近两年项目回顾系列——使用Kettle进行数据迁移（ETL） - 大树的博客 - 博客园

由于开发新的系统，需要将之前一个老的C/S应用的数据按照新的数据设计导入到新库中。此过程可能涉及到表结构不一致、大数据量（千万级，甚至上亿）等情况，包括异构数据的抽取、清洗等等工作。部分复杂的工作需要我们的DBA写代码用程序在JDBC或者Delphi中解决，而大部分稍简单的数据的迁移需要一个强大的ETL工具来解决。某日，技术经理让我找一个满足我们项目数据迁移需求的稳定、高效ETL工具。google了几把，网上大致有下列几款软件资料较多：Oracle的OWB（Oracle Warehouse Builder）、AICloudETL、Kettle等等。一一安装并尝试，最终因为Kettle资料丰富、成功案例多、可配合强大的任务调度工具来使用而选定用它。经过测试效率基本满足迁移需要。在2012年7月左右的时候写了一个简单的入门文档供项目组内部使用，现在贴出来，供有类似需求的朋友们参考。【之后我们还用Flex+Quartz开发了一个BS的类似于Kettle的数据转移工具，在我们的项目中每天晚上从一个库迁移数据到另一个库。将在之后的博文中大体讲述】

一　　关于Kettle

Kettle是一款国外开源的ETL工具，纯java编写，数据抽取高效稳定的数据迁移工具。Kettle中有两种脚本文件，transformation和job，transformation完成针对数据的基础转换，job则完成整个工作流程的控制。

二　　本项目中的ETL需求

本项目主要有以下要求：

１、能完成ASCii编码到UTF8编码的转换。

2、稳定。

3、可高效的完成批量数据的转移。

4、能记录、查看（最好能给出分析）转移过程中失败的数据。

5、易于使用，学习成本低。

经测试，以上需求Kettle均可满足，将在之后的操作说明中提及并在最后总结。

三　　操作说明

3.1软件获取

在官网http://kettle.pentaho.com/下载，该软件为绿色版，解压后点击Spoon.bat运行，需要JRE环境支持。（此文档中使用4.2.0 stable版本示例）

3.2 基本操作

Kettle左侧的功能区有“主对象树”和“核心对象”两个面板。其中“核心对象”较为常用。右侧为对象的属性编辑区。可以将左侧的对象拖动到右侧编辑区。同时按键盘shift键在两个对象上画线，可连接两个对象。多个对象连接成为一个transformation。

3.3 示例一：Kettle的基本操作和简单应用

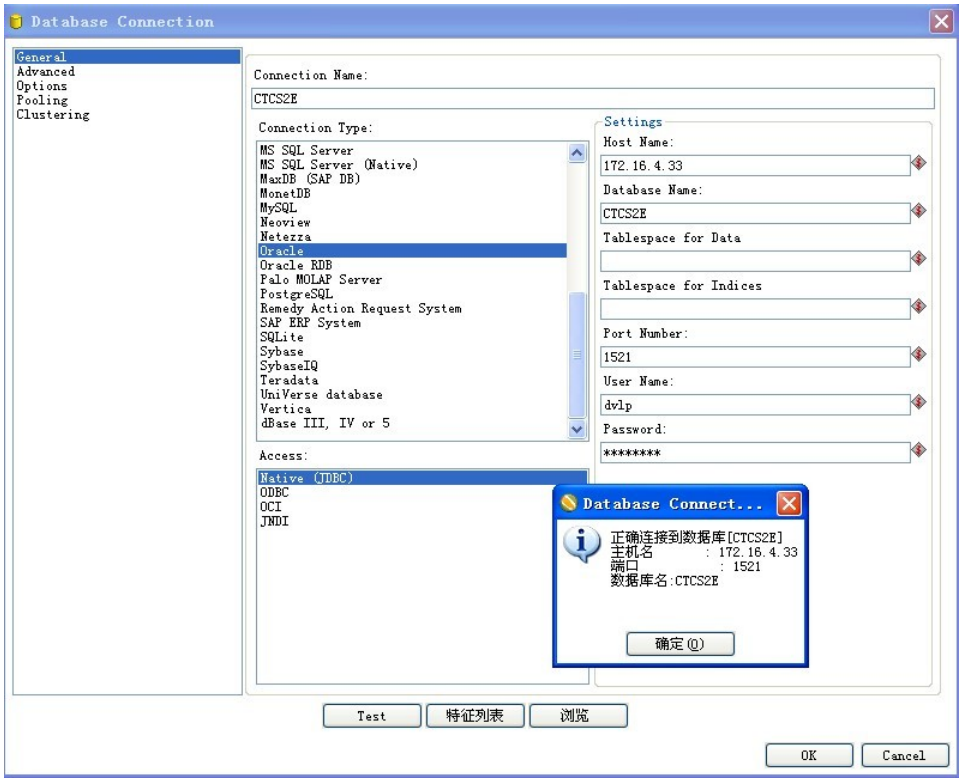
场景要求：此demo假设需抽取***.**.**.33上编码为US7ASCII的某一张表中数据提取到本地上编码为UTF-8的库中。

