

# ReportStudio 用户手册

(草稿)

上海安硕信息技术有限公司 二〇〇五年一月



# 修订历史纪录

日期	版本	作者	说明
2005-2-14	草稿	zllin@amarsoft.com	
2005-2-4	1.0.2	zllin@amarsoft.com	Fro als6



# 目 录

1	引音	1
1.1	编写目的	1
1.2	定义	1
1.3	参考资料	1
2	概念及体系结构	2
3	REPORTSTUDIO 的配置	3
3.1	概述	3
3.2	ORACLE 配置	3
3.3	Informix 配置	5
3.4	DB2 UDB 配置	8
3.5	SYBASE SQLSERVER 配置	10
4	启动、运行 REPORTSTUDIO	11
4.1	运行要求	11
4.2	环境搭建或恢复	11
4.3	工作区简述	12
5	导出、导入操作	14
5.1	概述	14
5.2	导出	14
5.2.	1 非全量导出	15
5.2.	2 全量导出	20
5.3	导入	21
5.3.	1 非全量导入	21
5.3.	2 全量导入	22
6	REPORTSTUDIO 定义说明	23
6.1	参数设置	23



6.1.1	概述	. 23
6.1.2	目录树定义	. 23
6.1.3	数据表定义	. 25
6.1.4	字段定义	. 26
6.1.5	代码定义	. 27
6.1.6	关键字定义	. 28
6.1.7	代码转换定义	. 29
6.1.8	函数说明	. 30
6.1.9	属性设置	. 31
6.1.10	中间表结构	. 32
6.2 排	B表设置	. 33
6.2.1	概述	. 33
6.2.2	报表清单	. 33
6.2.3	报表元素	. 35
6.2.4	报表元素关联	. 37
6.2.5	模板列表	. 38
7 报表	竞开发指南	. 39
7.1 指	f南一	. 41
7.1.1	报表表样	. 41
7.1.2	报表分析	. 41
7.1.3	过程	. 42
7.2 指	f南二	. 44
7.2.1	报表表样	. 44
7.2.2	报表分析	. 44
7.2.3	过程	. 44
7.3 排	f南三	. 46
7.3.1	报表表样	. 46
7.3.2	报表分析	. 46
7.3.3	过程	. 46



# 1 引言

# 1.1 编写目的

为了能更好地说明报表开发解析环境 ARS-4E(AmarSoft Report System - 4Engine)及其开发工具 ReportStudio 的具体应用,本用户手册意在为报表开发人员和报表维护人员提供一份可供参考的技术文档,同时也为前期的培训移交工作提供参考。

# 1.2 定义

为了避免混淆,对于一些在本文中提及到的特定称谓,作出一下说明:

- "恢复":将 ReportStudio 备份本地数据文件中的数据复制到目标数据库中。
- "备份": 将服务器中的数据复制备份到本地备份数据文件中。
- "本机":运行报表设置工具 ReportStudio 的计算机。
- "服务器":数据库运行所在的服务器计算机。
- "本地表": 本地数据文件中的数据表。
- "服务器表":数据服务器中的数据表。
- "脚本": 由 ReportStudio 导出生成的. sql 执行脚本。
- "文本": 由 ReportStudio 导出生成的文本文件。

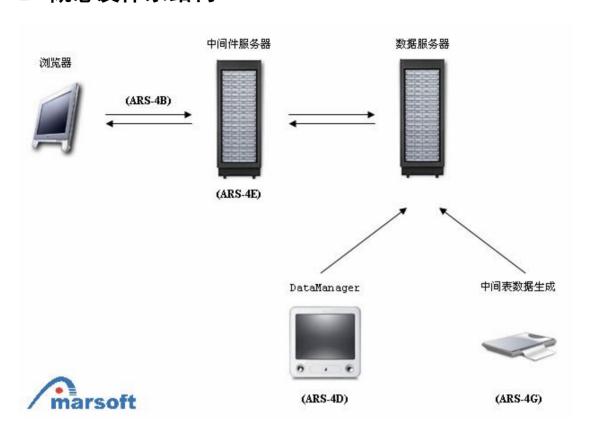
# 1.3 参考资料

配合本《开发手册》,可参考一下技术文档:

- 1、《报表部署文档》
- 2、《报表平台设计概要》
- 3、《展示页面调整参考》
- 4、《报表开发参考手册》



# 2 概念及体系结构



简单地说:客户通过浏览器提交统计条件,(ARS-4E)分析解析通过(ARS-4B)接受的参数,运行由(ARS-4E)设置的报表元素,返回在通过(ARS-4B)生成的中间表基础上复合统计后的结果。

展示页面 (ARS-4B): 接受客户通过浏览器提交的统计条件,并讲改条件下的统计查询结果返回展示在浏览器上;

报表引擎 (ARS-4E): 通过查询条件确定同一报表模板对应下的所有报表元素,逐个解析各个报表元素,将结果以二维数组的形式存放,最后根据设定的 OWC 位置生成 javascipt 脚本,并储存在结果数据文件中。对于相同条件下的查询,除了第一次查询是 与数据库直接交互外,其余都是从数据文件中读取数据;

ReportStudio (ARS-4D): ReportStudio 作为整个报表运行环境中重要组成部分,是报表制作的重要工具。通过 ReportStudio 设定报表的基本属性、报表模板及关联的报表元素等。经过分析后的一张报表通常拆分为一个或多个元素,通过提供的各种方法和定义实现各个元素的取值和相互间的关联。

中间表生成程序(ARS-4G):报表所使用核心中间表的数据生成程序。



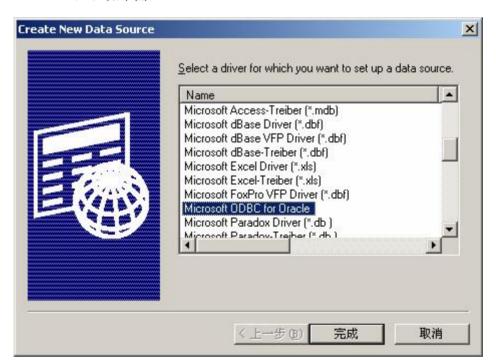
# 3 ReportStudio 的配置

# 3.1 概述

为了减少不同数据库连接方式不同带来的配置繁琐,ReportStudio 通过 ODBC 与数据库建立连接,以下各个数据库配置说明都建立在数据库客户端已经能成功连接数据库服务器的基础上。

#### 3.2 Oracle 配置

1、 在 Oracle ODBC 驱动的选择上,为了减少配置参数的繁琐,推荐使用了 MS 提供的 Oracle ODBC 驱动,而不选用 Oracle 自身提供的"Oracle ODBC Driver",如下图:



2、 如下图分别填写了"数据源名称"、"用户名"和"数据服务名"确定之后,由于自身不提供测试连接,需要通过第三方工具如 PB 等测试 ODBC 连接可用。





3、 ODBC 数据源配置完成后,修改 ReportStudio 自身的数据库连接配置文件 server.dsn 即可连接到数据库服务器上。

相应 server.dsn 文件

```
[CONFIG]

DBType=Oracle

[ODBC]

DRIVER = Microsoft ODBC for Oracle

UID = report

SERVER = bank
PWD = bank
```

