning), and '事实表加工' (Fact Table Processing). A decision node '是否有意外' (Are there any anomalies?) branches into '意外处理' (Anomaly Handling) and '形成客户评估清册' (Forming Customer Evaluation Register). The latter leads to 'SQL' and '并行处理' (Parallel Processing), which includes '前端显示预加工' (Front-end display pre-processing) and '数据并发路由' (Data concurrent routing).

第一步 先作数据并发抽取：
从业务数据库中对一些业务表的作增量抽取
从业务数据库中对一些业务表作全量抽取
对一些部门间的交换数据做文件-表的转换抽取

第二步 抽取完成后，做表的转换/清洗加工
非规范的数据作规范处理，不合理的数据作清洗操作，形成统一的业务标准

第三步 作事实表加工
对规范的数据按照时间、地区、产品等不同维度作分组聚类操作，形成事实表
在此次操作过程中，分析是否有意外产生：
如果有意外，调用意外处理过程
如果没有意外，继续加工调用第四步。

第四步 形成客户评估清册
调用含有变量的SQL服务，经过规则引擎的处理，形成客户评估清册

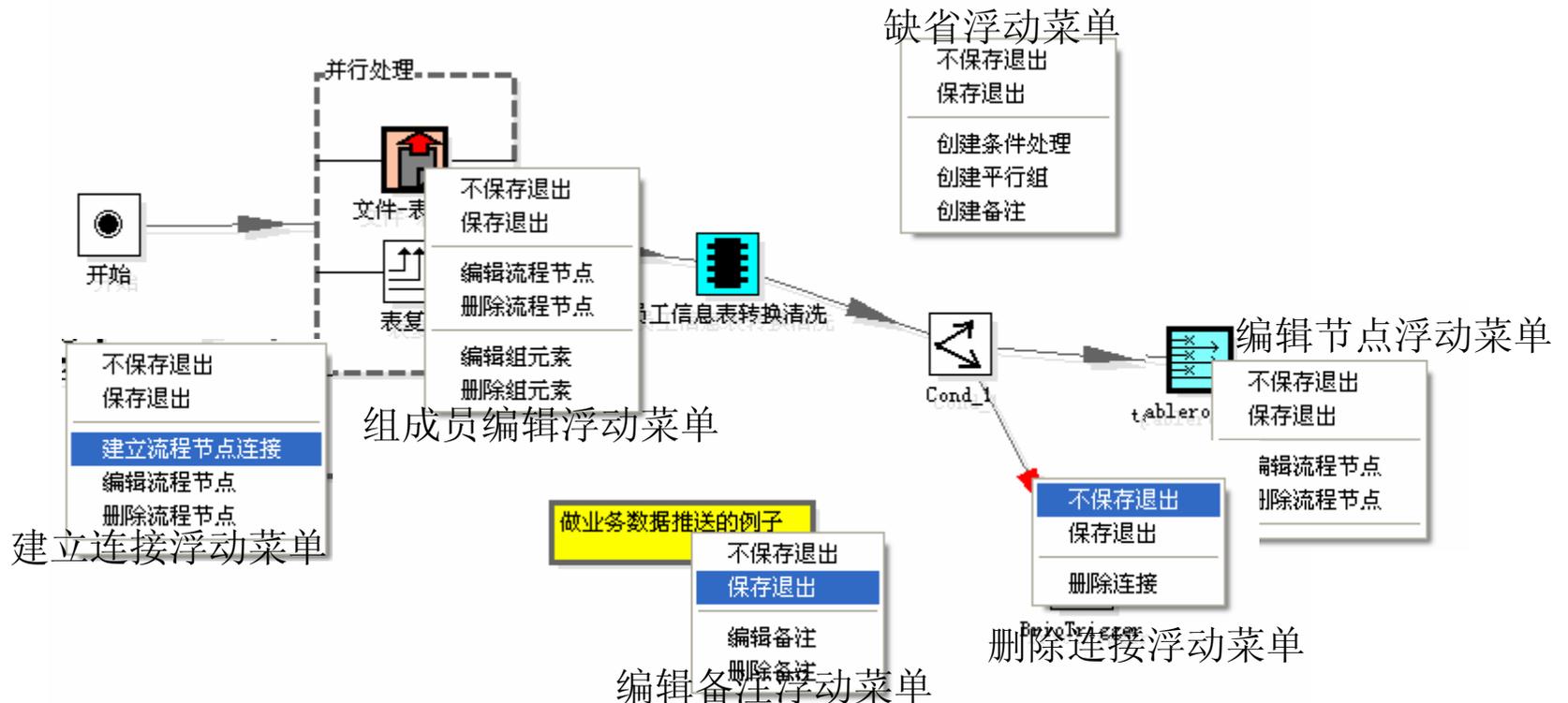
第五步 并行处理
其中一个并行加工过程为调用前端的业务显示模块预处理，供前端客服人员使用
其中另一个开发为数据并发路由，根据路由条件的不同，数据传送到不同的业务表中