详细步骤：

1、双击Spoon.bat运行软件，点击“没有资源库”，进入主界面。左上角点击”文件-新建-转换“保存为demo.ktr

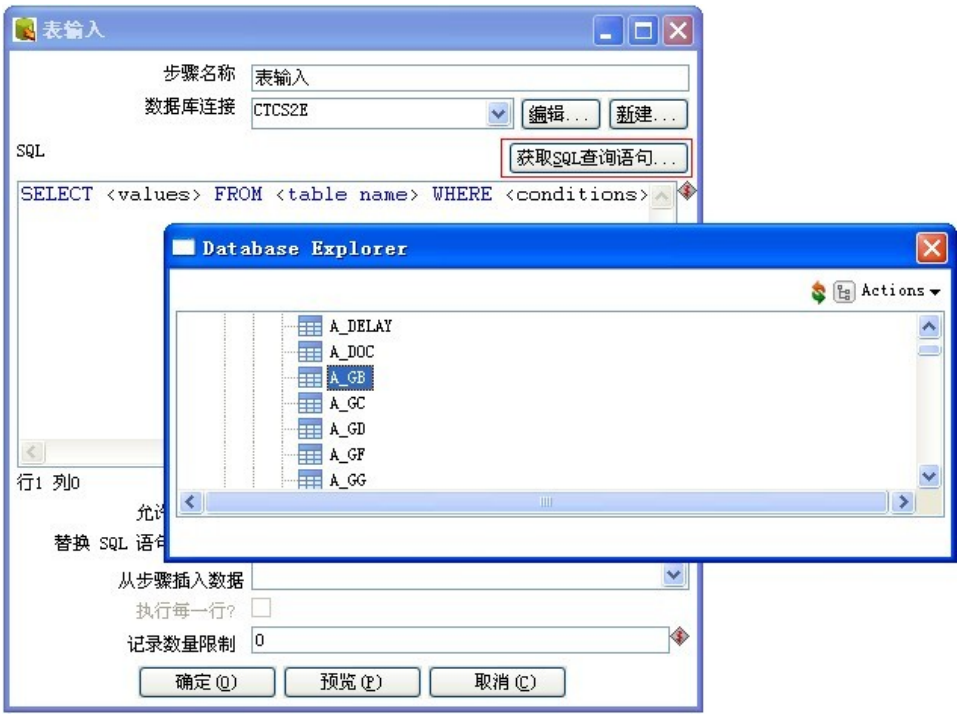
2、左侧选择“核心对象”面板。”在“输入”文件夹下选择“表输入”并把它拖动到右侧编辑区。

3、双击编辑区的“表输入”图标，编辑数据输入来源。点击“数据库连接”右侧的“新建”按钮，按demo背景中的要求，配置数据库参数。配置后点击“Test”，若配置有误将弹出异常提示，根据提示修正。若无误将显示如下信息：