(在连接成功但因为数据库中没有相应的报表系统表造成的提示处理说明见 『4.2 环境恢复』的统一说明。)



# 3.3 Informix 配置

1、因为不同的 Informix 版本对应的 ODBC 驱动和驱动名都不一样,因此需要根据 具体的情况选相应的驱动信息。

```
Informix 7.31 对应驱动名: INFORMIX x.xx 32 BIT Informix 9.21 对应驱动名: INFORMIX 3.31 32 BIT Informix 9.40 对应驱动名: IBM INFORMIX 3.43 32 BIT
```

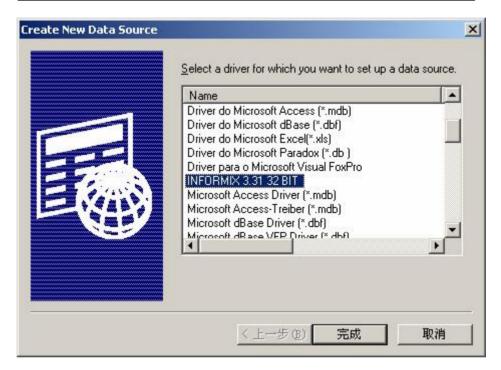


图 3.3-1

2、在"Informix ODBC Driver Setup"对话框中,实现通过 ODBC 连接数据库,只需要对"General"和"Connection"两个选项页作相应的配置即可。

"General"选项页: (见图 3.3-2)

"Data Source Name": 数据源名称

"Description": 本数据源描述

"Connection"选项页: (见图 3.3-3)

本选项页设置了连接对应数据库服务器所需的基本信息。

ServerName: Informix 数据库服务器名;

HostName: 远程数据库服务器名或 IP 地址;

Service: 服务名 (安装 Informix 时默认为 Turbo);

Protocol: 连接远程数据库服务器所用的网络协议;



DataBase Name: 所连接的 Informix 数据库名,通过下拉框可选择可用的数据库名:

User Id: 登陆用户名;

PassWord: 登陆密码。

配置结束后通过"Apply&Test Connection"按钮测试 ODBC 配置连接是否成功。



图 3.3-2



图 3.3-3





图 3.3-4

3、步骤2配置相应的 server.dsn 配置如下:

#### server.dsn 文件

```
[CONFIG]
DBType=Informix

[ODBC]
DRIVER = INFORMIX 3.31 32 BIT
UID = informix
PRO = onsoctcp
SERV = 1526
SRVR = ol_develope
```

(在连接成功但因为数据库中没有相应的报表系统表造成的提示处理说明见 『4.2 环境恢复』的统一说明。)



# 3.4 DB2 UDB 配置

- 1、先通过"配置助手"配置连通远程数据库服务器之后,为数据库实例起一个别名(DB2通过别名连接数据库实例),并测试通过 ODBC 方式连接成功。
- 2、在ODBC配置驱动可以看到DB2的驱动名称



3、确定之后出现选择数据库实例别名的对话框(这里假设为 Formal)



4、在弹出的"CLI/ODBC设置"对话框中输入数据库实例登陆 ID 和密码。





5、选择后测试所建 ODBC 连接成功与否



6、步骤2配置相应的 server.dsn 配置如下:

#### server.dsn 文件



7、

(在连接成功但因为数据库中没有相应的报表系统表造成的提示处理说明见 『4.2 环境恢复』的统一说明。)



# 3.5 Sybase SQLServer 配置

1,

2、

3、步骤 2 配置相应的 server.dsn 配置如下:

server.dsn 文件

[ODBC]			

4、

(在连接成功但因为数据库中没有相应的报表系统表造成的提示处理说明见 『4.2 环境恢复』的统一说明。)



# 4 启动、运行 ReportStudio

# 4.1 运行要求

『最低配置』	
操作系统	Microsoft Windows 98 或更高
可用内存	32 MB 或更高
可用硬盘空间	20 MB
『推荐配置』	
操作系统	Microsoft Windows 2000 或更高
可用内存	64 MB 或更高
可用硬盘空间	32 MB 或更高

# 4.2 环境搭建或恢复

针对没有报表开发环境的数据库服务器(或第一次运行 ReportStudio 的目标数据 库),存在一个环境的搭建过程,其过程是在数据库服务器中建立一些 ReportStudio 和 报表引擎需要的数据表,列表如下:

表名	表标题	备注
R_TABLE_INFO	数据表定义	
R_FIELD_INFO	表字段定义	
R_TREE_INFO	目录树定义	
R_CODE_INFO	系统代码定义	
R_KEY_INFO	关键字定义表	
R_CONVERT_INFO	代码转换	
R_FUNCTION_INFO	FUNCTIONS 说明	
R_PROPERTY_INFO	报表定义属性	
R_SHEET_ITEM	报表元素定义	
R_SHEET_MODEL	报表模板定义	
R_SHEET_LINK	报表元素关联定义	
R_FACT_STRUCTURE	中间表结构	

在缺省情况下,ReportStudio 连接的目标数据文件为本地的 Access,程序运行后可 以看到右下脚的状态栏中"数据库类型"显式为 Access, 如 数据库类型:Access 右图。

经过【3.ReportStudio 的配置】配置了相应的目标数据库 ODBC 连接之后运行程序, 如果成功连接上数据库会在状态栏中的"数据库类型"项看到相应的数据库类型,如右

(数据库类型:DB2)

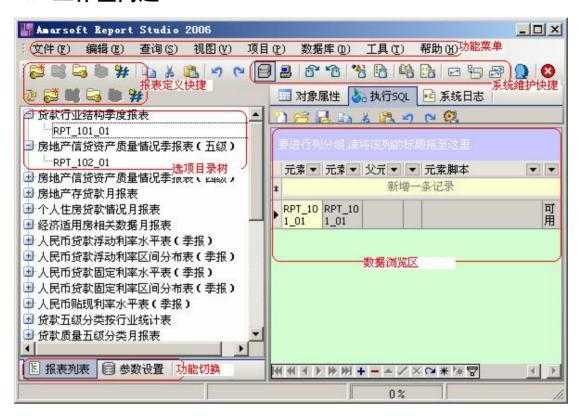
【提示】存在客户端防火墙的情况下请确定 ODBC 数据源在允许访问外部资源条件下连接外部数