流程(2)

为了提高操作速度，采用智能化的浮动菜单的方式来引导操作。

根据目前操作的不同，其浮动菜单也有不同（以下是六种不同情况下的浮动菜单）：

当编辑并行组的成员时弹出组成员“编辑浮动菜单”；当选择两个节点来建立节点连接时弹出“建立连接浮动菜单”；当选择一个节点时弹出“编辑节点浮动菜单”；当选择备注时弹出“编辑备注浮动菜单”；当选择连接线时，弹出“删除连接浮动菜单”；当没有选择时，弹出“缺省浮动菜单”



流程(3)

根据选择的节点类型不同，弹出的节点编辑对话框也不同，当选择开始节点时，弹出“开始节点编辑对话框”；当选择任务节点时，弹出“任务节点编辑对话框”；当选择条件节点时，弹出“条件编辑对话框”；当选择备注节点时，弹出“备注编辑对话框”。

The screenshot displays the '睿智数据集成管理' (Wise Data Integration Management) software interface. The main window shows a workflow editor with a tree view on the left and a central workspace. The workflow includes a '开始' (Start) node, a '并行处理' (Parallel Processing) container with '文件-表转换' (File-Table Conversion) and '表复制' (Table Copy) nodes, a '员工信息表转换清洗' (Employee Information Table Conversion and Cleaning) node, and a 'Cond_1' (Condition) node. A yellow callout box points to the '员工信息表转换清洗' node with the text '做业务数据推送的例子' (Example of business data push).

Four dialog boxes are overlaid on the interface:

- 开始节点编辑对话框 (Start Node Edit Dialog):** Located at the bottom left, it contains fields for '节点类型' (Node Type) set to 'StartType', and checkboxes for '是否有变量作为输入?' (Has variable as input?), '是否有变量作为输出?' (Has variable as output?), and '意外时是否终止运行?' (Stop execution on error?).
- 任务节点编辑对话框 (Task Node Edit Dialog):** Located at the top center, it contains fields for '节点类型' (Node Type) set to 'GroupMember/Task_4', '节点ID', '节点名字' (Node Name) set to '文件-表转换', '节点描述', '任务类型' (Task Type) set to 'Service', '服务类型' (Service Type) set to 'ExtractSingleFileToTable', and '服务ID' (Service ID) set to 'testservice'. It also has a checkbox for '是否有变量作为输入?' (Has variable as input?) and a '变量类型' (Variable Type) dropdown.
- 备注编辑对话框 (Note Edit Dialog):** Located at the bottom center, it contains a text area for '内容' (Content) with the text '做业务数据推送的例子' (Example of business data push).
- 条件编辑对话框 (Condition Edit Dialog):** Located at the bottom right, it contains fields for '节点信息' (Node Info) set to 'Node_9/Cond/Cond_1' and '节点名称' (Node Name) set to 'Cond_1'. It has a checked '意外处理?' (Error Handling?) checkbox and a '条件处理方法选择' (Condition Handling Method Selection) dropdown. It also has '如果' (If) and '是真的' (If True) sections with '执行' (Execute) dropdowns set to 'Node_8' and 'Node_6' respectively.