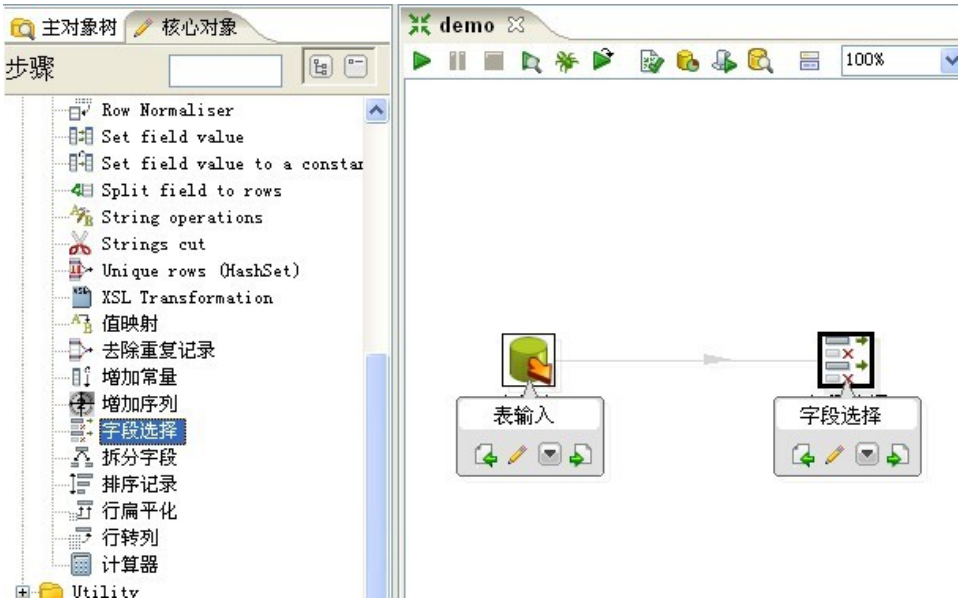
点击“确定”和“OK”，我们为此表输入对象确定了数据库。

4、继续点击“获取SQL”查询语句，选择输入表。这里我们选择A_GB这张表。



选择输入表后，请务必勾选“允许延迟加载”复选框。（否则可能导致乱码，将在文档最后说明。）其余选项默认，点击“确定”完成表输入对象的编辑。

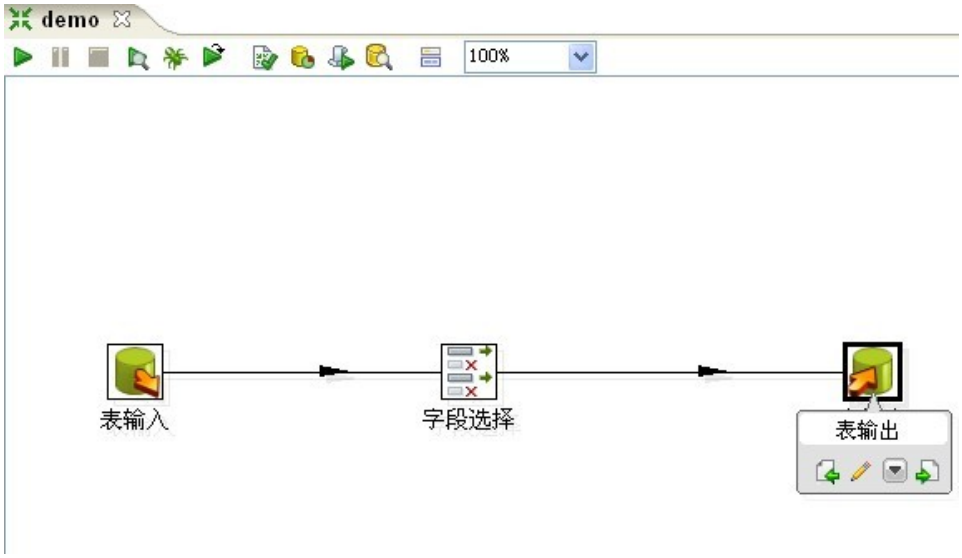
5、在左侧“核心对象”中的“转换”文件夹中选择“字段选择”功能，拖动到右侧编辑区。按住键盘shift同时鼠标从“表输入”为起点，“字段选择”为终点画一条连接。如图：