据库。

在确认远程数据库连接成功之后,将报表平台的开发环境恢复到目标数据库中,相关内容参考章节【5.3.2 全量导入】

# 4.3 工作区简述



ReportStudio 的工作区中分为"功能菜单"、"快捷菜单"、"选项目录树"和"数据浏览区",如上图。

"系统维护快捷"列出了 ReportStudio 中非常便利的导入导出功能。

快捷	快捷说明	备注
©⁴	恢复本地到服务器	参看【5.3.2 全量导入】
*8	备份服务器到本地	参看【5.2.2 全量导出】
*6	选择恢复本地到服务器	参看【5.3.1 非全量导入】
B	选择备份服务器到本地	参看【5.2.1 非全量导出】
	根据代码生成转换定义	根据输入的代码名称自动从信贷代 码表中提取相应的代码项到报表代 码转换定义中
	检查并补全中间表结构定义	自动对比中间表的结构和报表中设 置的中间表定义之间的差异,并在



	日志中给予提示,同时补齐定义中 缺失字段
保存当前数据到脚本	参看【5.2.1 非全量导出】
保存当前数据到文本	参看【5.2.1 非全量导出】
导出所有数据到脚本	参看【5.2.1 非全量导出】

"报表定义快捷"列出了在开发定义报表过程中经常使用到的几种操作。

快捷	快捷说明	备注
<b>2</b>	新增报表	参看【7 开发指南】
	新增报表元素	参看【7 开发指南】
	新增报表子元素	参看【7 开发指南】
	删除报表	参看【7 开发指南】
*	删除报表元素	参看【7 开发指南】

<sup>&</sup>quot;选项目录树":

配置 ReportStudio 自身和报表相关的功能入口。

#### "数据浏览区":

浏览展示结果数据。



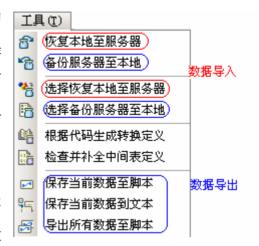
# 5 导出、导入操作

# 5.1 概述

为了满足在不同的数据库服务器之间的 数据转移、恢复或者更新,ReportStudio 提 供全量和非全量的导入导出功能。根据需要可 定制数据全量一次性或者一(多)张表的导入 和导出。

#### 【提醒】

在 ReportStudio 导入导出的过程中,需要使用到本地备份 Access 数据库,其通过



Local.dsn 文件配置。(通过 DBQ 指定备份 Access 库的文件名)。当 Local.dsn 中的 DBQ 名更改后,需要同步更改 Access 备份库的文件名,否则回报"无法连接数据库"的错误,虽仍然能使用,但不能做任何与备份库有关的操作。

```
[CONFIG]
DBType=Access

[ODBC]
DRIVER=Microsoft Access Driver (*.mdb)
UID=admin
UserCommitSync=Yes
Threads=3
SafeTransactions=0
PageTimeout=5
MaxScanRows=8
MaxBufferSize=2048
FIL=MS Access
DriverId=25
DefaultDir=
DBQ=Local.mdb
```

# 5.2 导出

- 一般如下情况下使用导出功能:
- 1、全量备份数据以便数据转移或搭建到另一个数据库中;
- 2、导出部分存在新改动的表恢复在另一数据库中实现数据的同步。



# 5.2.1 非全量导出

非全量导出分为非当前数据和当前数据。(图 5.1-1)

非当前数据:通过选择数据表的形式决定导出的数据/数据表。包括"备份数据表 至本地"和"导出所有数据至本地";

当前数据:导出保存的数据为在"数据浏览区"中展示的数据。包括"保存当前数 据至脚本"和"保存当前数据到文本"。

选择备份服务器至本地

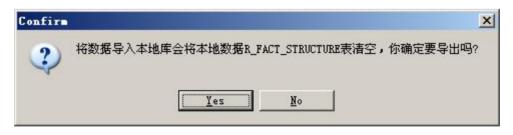
目的:将数据服务器中的指定表中数据导出到本地备份库中。

操作:

"工具"□"选择备份服务器至本地",出现如下图所示的对话框,



对话框中枚举出了所有相关的系统表,选上需要备份的数据表后确定,提示操作将 清空该表的本地数据, 如下图所示



检查无误后确定导出,成功操作看到如下图所示



至此,指定单表/多表导出结束。



# 导出所有数据至脚本

导出的数据将以 sql 执行脚本的形式储存在.sql 文件中,其中包括建表语句、数据库插入 Insert 语句等。

快捷方式 剪 "工具" ➡ " 导出所有数据至脚本",出现如下图所示对话框



输入了.sql 文件名后,点击"保存"后开始导出到 sql 脚本操作,进度通过 ReportStudio 下端的状态栏显示。导出终结后提示如下



至此,导出全部数据到 sql 脚本的操作完成。

得到的结果是一个.sql 文件, 其内容结构似如下图





# 保存当前数据至脚本

将"数据浏览区"中展示的数据以sql 脚本的形式导出到.sql 文件中。

通过快捷方式 ■或"工具" ➡" 保存当前数据至脚本"。



输入了.sql 文件名后,点击"保存"后开始导出到 sql 脚本操作,进度通过 ReportStudio 下端的状态栏显示。导出终结后提示如下



至此,导出当前数据到 sql 脚本的操作完成。

需要注意的是:文件名的命名将被直接用来当作 sql 语句中的 TABLE 名,因此.sql 的命名建议以数据展示的表名命名。

如上图中以"savecurrenttosql.sql"命名的文件后,得出的文件内容结构如下图所示:

```
INSERT INTO savecurrenttosql.sql VALUES ('R_CODE_INFO', 'Access', 'R_INSERT INTO savecurrenttosql.sql VALUES ('R_CODE_INFO', 'DB2', 'R_CODE_INSERT INTO savecurrenttosql.sql VALUES ('R_CODE_INFO', 'Informix', 'FINSERT INTO savecurrenttosql.sql VALUES ('R_CODE_INFO', 'Oracle', 'R_CINSERT INTO savecurrenttosql.sql VALUES ('R_CODE_INFO', 'SQL Server', INSERT INTO savecurrenttosql.sql VALUES ('R_CONVERT_INFO', 'Access', 'INSERT INTO savecurrenttosql.sql VALUES ('R_CONVERT_INFO', 'DB2', 'R_CINSERT INTO savecurrenttosql.sql VALUES ('R_CONVERT_INFO', 'Informix' INSERT INTO savecurrenttosql.sql VALUES ('R_CONVERT_INFO', 'Informix' INSERT INTO savecurrenttosql.sql VALUES ('R_CONVERT_INFO', 'Oracle', 'INSERT INTO savecurrenttosql sql VALUES ('R_CONVERT_INFO', 'Oracle', 'INSERT INTO save
```

可以看到标明为"savecurrenttosql.sql",而不是真正的表名。



# **日本** 保存当前数据到文本

将"数据浏览区"中展示的数据以分隔符文本的形式导出到文本文件中。

通过快捷方式 5 或 "工具" ⇒ "保存当前数据至文本"。



输入了.txt 文件名后,点击"保存"后开始导出到文本操作,进度通过 ReportStudio 下端的状态栏显示。导出终结后提示如下



至此,导出当前数据到文本的操作完成。

导出的.txt 文件以"|"作为分隔符分割各个字段的值,内容格式如下图所示:



# 5.2.2 全量导出

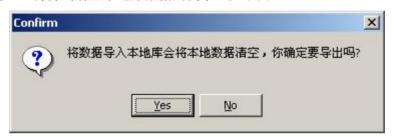
"全量导出"目前只包括"备份服务器至本地"。



# 备份服务器至本地

全量导出将导出所有的数据,包括数据表结构和数据。其操作步骤等同与将数据导 成 sql 脚本,再执行脚本的过程。也是经常使用的数据全量转移的操作。

在更新本地库之前会清空本地库中的原有数据,通过 □ 或 "工具" □ " 备份服务 器至本地"进入,将得到清空本地库数据的提示,如下图:



在导出的过程中,可以通过底部状态栏观察进度,其中包括正在导出的表和导出表 中的数据数。

在所有表无差错导出结束后,得到导出成功的提示,如下图。



至此,导出全量数据到本地库的操作完成。

可以查看本地库中的数据和导出数据库中的数据是同步的。



# 5.3 导入

### 5.3.1 非全量导入

"非全量导入"目前只包括"恢复本地到服务器"。



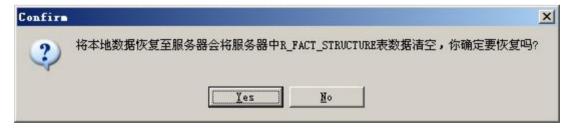
#### 选择恢复本地到服务器

只恢复选定的表到服务器中,是『备份数据表至本地』的逆过程。

通过"工具"□"选择恢复本地到服务器",出现如下图所示的对话框,



对话框中枚举出了所有相关的系统表,选上需要恢复的数据表后确定,提示操作将 清空该表的服务器数据,如下图所示



检查无误后确定导出,成功操作看到如下图所示



至此,指定单表/多表导入结束。



# 5.3.2 全量导入

"全量导入"目前只包括"恢复本地数据到服务器"。



#### 恢复本地到服务器

全量导入将备份库中的所有数据导入到数据库服务器中,包括数据表结构和数据。 其操作步骤等同与将数据导成 sql 脚本,再执行脚本的过程。也是经常使用的数据全量 转移的操作。

是『备份服务器至本地』的逆操作。

在更新本地库之前会清空数据库中的原有数据,通过 "工具" "恢复本地数据到服务器"进入,将得到清空服务器数据的提示,如下图:



如同全量导出的过程,同样可以通过底部状态栏观察进度,其中包括正在导入的表和导入表中的数据数。

在所有表无差错导入结束后,得到导入成功的提示,如下图。



至此, 导入全量数据到数据库服务器的操作完成。



# 6 ReportStudio 定义说明

# 6.1 参数设置

# 6.1.1 概述

# 6.1.2 目录树定义

通过"参数设置"□"目录树设置",可在数据浏览区看到如下图的界面。

编号	▼	类型	•	标题    ▼	目标    ▼	父编号	▼	属性
0001		节点		参数设置				
00010001		表名	类型	录树定义	R_TREE_INFO	0001		
00010002		表名		表定义	R_TABLE_INFO	0001		
00010003		表名		字段定义	R_FIELD_INFO	0001		
00010004		表名		函数定义	R_FUNCTION_INFO	0001		
00010005		表名		代码定义	R_CODE_INFO	0001		
00010006		表名		代码转换定义	R_CONVERT_INFO	0001 3	结点编号。	
00010007		表名		关键字定义	R_KEY_INFO	0001		
00010008		表名		属性定义	R_PROPERTY_INFO	0001		
0002 节	点编号	节点		报表设置				
00020001		表名		中间表结构定义	R_FACT_STRUCTURE	0002		
00020002		表名		报表模板	R_SHEET_MODEL	0002		
00020003		表名		报表元素	R_SHEET_ITEM	0002		
00020004		表名		报表关联	R_SHEET_LINK	0002		

其中的6个字段含义如下:

字段名	含义		备注
编号	该节点自身编号(唯一、不重复)		
类型	该节点所属类型	Node	节点方式
		SQL	SQL方式
		Table	数据表方式
标题	该节点显示标题		
目标	该节点二级数据源(表、数据集、)		
父编号	该节点所属父节点		
属性	该节点一级数据源		

#### 【说明】

- 1、"Node"节点方式: 用于构建根目录节点,如"父节点"。父节点只有"节点编号",既没有"数据表",也没有"父节点编号";
- 2、"Table"数据表方式:"更新表"中为目标表,在目录树中双击该节点,在数据浏览区中将打开该表中的所有数据,同一父节点下的各节点"节点编号"具有相同的"父节点编号";
- 3、"SQL"脚本方式: 以"节点属性"中 SQL 脚本结果作为节点数据展示内容,



"更新表"中仍为目标表。



# 6.1.3 数据表定义

通过"参数设置"□"数据表定义",可在数据浏览区看到如下图的界面。



在"数据表定义"中,定义了ARS环境平台所涉及的所有数据表定义脚本。

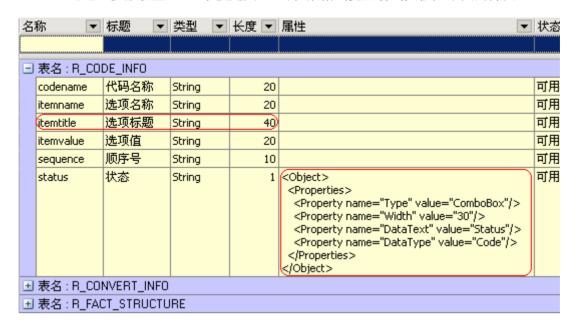
#### 【说明】

在全量导入或者环境构建时,通过这些表定义 SQL 脚本生成相应的表结构。



#### 6.1.4 字段定义

通过"参数设置"□"字段定义",可在数据浏览区看到类似如下图的界面。



通过"字段定义"定义设置在数据浏览区中对应表字段的中文名称及其他属性。

#### 【说明】

- 1、"字段名"与"字段标题"建立展示标题映射。当显示 R\_CODE\_INFO 表中的 itemtitle 时将以"选项标题"替代,当缺少相应映射时,在数据浏览区字 段标题栏将显示原始字段名。
- 2、"字段长度"控制显示的字段宽度,以获得较好的显示效果。

字段类型	含义	备注
String	字符型	
Number	数字型	
Code	代码型	调用在"系统代码"中设置下拉框代码,
		如图中的"FieldType"
		String ▼ Code 代码 Numb 数字 String 字符

4、



# 6.1.5 代码定义

通过"参数设置"□"代码定义",可在数据浏览区看到类似如下图的界面。



通过"系统代码定义"设定下拉框代码。

在下拉框中显示"元素值"和"元素标题"(图中①),返回值为"元素值"(图中②),通过顺序号(图中③)强制排定显式顺序,而只有状态为"可用"(图中④)的代码定义才有效。