流程(4)

变量流：变量可以作为整个流程的输入，而且处理节点也可以由变量作为节点的输入，进而形成变量流。一般基础变量作为整个流程的输入变量，在“开始节点编辑对话框”里选择；组合变量作为可以作为处理节点的输入变量，在“任务节点编辑对话框”里选择。处理节点的输入变量即可来源于流程的全局变量，也可以是特殊处理的转换变量值。

睿智数据集成管理

系统 资源 基础服务 业务服务 流程 高级 帮助

基础配置
数据库连接
变量
基础变量
组合变量
转换变量
规则
服务
基础服务
FTP传送服务
FTP服务器端处理
Socket通讯客户
Socket通讯服务
定时服务
ETL定时服务
动态SQL服务2
Shell脚本服务
业务服务
动态SQL服务1
动态SQL函数服务
客户加工过程

运行监控
正在运行
成功完成
停止的
意外的

节点信息

节点类型: Task/Task_10
节点ID: Node_7
节点名字: 员工信息表转换清洗
节点描述:
任务类型: Service
服务类型: TableMapService
tablemap1
服务ID:
是否有变量作为输入?
变量类型: 组合类型
变量ID: Task1Input

选择节点输入变量

节点信息

节点类型: StartType
是否有变量作为输入?
变量类型: 基础类型
变量ID: TestProcessInput
是否有变量作为输出?
变量类型:

选择全局变量

节点信息

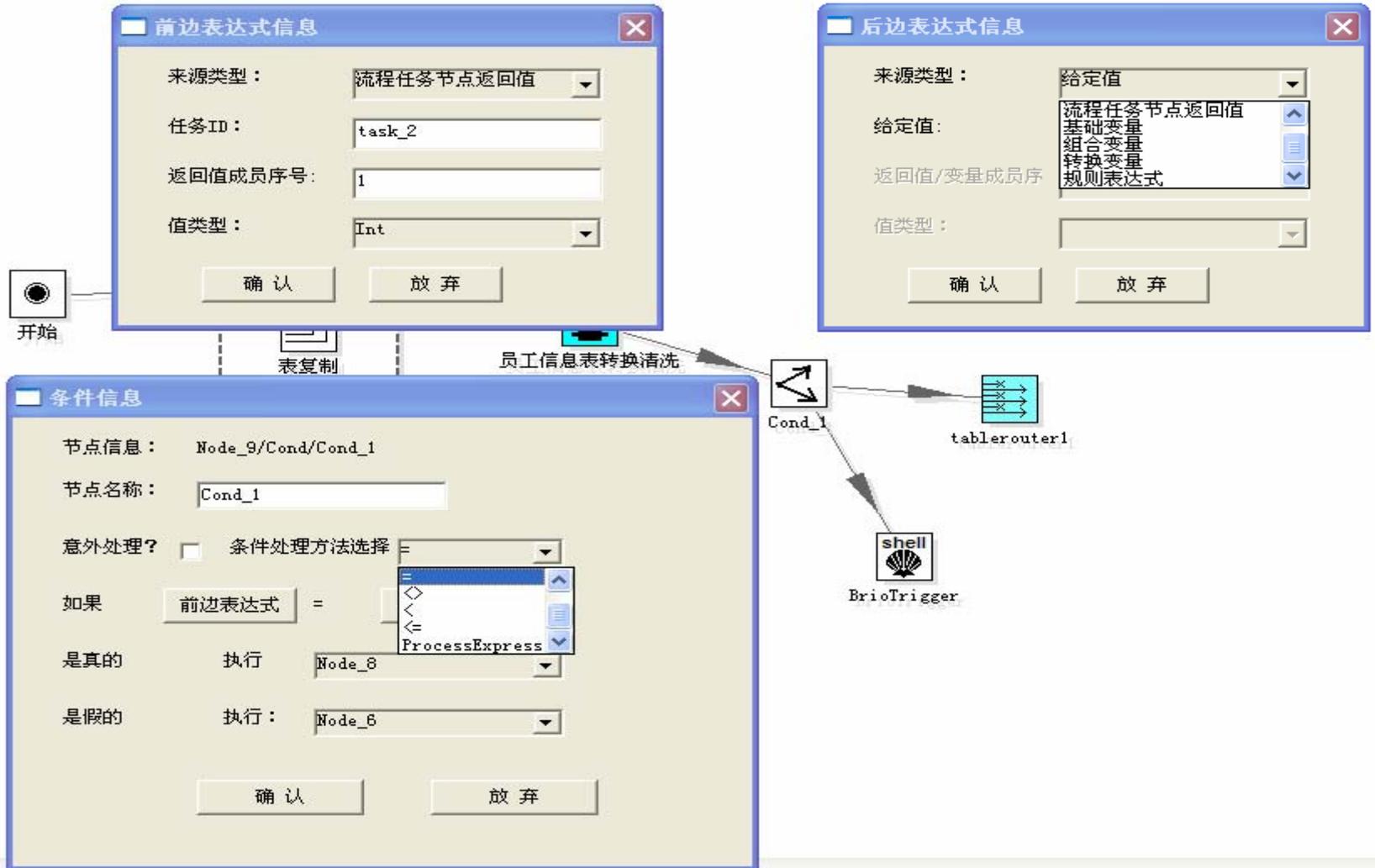
节点类型: Task/Task_8
节点ID: Node_6
节点名字: shell
节点描述:
任务类型: Service
服务类型: TableRouterService
tablerouter1
服务ID:
是否有变量作为输入?
变量类型: 组合类型
变量ID: Task2Input