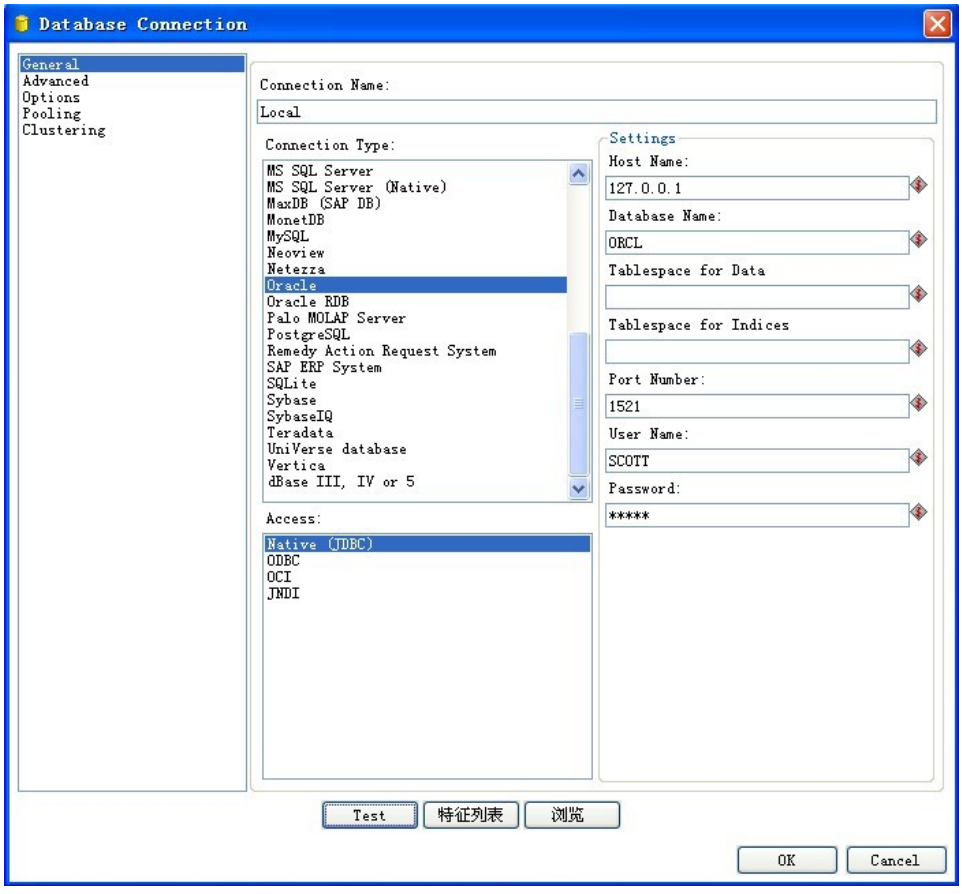
6、双击“字段选择”，打开编辑窗口，选择“元数据”面板，点击右侧“获取改变的字段”，将自动列出之前表输入中所有字段。

7、根据要抽取的目标表字段名（输出字段名），给每一个输入字段改成和输出字段相同的名称。同时请务必在Encoding一栏中选择输出库的编码。根据demo的背景要求，此处我们选择GBK。

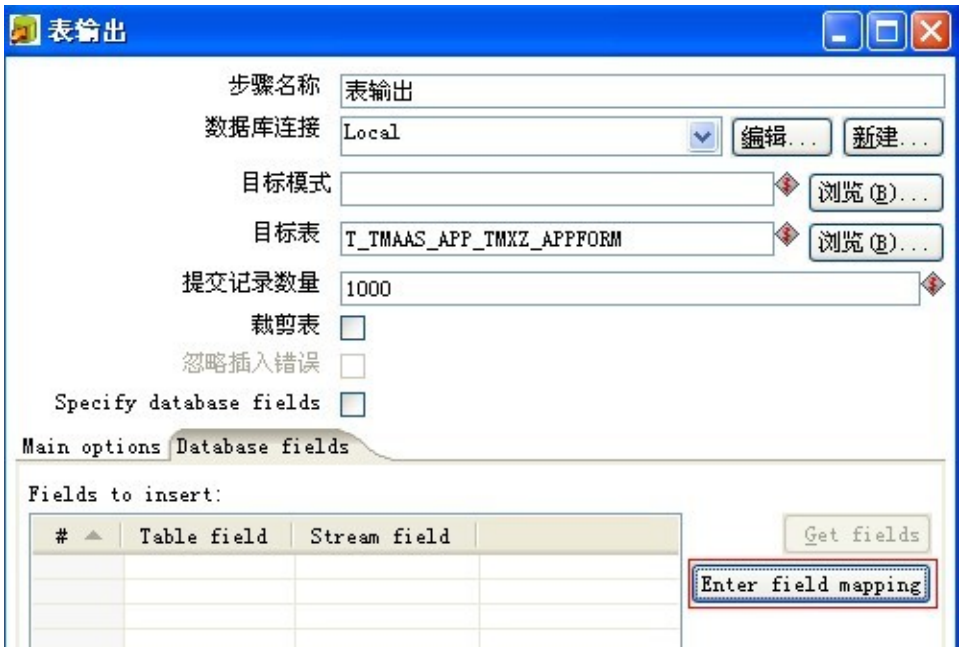
8、编辑完“字段选择”后点击“确定”关闭窗口。同上，在“输出”文件夹中拖动一个“表输出”到右侧编辑区，并画连接。



