Informatica®
10.5.2

Developer 转换指南

Informatica Developer 转换指南
10.5.2
2022 年 4 月

发布日期: 2022-07-05

目录

前言

请参阅《*Informatica® Developer 转换指南*》，了解 Informatica 转换的配置、准则、用法和运行时行为。了解如何根据环境和运行时引擎在合适的用例中使用转换。根据计划运行映射的位置和方式查看每个转换的支持。

Informatica 资源

Informatica 通过 Informatica Network 和其他在线门户为您提供一系列产品资源。使用这些资源，可以充分利用 Informatica 产品和解决方案，并向其他 Informatica 用户和主题专家学习。

Informatica Network

在 Informatica Network 中可以获得许多资源，包括 Informatica 知识库和 Informatica 全球客户支持。要进入 Informatica Network，请访问 https://network.informatica.com。

作为 Informatica Network 成员，您可以选择以下服务：

- 在知识库中搜索产品资源。
- 查看产品可用性信息。
- 创建并检查您的支持案例。
- 查找当地的 Informatica 用户组网络并与您的伙伴进行协作。

Informatica 知识库

使用 Informatica 知识库可查找产品资源，例如操作方法文章、最佳实践、视频教程以及常见问题的答案。

要搜索知识库，请访问 https://search.informatica.com。如果您对知识库有任何疑问、意见或建议，请与 Informatica 知识库团队联系，电子邮件地址为 KB_Feedback@informatica.com。

Informatica 文档

使用 Informatica 文档门户可浏览大量当前与最近产品版本的文档库。要浏览文档门户，请访问 https://docs.informatica.com。

如果您对产品文档有任何疑问、意见或建议，请与 Informatica 文档团队联系，电子邮件地址为 infa_documentation@informatica.com。

Informatica 产品可用性矩阵

产品可用性矩阵 (PAM) 指明了产品版本支持的操作系统版本、数据库以及数据源和目标的类型。您可以在以下网址中浏览 Informatica PAM：https://network.informatica.com/community/informatica-network/product-availability-matrices。

Informatica Velocity

Informatica Velocity 是由 Informatica 专业服务根据数百个数据管理项目的实际经验所开发出来的，其中汇集了大量使用技巧和最佳实践。Informatica Velocity 代表了 Informatica 顾问的集体知识，这些顾问与世界各地的组织合作，共同计划、开发、部署和维护成功的数据管理解决方案。

您可以在以下网址中找到 Informatica Velocity 资源：http://velocity.informatica.com。如果您对 Informatica Velocity 有任何疑问、意见或建议，请通过 ips@informatica.com 与 Informatica 专业服务联系。

Informatica Marketplace

Informatica Marketplace 是一个论坛，该论坛中提供的解决方案可扩展和增强您的 Informatica 实施。利用 Informatica 开发人员和合作伙伴在 Marketplace 中提供的数以百计的解决方案，可提高您的工作效率并加快项目实施时间。您可以在以下网址中找到 Informatica Marketplace：https://marketplace.informatica.com。

Informatica 全球客户支持部门

您可以通过电话或 Informatica Network 与全球支持中心联系。

要查找您当地的 Informatica 全球客户支持部门电话号码，请访问 Informatica 网站，链接为：https://www.informatica.com/services-and-training/customer-success-services/contact-us.html。

要在 Informatica Network 上查找联机支持资源，请访问 https://network.informatica.com，然后选择 eSupport 选项。