选择节点输入变量

流程(5)

条件的支持：条件节点作为流程的判断处理，根据条件的满足情况作相应的处理。

条件可以来源于流程任务节点的返回值、处理的变量、规则表达式、静态值等



流程(6)

意外处理：当流程的处理节点出现意外时，可以终止整个流程的运行，也可以无视意外继续运行，也可以交由流程的条件节点来处理。

The screenshot displays the 'Xunzhi Data Integration Management' software interface. The main window shows a workflow editor with the following components:

- 开始** (Start) node
- 并行处理** (Parallel Processing) container containing:
 - 文件-表转换** (File-Table Conversion) node
 - 表复制** (Table Copy) node
- 员工信息表转换清洗** (Employee Information Table Conversion and Cleaning) node
- 是否意外?** (Is there an error?) condition node
- tablerouter1** node
- shell** node (labeled BrioTrigger)

Two dialog boxes are overlaid on the workflow:

- 条件编辑对话框** (Condition Editing Dialog):
 - 节点信息: Node_9/Cond/Cond_1
 - 节点名称: 是否意外?
 - 意外处理? 条件处理方法选择 [Dropdown]
 - 如果: 前一个节点处理产生意外
 - 是真的: 执行 Node_8
 - 是假的: 执行 Node_6
 - Buttons: 确认, 放弃
- 开始节点编辑对话框** (Start Node Editing Dialog):
 - 节点类型: StartType
 - 是否有变量作为输入?
 - 变量类型: [Dropdown]
 - 变量ID: [Dropdown]
 - 是否有变量作为输出?
 - 变量类型: [Dropdown]
 - 变量ID: [Dropdown]
 - 意外时是否终止运行?
 - Buttons: 确认, 放弃