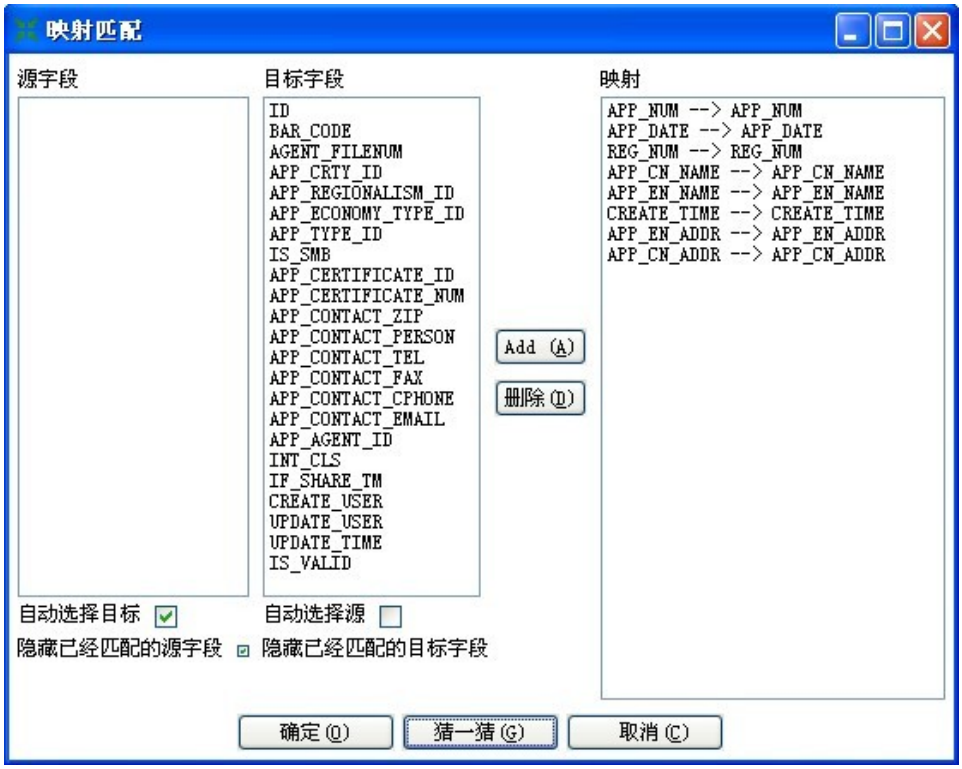
9、双击“表输出”打开其编辑窗口。同“表输入”一样，按照demo背景要求，配置好本地库。