# 6.1.6 关键字定义

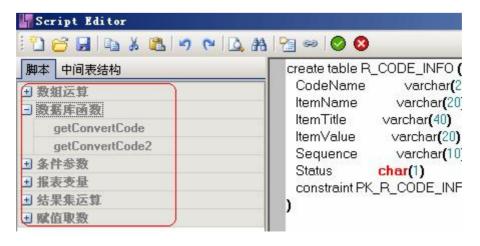
通过"参数设置"□"关键字定义",可在数据浏览区看到类似如下图的界面。



通过"关键字定义"定义设定在报表元素编辑环境中的操作按钮对应信息,通过指定"类型"分类关键字。

"关键字"中的具体用法参看《报表开发参考手册》

关键字类型	含义	备注说明
ClientProperty	客户端属性	
DatabaseFunction	数据库函数	
ReportCondition	条件参数	
ArrayFunction	数组运算	
ReportParam	报表变量	
ServerProperty	服务器端属性	
SetFunction	赋值取数	
ResultFunction	结果集运算	





# 6.1.7 代码转换定义

通过"参数设置"□"代码转换定义",可在数据浏览区看到类似如下图的界面。

输	入值1 ▼	· 输入值2 ▼	返回值      ▼	状态 ▼	
□ 代码名称 : OverdueTerm					
	0	90	1	可用	
	91	180	2	可用	
	181	270	3	可用	
	271	365	4	可用	

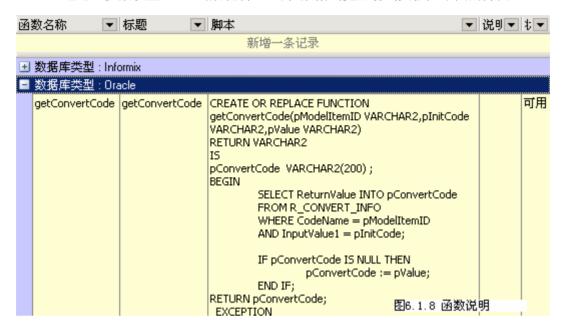
图 6.1.7 代码转换定义

通过"代码转换定义"设定代码映射关系,通过 getConvertCode 和 getConvertCode2 这两个 DBFunction 来调用实现代码转换,参看《报表开发参考手册》中的"getConvertCode"部分。



# 6.1.8 函数说明

通过"参数设置"□"函数说明",可在数据浏览区看到类似如下图的界面。



列举说明在 ARS 平台中提供的 SQL 语句脚本中可以使用的 DBFunctions,内容包括方法用法,示例等。



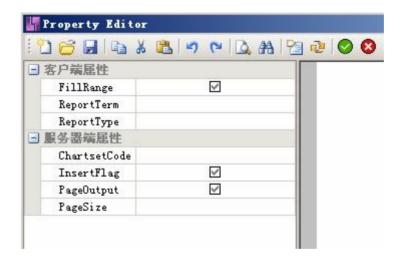
# 6.1.9 属性设置

根据设定的属性类型,在编辑相应的属性时提供设定的属性快捷。



在编辑报表属性的编辑框中出现的是报表属性设定(SheetProperty)设定的属性内

#### 容,如下图:





# 6.1.10 中间表结构

通过"中间表结构"配置中间表字段与管理系统表字段的映射关系,中间表生成程序根据这些映射组合构建中间表取数 SOL 语句。



详细内容参看《中间表生成程序说明》



# 6.2 报表设置

## 6.2.1 概述

操作对象:元素(合成)数据集

思路:通过将报表模板分解成多个部分,称之为元素,设置相关"操作"和"定义"操作报表元素来返回我们需要的格式,最后将所有元素重新组合在一起通过展示页面展示查询结果。

主要包括 4 部分: "报表清单"、"报表元素"、"报表关联"和"模板列表"。

### 6.2.2 报表清单

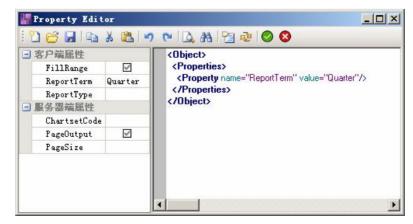
"报表清单"列举出所有报表的模板号、名称及一些脚本控制项。声明基本报表级 属性和访问权限。

編号 ▼	标题    ▼	属性       ▼	Į.	顺▼	描述			
	新增一条记录							
RPT_201	贷款五级分类按行业统计表			201	javascript:top.doAction("RP T_201")			
RPT_202	贷款质量五级分类月报表			202	javascript:top.doAction("RP T_202")			
RPT_203	贷款质量迁徙情况表			203	javascript:top.doAction("RP T_203")			
RPT_204	贷款按逾期时间分类情况表	<object> <pre></pre> <pre></pre> <pre></pre> <pre></pre> <pre> <pre>Tem</pre> <pre>value="Quarter"/&gt; </pre> <pre></pre> <pre>&lt;</pre></pre></object>		204	javascript:top.doAction("RP T_204")			
RPT_205	最大十家贷款及贷款质量情况表			205	javascript:top.doAction("RP T_205")			
RPT_206	最大十家集团客户授信情况表			206	javascript:top.doAction("RP T_206")			
RPT_207	最大二十家股东贷款及授信情况表	图 6.2.2 报表模板洁单		207	javascript:top.doAction("RP T_207")			

『模板编号』标识报表模板自身的唯一编号。通过"模板编号"实现与报表元素一对多的关联关系。

『模板标题』显示在展示页面的报表模板名称。

『模板属性』设置报表级别的属性,包括报表类型和上报频度等。





#### 客户端属性

- 1、报表类型 (ReportType):
  - a、Fixed: 固定型报表(默认值)
  - b、V List: 纵向清单型报表
  - c、 H List: 横向清单型报表
- 2、上报时间频度 (ReportTerm):
  - a、Year: 年报
  - b、HalfYear: 半年报
  - c、Quarter: 季报
  - d、Month: 月报(默认值)
  - e、Day: 日报

#### 服务器端属性

- 1、编码转换(ChartSetCode)
  - a) ARS CHARTSET GBK = 1
  - b) ARS\_CHARTSET\_ISO8859 = 0
  - c) ARS CHARTSET gb2312 = 2
  - d) ARS CHARTSET NONE = -1
- 2、是否分页输出(PageOut)
- 3、每分页数据量(PageSize)

『列表编号』设置展示页面次序。方法与"AmarTreeView"设置方法一致。

『描述』设置展示页面中触发的点击 JS 事件。JS 脚本中传入的参数要求与报表编号一致以触发导入的 HTML 表格式模板。

#### 『权限控制』

设置访问页面的权限级别。



# 6.2.3 报表元素

有时候一张报表无法一次把数全部取出来,则要分几次进行,每一次的取数称做本 张报表的一个元素。一张报表可以有一个或一个以上个元素。由"报表设置"□"报表 元素"或"报表列表"中选定相应的报表元素进入编辑窗口,可在数据浏览区看到类似 如下图的界面。