第 1 章

转换简介

本章包括以下主题：

转换简介概览

转换是一种用于生成、修改或传递数据的对象。

Informatica Developer 提供了一组执行特定功能的转换。 例如，汇总器转换执行数据组的计算。

映射中的转换代表数据集成服务对数据执行的操作。 数据通过您在映射或 Mapplet 中链接的转换端口传递。

转换可以是主动转换或被动转换。转换可以连接到数据流，也可以取消转换的连接。

主动转换

主动转换会更改通过转换传递的行数。 或者更改行类型。

例如，筛选器转换是主动转换，因为其会删除不满足筛选条件的行。 更新策略转换是主动转换，因为其会标记要插入、删除、更新或拒绝的行。

不能将多个主动转换或者一个主动转换和一个被动转换连接到同一个下游转换或转换输入组，因为数据集成服务可能无法将由主动转换传递的多个行连接起来。

例如，映射中的一个分支包含标记要删除的行的更新策略转换。 另一个分支包含标记要插入的行的更新策略转换。 如果将这些转换连接到一个转换输入组，则数据集成服务无法将行的删除和插入操作组合在一起。

被动转换

被动转换不更改通过转换传递的行数，并保持事务边界和行类型。

如果上游分支中的所有转换都是被动转换，可以将多个转换连接到同一个下游转换或同一个转换输入组。源自该分支的转换可以是主动转换也可以是被动转换。

未连接的转换

转换可以连接到数据流，也可以取消转换的连接。 未连接转换不连接到映射中的其他转换。 未连接转换在其他转换内调用，并向该转换返回一个值。

多策略转换

策略是指转换可针对数据执行的一项或多项操作的集合。您可以向转换中的每个策略分配一组不同的输入端口和输出端口。转换会将您定义的策略存储在一个转换对象中。

使用**相关性**视图可查看每个策略使用的端口。

可以在以下转换中定义多个转换策略：

- 大小写转换器
- 分类器
- 判定
- 键生成器
- 标签创建器
- 匹配
- 合并
- 解析器
  要在解析器转换中使用多个策略，请将转换配置为解析标志。
- 标准创建器

转换说明

Developer tool 包含通用转换和数据质量转换。在 Data Engineering Quality 中可使用数据质量转换。

下表介绍了每个转换：

转换	类型	说明
地址验证器	主动或被动	验证并提高邮政地址记录的准确性，并添加有助于用户选择邮件收件人以及投递邮件的信息。
关联	活动	在匹配转换分配给不同群集的重复记录之间创建链接。
汇总器	活动	执行聚合计算。
离散记录异常	活动	标识可能包含数据错误的记录，并将记录加载到 Analyst 工具用户可以审阅和更新的表中。
大小写转换器	被动	标准化字符串大小写。

转换	类型	说明
分类器	被动	写入可汇总输入端口字段中的信息的标签。在字段包含大量文本时使用。
比较	被动	生成数值得分来指示成对输入字符串之间的相似度。
合并	活动	从匹配转换标识为重复的记录中创建合并记录。
数据屏蔽	被动	将敏感生产数据替换为非生产环境中的实际测试数据。 此转换可在管道内连接或不连接。
数据处理器	活动	处理映射中的非结构化和半结构化文件格式。
判定	被动	评估输入数据中的条件，并基于这些条件的结果创建输出。
重复记录异常	活动	标识可能包含重复信息的记录，并将记录加载到 Analyst 工具用户可以审阅和更新的表中。
表达式	被动	计算值。
筛选器	活动	筛选数据。
层次结构到关系	活动	处理层次结构输入并将其转换为关系输出。
Java	主动或被动	执行使用 Java 编码的用户逻辑。存储库会存储用户逻辑的字节代码。
联接器	活动	联接来自不同数据库或平面文件系统的数据。
键生成器	活动	根据选定列中的数据值将记录分配到组。
标签创建器	被动	写入描述输入端口字段中的字符或字符串的标签。
查找	主动或被动	从平面文件、逻辑数据对象、引用表、关系表、视图或同义词中查找并返回数据。 此转换可在管道内连接或不连接。
宏	被动	为 Mapplet 启用动态功能。
匹配	活动	生成得分以指示输入记录间的相似度。
合并	被动	从多个输入列读取数据值，并创建单个输出列。
规范器	活动	处理包含多次出现的数据的源行并为这些数据的每个实例返回一个目标行。
输出	被动	定义 Mapplet 输出行。
解析器	被动	根据输入端口上的值所含的信息类型，将这些值解析到单独的输出端口。
等级	活动	将记录数限制到范围的最高或最低值。
读取	被动	从源读取数据。
关系到层次结构	活动	处理关系输入并将其转换为层次结构输出。