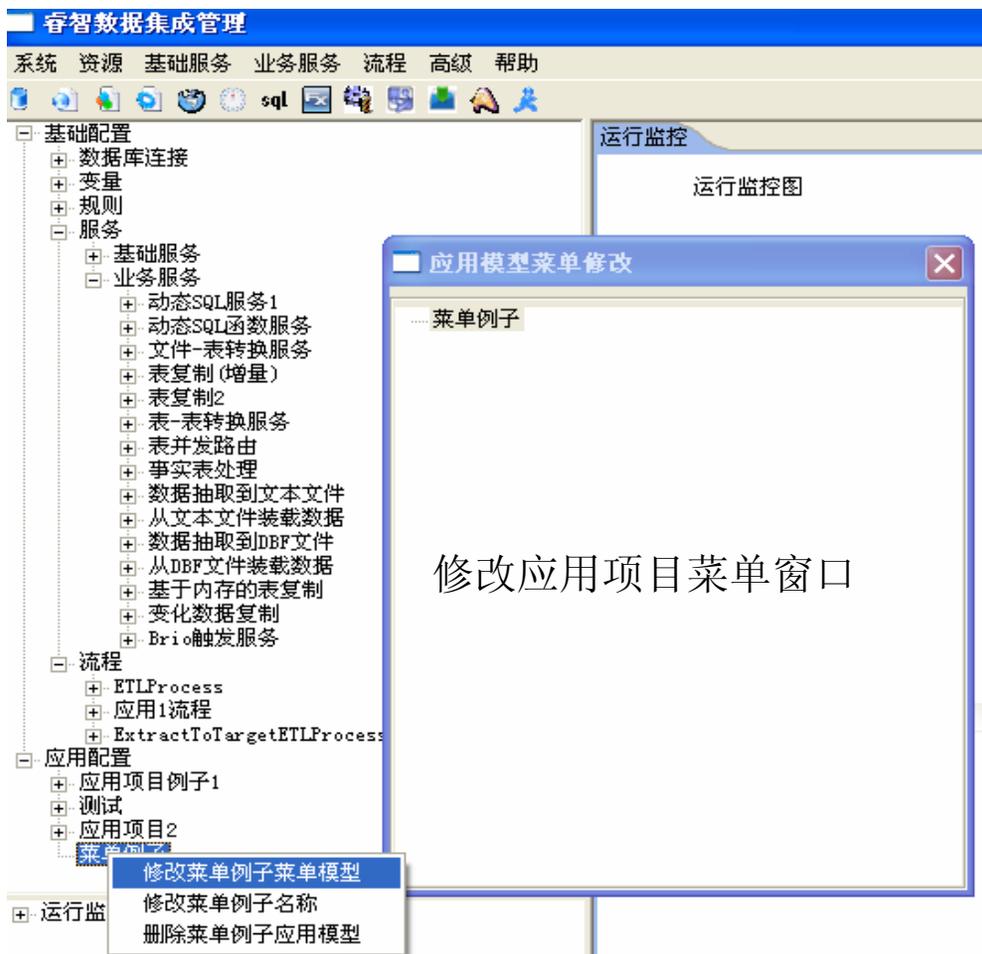
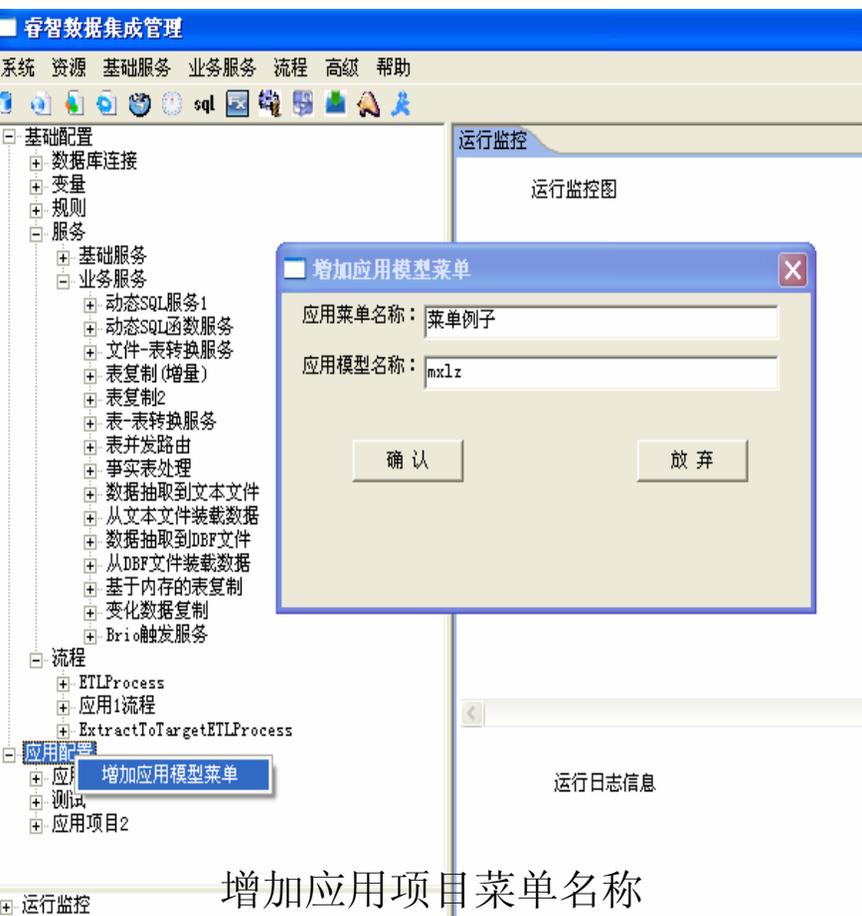
Text annotations in the image:

- When selected, it indicates that when an error occurs in the previous processing node, the corresponding error handling is performed.
- When selected, it indicates that when an error occurs in the processing node, the entire workflow is terminated. Otherwise, according to the error handling of the condition node, the workflow starts.

应用工程项目创建(1)

通过下操作可以增加应用工程项目的主菜单：

1. 增加应用项目的菜单名称和应用模型的名称
2. 修改该菜单，此时将弹出应用菜单的修改窗口



应用工程项目创建(2)

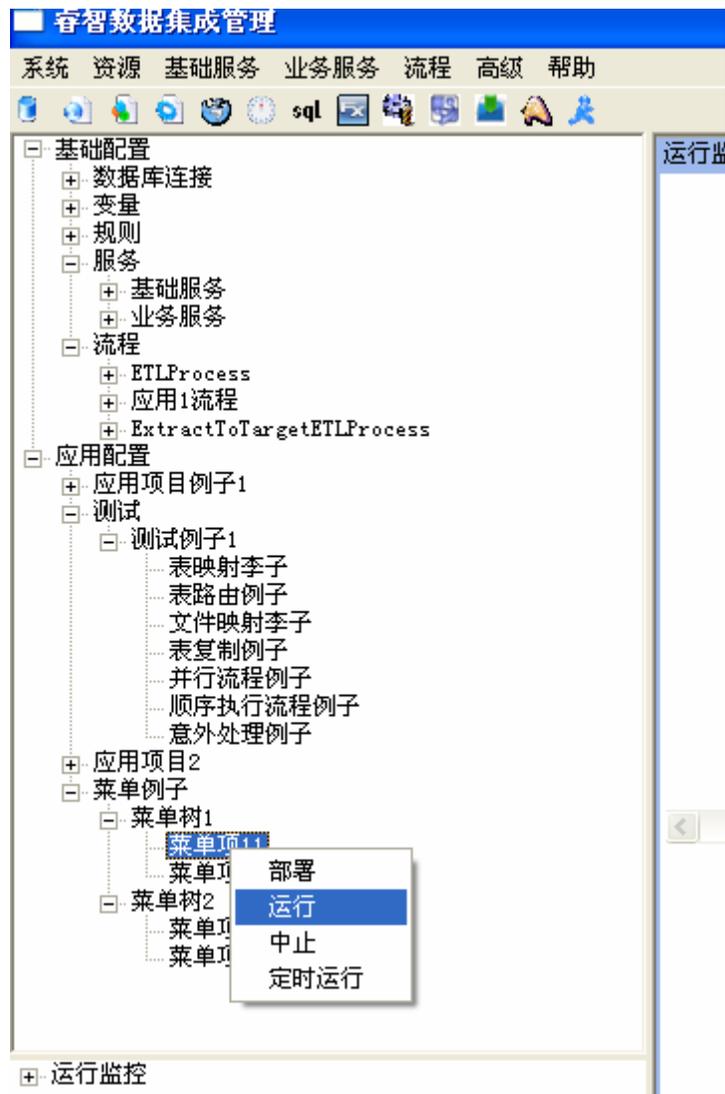
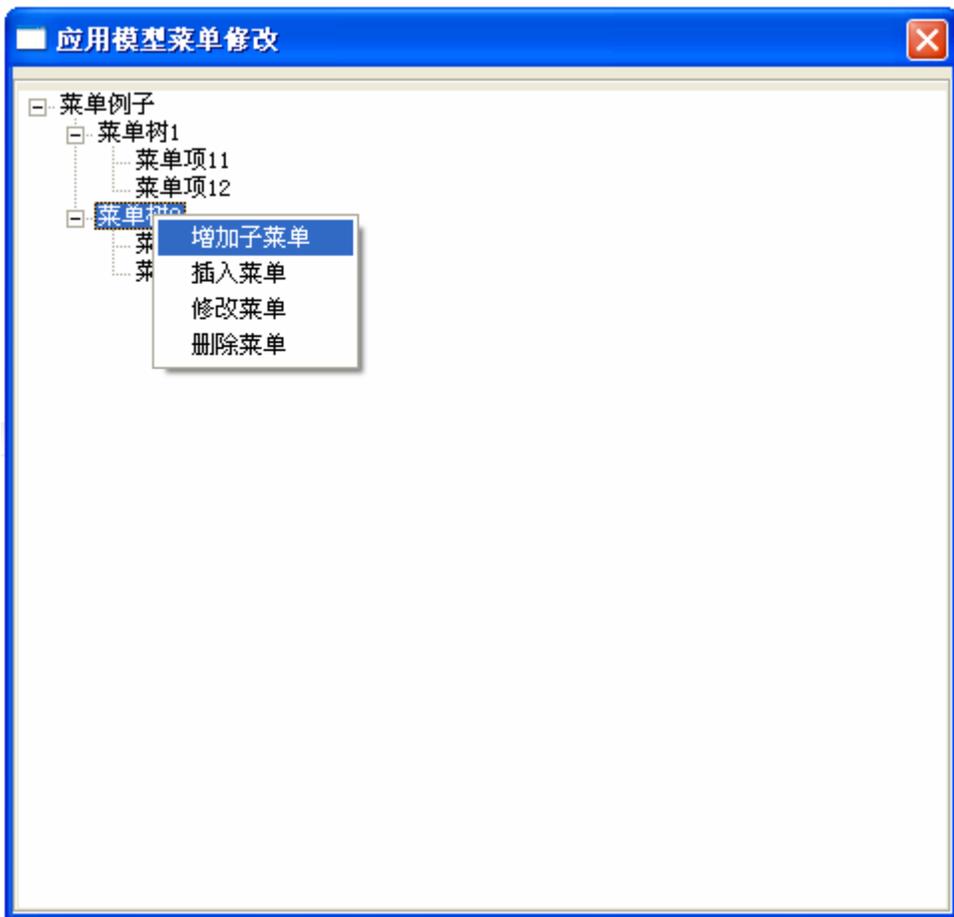
修改应用项目菜单：