#### 『报表元素编号』

报表元素唯一标识。通过"元素编号"与报表模板关联。

✓ 元素命名规则:以报表模板号开始,通过下划线 "\_"和元素排序组合而成。

#### 『报表元素标题』

报表元素名称。

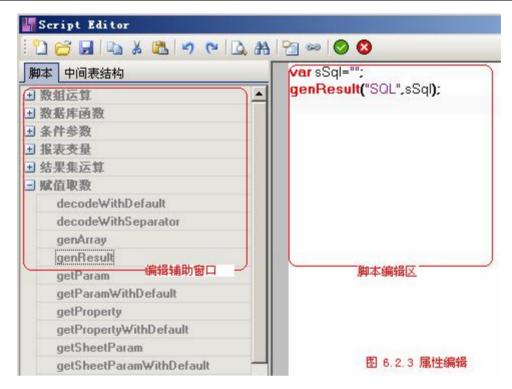
✓ 标题命名约定:元素标题体现报表元素本身实现的功能和结果,为了 便于快速识别元素自身功能。

#### 『模板元素属性』

通过"模板元素属性"实现取数、操作、定义。(操作、定义说明参看《报表开发参考手册》)。

通过双击元素属性定义浏览窗口重的"元素脚本"进入元素脚本的编辑窗口,类似如下图:





如图 6.2.3 所示,属性编辑窗口分为"属性编辑区"和"操作选择区"。

"编辑辅助窗口"以按钮形式列举出了系统平台支持的操作和方法,选择后单击相应的按钮将在编辑区中写入对应方法的注释说明和默认的操作语法,如选择 genResult,自动写下如下代码:

```
genResult("SQL","");
```

建议对一些特殊处理的操作在操作注释中遵循代码规范说明清楚,便于维护修改。



# 6.2.4 报表元素关联

一张报表模板对应多个报表元素,通过报表元素关联实现关联。做好了报表的模板写好了元素的取数,还需要把他们关联起来才能完成整个报表。

由"报表设置"□"报表元素关联",可在数据浏览区看到类似如下图的界面。

模板 △ ▼	元素    ▼	关联顺序▼	关联属! ▼	关联脚4▼	状态
		新増-	一条记录		
人民币贷款固定利率水平表(季报)	RPT_109_01				可用
人民币贷款固定利率区间分布表(季报)	RPT_110_01				可用
人民币贴现利率水平表(季报)	RPT_111_01	_			可用
贷款五级分类按行业统计表	RPT_201_01	D D			可用
贷款质量五级分类月报表	RPT_202_01				可用
贷款质量迁徙情况表	RPT_203_01	_			可用
贷款按逾期时间分类情况表	取出未到期贷款	<b>(</b> )			可用
贷款按逾期时间分类情况表	取出逾期类贷款				可用
最大十家贷款及贷款质量情况表	RPT_205_01				可用
最大十家集团客户授信情况表	RPT_206_01				可用
最大二十家股东贷款及授信情况表	RPT_207_01				可用
客户大额授信统计表	RPT_208_01				可用
贷款风险分类按机构统计表	RPT_301_01		图 6.2.4 元	素美联	可用

#### 『模板』

"报表模板"中设置的"模板标题"

#### 『元素』

"报表元素"中设置的"元素标题"

#### 『关联顺序』

强制性指定元素的执行顺序

『关联属性』

(预留扩展)

『关联脚本』

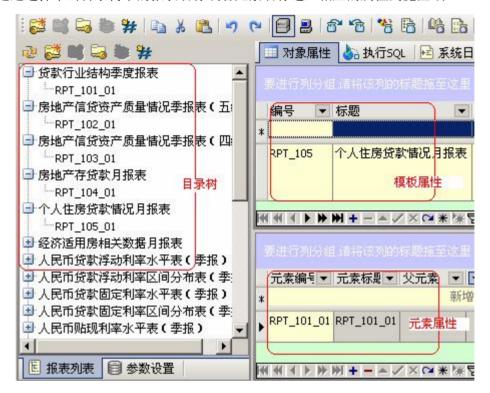
(预留扩展)



### 6.2.5 模板列表

模板列表下列举出已经设置的报表模板,双击模板号(名)(具体调整方法参看《报表开发参考手册》)过滤出所有该报表模板对应的报表元素。

通过选择单击目录树中的报表名称或者元素名称进入相应的属性浏览区域。





# 7 报表开发指南

在以下指南中,针对在报表开发过程中遇到的几个常遇到的开发方式做简要的分解分析,并对主要步骤作出代码分析。(参看《开发手册》)

注: 例样 SQL 语法以 Oracle 示例,不代表 ANSI-SQL 标准语法。

针对在开发过程中频繁使用到的一些报表元素增删改操作,对其功能按钮做以下说明:

# **新增报表**

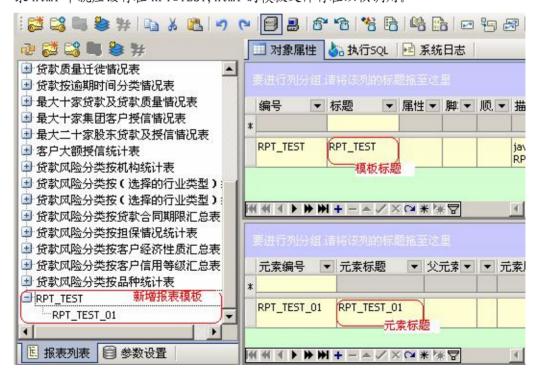
"新增报表"按钮之后弹出的模板编号确定窗口,如下图



确认编号之后可见模板列表中新增了一个报表结点 RPT\_TEST,并自动传见其项下元素 RPT TEST 001:

【提示】通过更改"模板标题"和"元素标题"可实现在列表中显式中文

【注意】模板编号必须与模板描述中指定调用的 htm 模板名称一致。如上例:模板编号为 RPT\_TEST ,模板描述为 javascript: top. doAction("RPT\_TEST"),则在模板目录 html 中就应该存在 RPT\_TEST, html 的模板文件存在以供调用。





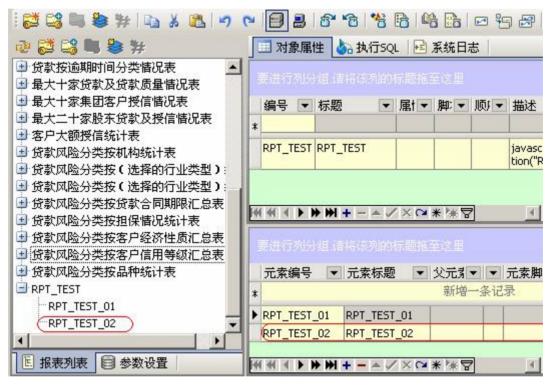
# 新增元素

倘若报表由多个元素组成,则需要新增其他的报表元素。在点击选中报表模板之后,

通过"新增元素"按钮 可出现新增元素编号确认窗口,如下图:



确认之后可以在列表中的相应模板之下看到新增的元素,如下图:



# 新增子元素

(扩展功能, 暂不用)



在确认需要删除特定报表模板之后通过"删除报表"按钮 删除报表。

【注意】该操作仅仅删除了报表定义的模板和关联,原报表下的所有元素本身没有删除。

# # 删除报表元素

在确认需要删除特定报表元素之后通过"删除元素"按钮 删除指定的元素。

密级: 第 40 页/共 40 页 Amarsoft Technology Co., Ltd.