10、同“表输入”，选择输出的目标表。选择后在“Darabase fileds”面板中点击“Enter field mapping”映射输入输出关系。

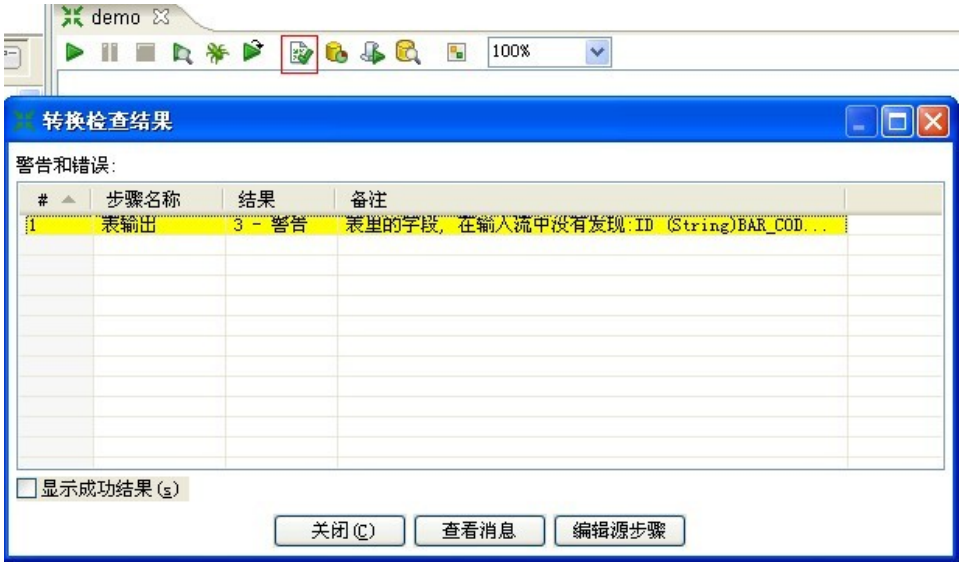


11、因之前已经在“字段选择”中为每一个输入字段改名，这里点“猜一猜”，会根据字段近似度自动匹配映射关系。

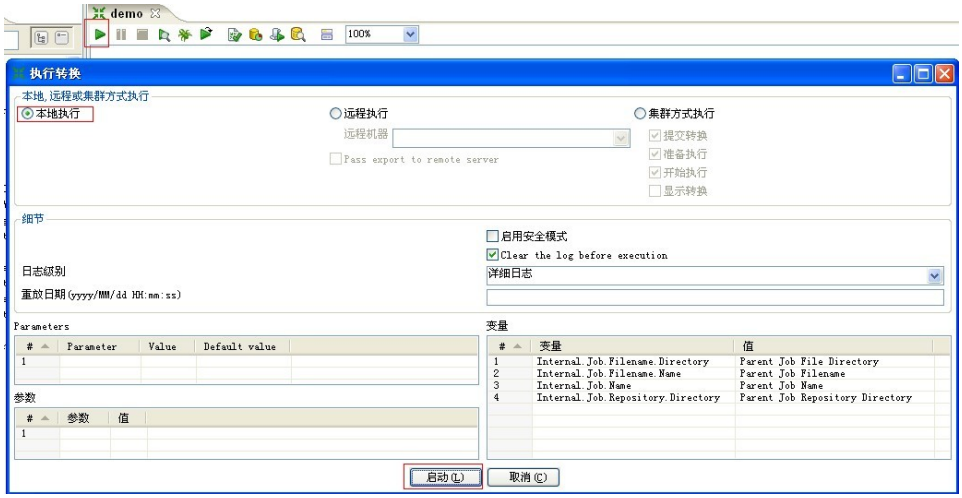


映射完输入输出字段的关系，检查无误后，点击“确定”关闭窗口。

12、至此，我们最简单的一个抽取示例的转换建立完毕，点击“校验这个转换”，Kettle会校验并给出简单的报告。此处只有一个警告，经检查并不影响我们的抽取转换工作，点击“关闭”。



13、点击“运行这个转换”，选择“本地执行”，点击“启动”来执行这个转换。



14、转换的过程可以在控制台实时显示。同时“日志”的详细程度是可选的。

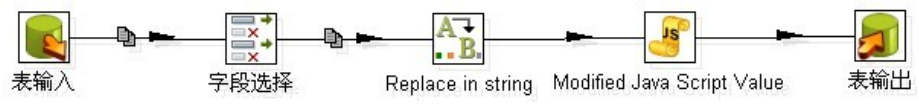
15、执行完毕后，控制台日志若无异常信息，说明转换成功，可以去我们本地库查看。发现确实已被导入新库，两者记录数相同且无乱码。