在修改应用项目菜单窗口内可以增加子菜单树，也可以增加子菜单。



应用工程项目创建(3)

继续完善菜单项，完成后如下所示，可以被部署、被运行、被终止、也可以被定时运行

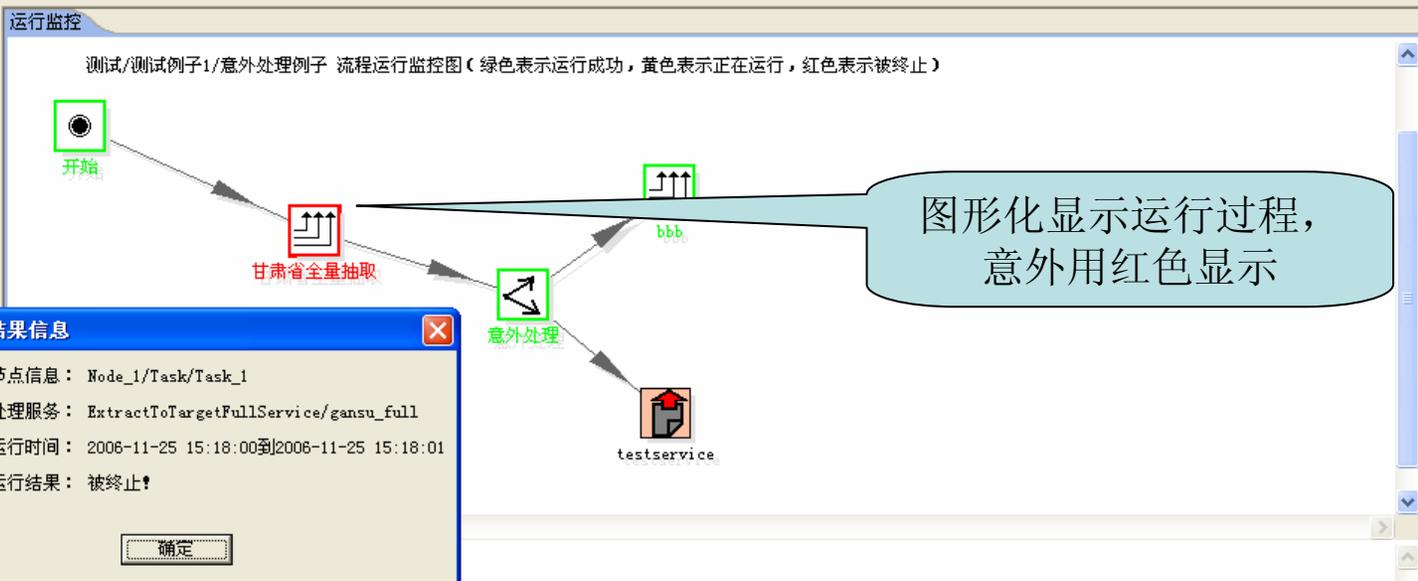


系统运行监控(2)

睿智数据集成管理

系统 资源 基础服务 业务服务 流程 高级 帮助

- 基础配置
 - 数据库连接
 - 变量
 - 规则
 - 服务
 - 基础服务
 - 业务服务
 - 流程
 - ETLProcess
 - 应用1流程
 - ExtractToTargetETLProcess
- 应用配置
 - 应用项目例子1
 - 测试
 - 测试例子1
 - 表映射例子
 - 表路由例子
 - 文件映射例子
 - 表复制例子
 - 并行流程例子
 - 顺序执行流程例子
 - 意外处理例子
 - 应用项目2
 - 菜单例子
 - 菜单树1
 - 菜单项11
 - 菜单项12
 - 菜单树2
 - 菜单项21
 - 菜单项22



图形化显示运行过程, 意外用红色显示

节点运行结果信息

节点信息: Node_1/Task/Task_1

处理服务: ExtractToTargetFullService/gansu_full

运行时间: 2006-11-25 15:18:00到2006-11-25 15:18:01

运行结果: 被终止!

确定

```
2006-11-25 15:18:00 {Service/ExtractToTargetFullService/gansu_full/0} process begin
2006-11-25 15:18:00 {Service/ExtractToTargetFullService/gansu_full/0} process Table:DJ_NSRXX@local_oracle_source
2006-11-25 15:18:01 {Service/ExtractToTargetFullService/gansu_full/0} exception: 意外: ORA-00942: 表或视图不存在
2006-11-25 15:18:01 {Service/ExtractToTargetFullService/gansu_full/0} exception: 意外: ORA-00942: 表或视图不存在
2006-11-25 15:18:01 {Service/ExtractToTargetFullService/bbb/0} process begin
2006-11-25 15:18:01 {Service/ExtractToTargetFullService/bbb/0} process Table:CSBGTT@local_oracle_db
2006-11-25 15:18:01 {Service/MemoryCopyTableService/MemoryCopyTableService/0} process Table:CSBGTT@local_oracle_db
2006-11-25 15:18:01 {Service/ExtractToTargetFullService/bbb/0} finished process
2006-11-25 15:18:01 {Service/ExtractToTargetFullService/bbb/0} process end
```

显示运行情况明细

二次开发

- 二次开发
 - 如何引入特殊格式文件数据？
 - 如何扩展特殊数据转换/清洗处理？
 - 如何增加转换变量？
 - 如何增加服务？
 - 如何**OEM**？
- 详细信息参见 《二次开发部分》