# 7.1 指南一

『简单固定型』

由一个(或多个)独立报表元素组成,元素与元素之间没有关联关系,且表样行列固定不变的报表,属于简单固定型报表。

### 7.1.1 报表表样



# 7.1.2 报表分析

报表表样可简单拆分为一个以国标大类作为纬度、五级余额作为度量的报表元素。

#### 【要点】

- 因为有数据覆盖面可能造成国标行业分类统计数据缺失,因此需要对结果数据集作行业分类的匹配,否则会造成显示结果纬度错位。
- 2、 数据集起点(Position)为(7,2);
- 3、 在(2,1)填写"填报日期"、在(3,1)单元格填写填报单位;
- 4、 通过 ms EXCEL 实现在第 30 行的纵向求和。
- 5、 Unit 单位 作为变量传入到 sql 中实现动态的单位换算。
- 6、 InputDate 查询日期 作为变量传入实现查询时点变化。
- 7、 OrgID 查询机构 作为变量传入实现查询机构变化。



## 7.1.3 过程

- 1、 新增报表模板("报表模板编号"与"模板描述"中触发的 JS 脚本参数要一致);
- 2、"报表元素"新增报表元素
  - a) 元素编号以"模板号"和元素序号构成;
  - b) 标题描述说明本元素的核心功能;
- 3、在"元素脚本"中作如下元素定义

```
====== (BEGIN) ======
                      贷款分类汇总
/* Title:
                      通过"首次发放日"区分四级五级余额,防止重复统计。
 * Content:
* Developer:
                      <someone>
//定义 SOL 变量(JS 语法,红色标明为变量)
var sSQL= "SELECT IndustryType, COUNT (DuebillNo) AS BusCount,"
     +" SUM (NVL (Balance, 0)) / " + Unit + " AS Balance14,"
          SUM (DECODE (FirstDisburseFlag, '1', 1, 0) * NVL (ClassBalance1, 0)"
     +"
            ) / " + Unit + " AS ClassSum1,"
     +"
           SUM (DECODE (FirstDisburseFlag, '1', 1, 0) * NVL (ClassBalance2, 0)"
             ) / " + Unit + " AS ClassSum2,"
     +"
         COUNT (DECODE (FiveClass, '03', 1, '04', 1, '05', 1, 0)"
     +"
             ) AS UnNormalCount,"
          SUM ( DECODE (FirstDisburseFlag, '1', 1, 0)"
     +"
               * NVL (ClassBalance3 + ClassBalance4 + ClassBalance5, 0)"
     +"
             ) / " + Unit + " AS AllUnNormal,"
     +"
          SUM (DECODE (FirstDisburseFlag, '1', 1, 0) * NVL (ClassBalance3, 0)"
     +"
              ) / " + Unit + " AS ClassSum3,"
          SUM (DECODE (FirstDisburseFlag, '1', 1, 0) * NVL (ClassBalance4, 0)"
     +"
              ) / " + Unit + " AS ClassSum4,"
     +"
          SUM (DECODE (FirstDisburseFlag, '1', 1, 0) * NVL (ClassBalance5, 0)"
             ) / " + Unit + " AS ClassSum5 "
     FROM T FACT LOAN M "
    WHERE InputDate = '" + InputDate + "' "
      AND OrgID LIKE '" + OrgID + "%' "
+" GROUP BY IndustryType "
+" ORDER BY IndustryType";
```



```
//genResult 获取数据
                                      //获得元素结果
genResult("SQL", sSQL);
//matchColumnByArray 根据数组进行列匹配
matchColumnByArray(1,"01,02,03,04,05,06,07,08,09,10,11,12,13,14,15,16,17,18,
19,20,21");
                                      //以数组方式设置匹配数组,补全国标大类
//setPosition 指定起始位置
setPosition("7,2");
                                      //设置起始位置
//deleteColumn 删除指定列
deleteColumn("1");
                                      //删除首列行业分类编号
//setCell 设定单元格内容
setCell("2,1",FillinDate);
                                      //填报日期
setCell("3,1",FillinOrgName);
                                      //填报机构
setCell("3,12",UnitName);
                                      //查询金额单位
                    ===== (END) ======
```

至此,简单固定型报表主要开发过程结束,展示页面能看到该元素的查询结果。



## 7.2 指南二

『简单清单型』

由一个独立报表元素组成,但是元素的结果行数不固定,为可变动数据行,此类表样属于简单清单型报表。

### 7.2.1 报表表样

	A	В	C	D	E	F	G	Н
3	年 月 日							
4	填报单	位:						单位: 万元
5	排序	行业名称	贷款余额		不良贷款			
6	34F/3*	11 11 11 11	余额	比年初	比上期	余额	比年初	比上期
7								
8								
9		合 计	0.00	0.00	0.00	0.00	0.00	0.00
10	行长:				复核:			制表:
11								

对于"简单清单型"报表,调整表样为只剩下**两行空行**(类似上图所示),如果有 纵向求和需求则增加一行求和行(具体请参看《开发手册》)。

# 7.2.2 报表分析

- 1、数据集起点(Position)为(7,2);
- 2、在(3,1)填写"填报日期"、在(4,1)单元格填写填报单位;
- 3、通过 ms EXCEL 实现在第 9 行的纵向求和。

### 7.2.3 过程

- 1、"报表清单"中增加报表模板("报表模板编号"与"模板描述"中触发的 JS 脚本参数要一致),**并将"模板属性"ReportType 设置为"V\_list"**,(关于"报表属性"设置参看『6.2.2 报表清单』);
- 2、"报表元素"新增报表元素
  - a) 元素编号以"模板号"和元素序号构成;
  - b) 标题描述说明本元素的核心功能;
- 3、在"元素脚本"中作如下元素定义

#### //定义变量

var sSQL = "SELECT NVL( IndustryType, '01'), "

+"GetItemName( 'IndustryType', NVL( IndustryType, '01')), "



```
+"
        SUM( Balance), "
                                      //本月余额
        SUM( Balance-BalanceLY), "
   +"
                                     //比年初
   +"
        SUM( Balance-BalanceLM), "
                                    //比上月
        SUM(ClassBalance3 + ClassBalance4 + ClassBalance5), "
   +"
        SUM((ClassBalance3 + ClassBalance4 + ClassBalance5)
        -(ClassBalance3LY+ ClassBalance4LY+ ClassBalance5LY), "
   +"
       SUM((ClassBalance3 + ClassBalance4 + ClassBalance5)
        -(ClassBalance3LM+ ClassBalance4LM+ ClassBalance5LM) "
      FROM T FACT LOAN M "
+"
    WHERE OrgID LIKE '" + OrgID + "%' "
+"
       AND InputDate = '" + InputDate + "' "
       AND FirstDisburseFlag = '1' "//通过首次放款标志避免重复统计
+" GROUP BY NVL( IndustryType, '01') "
+" ORDER BY NVL( IndustryType, '01') ";
//genResult 获取数据
genResult("SQL", sSQL);
//deleteColumn 删除指定列
deleteColumn("1");
//setPosition 指定起始位置
setPosition("6,2");
//setCell 设定单元格内容
setCell("3,1",FillinOrgName);
setCell("3,5",FillinDate);
setCell("3,8",UnitName);
                    =====(END)=====
```