3.4示例二：字段合并、计算等复杂背景下的应用

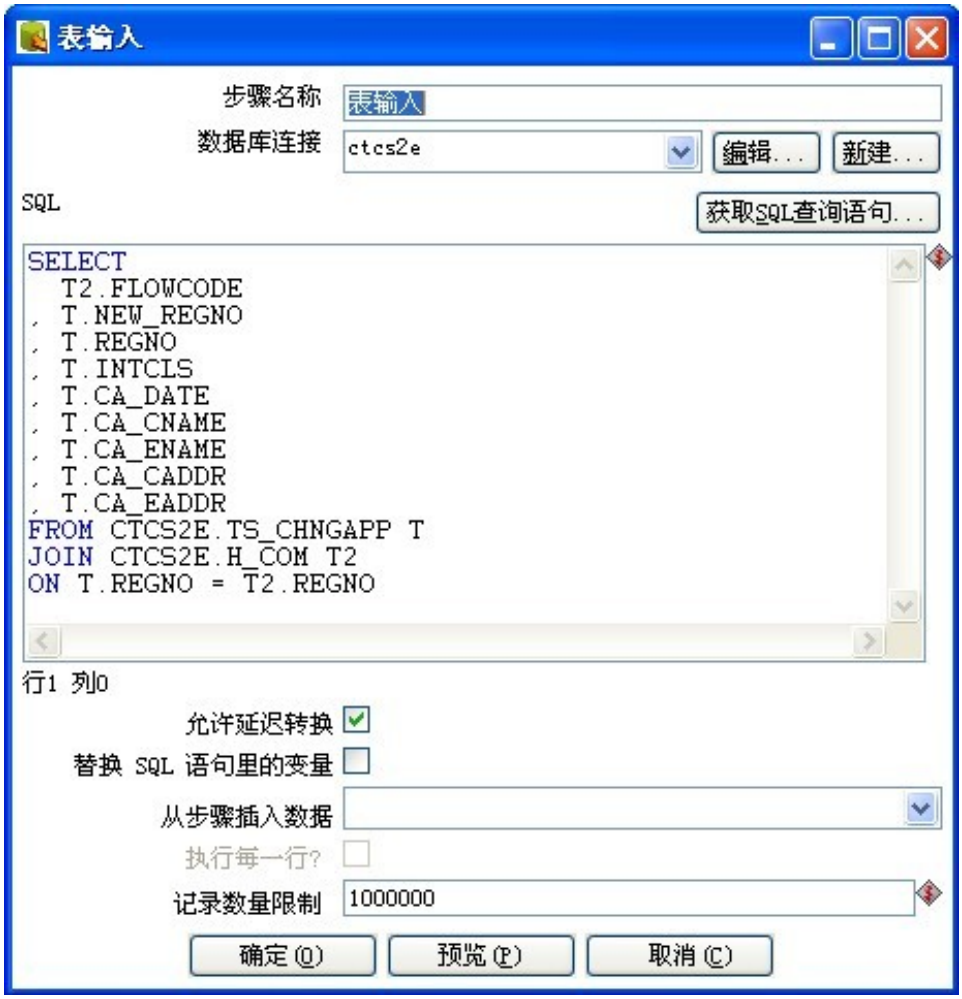
场景要求：要求数据输入来自于两张表，且输出表的某字段需两张输入表的字段进行合并。并可能对某些字段进行字符串操作、日期运算、数学计算等。此示例演示字符串操作、列合并。日期运算和数学计算与此类似，不再赘述。

简略步骤：

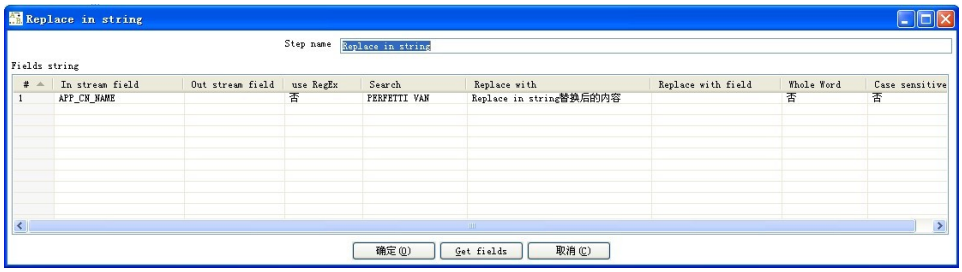
1、基本操作同示例一，其中需引入“Replace in string”和“Modified JavaScript Value”对象。