至此,简单固定型报表主要开发过程结束,展示页面能看到该元素的查询结果。

#### [TIPS]

- 1、清单型报表的属性中 ReportType 为 V\_List;
- 2、Html 表样需要调整。



# 7.3 指南三

『传递参数型』

在不同的元素之间,需要使用同一个变量(或字符序列),从而涉及到全局变量的使用,涉及此类使用的报表属于"传递参数型"。

### 7.3.1 报表表样



### 7.3.2 报表分析

表样(如上图)可分解为两个报表元素,一个为年内余额部分,另一个为去年同期 余额,但是两者之间都要遵循在同样十大户的前提下。

#### 【要点】

- 1、本年余额与去年同期余额不在同一数据表内:
- 2、 查询条件有相同限制条件(本用例中的十大户客户编号 CustomerID):
- 3、数据列抽取 fetchColumn (参看《开发手册》)
- 4、设置全局变量 setSheetParam; (参看《开发手册》)
- 5、对传递的参数作再处理 decodeWith\*\*\*;(参看《开发手册》)

#### 7.3.3 过程

- 1、"报表清单"中增加报表模板("报表模板编号"与"模板描述"中触发的 JS 脚本参数要一致);
- 2、"报表元素"新增报表元素
  - a) 元素编号以"模板号"和元素序号构成;
  - b) 标题描述说明本元素的核心功能;



#### 3、在"元素脚本"中作如下元素定义

====( BEGIN )====

```
[Fix0104 001]
  /* Title:
                        最大十家客户贷款余额
                        为保持两元素同为十大户条件,需抽取 fetchColumn 客户 ID
  * Content:
                 设置其为全局变量,便于在另一元素中获取并作为查询条件的一部分,
   * Developer:
                        <someone>
  */
  //定义变量
  var sSQL = "SELECT * FROM ("
      +"SELECT CustomerID, CustomerName, SUM (Balance) / "+Unit+" AS Sum1, "
     +" SUM (balanceLY) / "+Unit+" AS Sum2, '', "
     +" SUM (DECODE (FirstDisburseFlag,'1', 1, 0) * NVL (ClassBalance1, 0) "
               ) / "+Unit+" AS ClassSum1, "
     +" SUM (DECODE (FirstDisburseFlag,'1', 1, 0) * NVL (ClassBalance2, 0) "
     +"
              ) / "+Unit+" AS ClassSum2, "
        SUM (DECODE (FirstDisburseFlag, '1', 1, 0) * NVL (ClassBalance3, 0) "
               ) / "+Unit+" AS ClassSum3, "
     +"
     +" SUM (DECODE (FirstDisburseFlag,'1', 1, 0) * NVL (ClassBalance4, 0) "
     +"
                ) / "+Unit+" AS ClassSum4, "
     +" SUM (DECODE (FirstDisburseFlag,'1', 1, 0) * NVL (ClassBalance5, 0) "
     +"
                ) / "+Unit+" AS ClassSum5, "
     +" SUM (NVL (NormalBalance, 0)) / "+Unit+" AS NrmBalance, "
     +" SUM (NVL (OverdueBalance, 0)) / "+Unit+" AS OvdBalance, "
     +" SUM (NVL (DullBalance, 0)) / "+Unit+" AS DulBalance, "
     +" SUM (NVL (BadBalance, 0)) / "+Unit+" BadBalance, "
     +" SUM (NVL (InterestBalance1, 0) + NVL (InterestBalance2, 0)"
           +" ) / "+Unit+" AS IntBalance "
            FROM T FACT LOAN "
       +"
          WHERE OrgID LIKE '"+OrgID+"%' "
            AND InputDate = '"+InputDate+"' "
       +"
```



```
+" GROUP BY CustomerID, CustomerName "
    +" ORDER BY Sum1 DESC) "
+" WHERE ROWNUM <= 10 ";
//genResult 获取数据
genResult("SQL", sSQL);
//fetchColumn 抽取第 i 列并转换为形如 "'A','B','C','D'" 的字符串
var sOrgIndex = fetchColumn (1);
//setSheetParam 设置 Sheet 变量值
setSheetParam("OrgIndex",sOrgIndex);
//deleteColumn 删除指定列
deleteColumn("1");
//setPosition 指定起始位置
setPosition("6,1");
//setCell 设定单元格内容
setCell("2,1",FillinDate);
setCell("3,1",FillinOrgName);
setCell("3,14",UnitName);
 [Fix0104 002]
/* Title:
                     最大十家客户去年同期贷款余额
* Description:
* Content:
                     为保持两元素同为十大户条件,通过 getSheetParam 获取客户 ID
               字符序列,通过 decodeWith***解析为需要的匹配序列和 SQL 条件。
* Developer:
                     <someone>
                    Feb 21,2005
* DateTime:
* Tester:
*/
/* getSheetParam
 * 获取 Sheet 变量值
 * sParamName 变量名
```



```
*/
/*获取元素一中定义的全局变量*/
var sOrgIndex = getSheetParam("OrgIndex");
var sOrgIndex1 = "";
var sOrgIndex2 = "";
var sSOL = "";
if(sOrgIndex != ""){
   sOrgIndex1 = decodeWithSeparator(sOrgIndex,"'");
   sOrgIndex2 = decodeWithSeparator(sOrgIndex,"");
   sSQL = "SELECT CustomerID, SUM(Balance) / "+Unit
      +" FROM T FACT LOAN M "
      +" WHERE InputDate = '" + PreSyncTerm + "' "
             AND OrgID LIKE '" + OrgID + "%' "
      +"
              AND CustomerID IN("+sOrgIndex1+") "
      +"GROUP BY CustomerID";
   /* genResult
    * 获取数据
    * sSourceType 源数据类型
       SQL: 从 SQL 语句获取数据
    * sSource 数据来源语句
   genResult("SQL",sSQL);
   /* matchColumnByArray
    * 根据数组进行列匹配
    * iColumn 列数
    * sMatchArray 匹配数组
        值1,值2
   matchColumnByArray(1,sOrgIndex2);
   /* deleteColumn
```



```
* 删除指定列
```

```
* sColumnList 列数数组

* 列数 1,列数 2

*/
deleteColumn("1");
/* setPosition

* 指定起始位置

* sPosition Excel 位置

* 行数,列数

*/
setPosition("6,4");
```

}