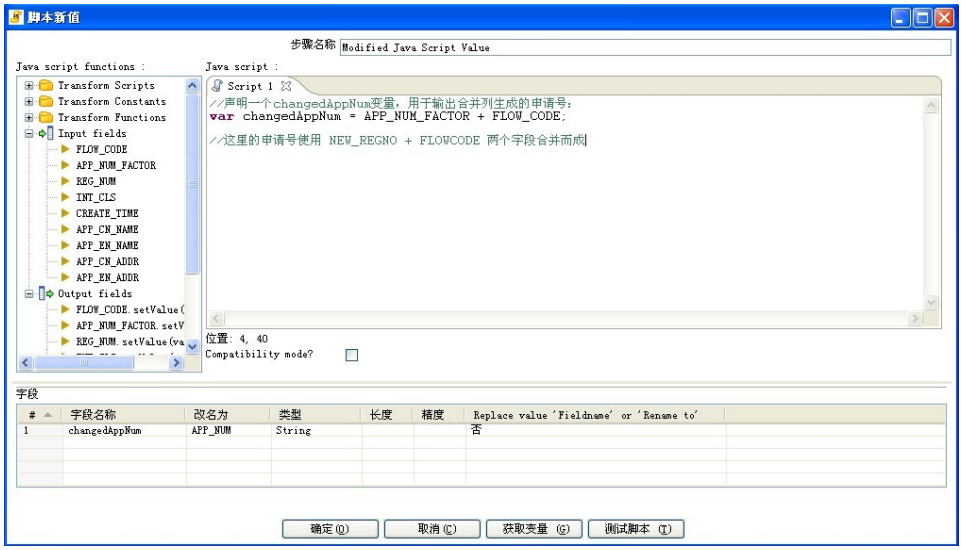
2、表输入：使用一个简单的关连查询，查出所有要抽取的字段和需要合并的列。



3、Replace in string对象：需填写要被替换的输入字段“In Stream field”，这里我们替换APP_CN_NAME字段。是否使用正则表达式“useRegEx”选择“否”，“Search”搜索字符串假设搜索“PERFETTI VAN”，“Replace with”替换为“Replace in string替换后的内容”。“Whole word”是否整个单词和“Case sensitive”大小写敏感均选择“否”。



4、“Modified JavaScript Value”对象：此对象通过编写javaScript脚本来对记录进行高级操作。Kettle内置mozilla的rhino来运行脚本，完成对输入记录的一系列操作。



左侧有大量的字符串、日期、数学运算的库函数可以调用。这里只简单将两列合并为新字段。（若数学、日期运算较复杂，也可以使用“计算器”对象）

此demo中Javascript对象中的值为：

四　　Kettle针对此项目的注意事项

4.1 编码问题

项目要求能完成ASCii编码到UTF8编码的转换。资料显示Kettle默认输入、输出均使用UTF-8编码。为保证不乱码需注意：

输入：此项目的输入是ASCii，故在“表输入”编辑面板务必勾选“允许延迟转换”，便会根据数据库自身的编码读入。否则将会默认以UTF-8读入，可能导致乱码。

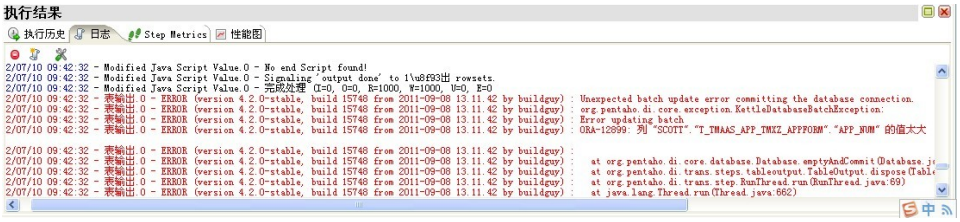
输出：在输出前请使用“字段选择”对象。同时在“字段选择”的“元数据”面板中设置输出编码。可以指定任意输出字符集。

4.2 效率问题

项目要求ETL工具需高效的完成批量数据的转移。查看日志发现Kettle每次输入5W条记录，经过处理再输出。经测试，100W条记录，从172.16.4.33至本地，耗时14min22s。

4.3异常信息

由于Kettle由Java编写，出错时，其异常信息也按照Java异常信息格式打印。如图某错误的日志为：



其信息是：

13.11.42 by buildguy) : org.pentaho.di.core.exception.KettleDatabaseBatchException:

2012/07/10 09:42:32 - 表输出.0 - ERROR (version 4.2.0-stable, build 15748 from 2011-09-08 13.11.42 by buildguy) : Error updating batch

2012/07/10 09:42:32 - 表输出.0 - ERROR (version 4.2.0-stable, build 15748 from 2011-09-08 13.11.42 by buildguy) : ORA-12899: 列 "SCOTT"."T_TMAAS_APP_TMXZ_APPFORM"."APP_NUM" 的值太大 (实际值: 9, 最大值: 4)

由以上异常信息可明显看出在批量更新时出错，错误在“表输出”时出现，具体原因是SCOTT用户下的T_TMAAS_APP_TMXZ_APPFORM表的APP_NUM字段的输出值太大。经检查，该字段最大长度为4，合并后向其输出的长度为9，故抛此异常。

此信息会对异常有较准确的范围描述和简单的原因分析，有利于分析。但未标明是哪一条记录导致。（由于ETL过程可能有复杂的表关联和字段处理，产生异常不一定是输入流中数据的问题，可也能是关联问题、脚本将字段变换后和输出不匹配等问题。尤其是关联后的记录经脚本处理后与输出表结构不匹配时，软件难以定位原始记录，需人工分析。）

4.4、易用性

Kettle由Java编写，在生产中可方便地与Java项目整合，配合任务调度工具可完成强大的ETL工作，使用较为广泛，参考资料丰富。