转换	类型	说明
REST Web 服务使用者	活动	作为 Web 服务客户端连接到 REST Web 服务来访问或转换数据
路由器	活动	根据组条件将数据路由到多个转换。
序列生成器	被动	生成值的数字序列。
排序器	活动	根据排序键对数据进行排序。
SQL	主动或被动	对数据库执行 SQL 查询。
标准创建器	被动	生成输入字符串的标准化版本。
联合	活动	合并来自不同数据库或平面文件系统的数据。
更新策略	活动	确定是否插入、删除、更新或拒绝行。
Web 服务使用者	活动	作为 Web 服务客户端连接到 Web 服务，以访问或转换数据。
加权平均值	被动	读取匹配转换为数据集中的记录生成的匹配得分，并计算每对记录的平均得分。您可以对转换为每对记录生成的得分应用不同权重。
写入	被动	表示映射写入数据的目标。

本地和非本地环境下的转换

在非本地环境和本地环境下运行的映射可以返回不同的结果。

请考虑以下处理差异：

- 非本地环境使用分布式处理并在不同的节点上处理数据。每个节点都无权访问其他节点正在处理的数据。因此，运行时引擎可能无法确定数据的产生顺序。所以，当您在非本地环境下运行某个映射并在本地环境下运行同一映射时，这两个映射都会返回正确的结果，但结果可能不相同。
- 非本地环境下的每个运行时引擎对映射逻辑的处理都可以不同。在非本地环境下，Informatica 转换可能完全受支持、有限受支持或不受支持。同样，在本地环境下，某些 Informatica 转换和转换行为可能不受支持。

下表列出了非本地环境中的转换和对不同引擎的支持：

转换	支持的引擎
此表中未列出的转换仅在本地环境中受支持。	
地址验证器	- Blaze - Spark* - Databricks Spark
汇总器	- Blaze - Spark* - Databricks Spark

转换	支持的引擎
大小写转换器	- Blaze - Spark** - Databricks Spark
分类器	- Blaze - Spark* - Databricks Spark
比较	- Blaze - Spark** - Databricks Spark
整合	- Blaze - Spark** - Databricks Spark
数据屏蔽	- Blaze - Spark*
数据处理器	- Blaze - Spark**
判定	- Blaze - Spark** - Databricks Spark
表达式	- Blaze - Spark* - Databricks Spark
筛选器	- Blaze - Spark* - Databricks Spark
Java	- Blaze - Spark*
联接器	- Blaze - Spark* - Databricks Spark
键生成器	- Blaze - Spark** - Databricks Spark
标签创建器	- Blaze - Spark** - Databricks Spark
查找	- Blaze - Spark* - Databricks Spark
宏	- Blaze - Spark* - Databricks Spark

转换	支持的引擎
匹配	- Blaze - Spark** - Databricks Spark
合并	- Blaze - Spark** - Databricks Spark
规范器	- Blaze - Spark* - Databricks Spark
解析器	- Blaze - Spark* - Databricks Spark
Python	- Spark* - Databricks Spark
等级	- Blaze - Spark* - Databricks Spark
路由器	- Blaze - Spark* - Databricks Spark
规则规范	- Databricks Spark
序列生成器	- Blaze - Spark
排序器	- Blaze - Spark* - Databricks Spark
标准创建器	- Blaze - Spark* - Databricks Spark
联合	- Blaze - Spark* - Databricks Spark
更新策略	- Blaze - Spark - Databricks Spark
加权平均值	- Blaze - Spark** - Databricks Spark

转换	支持的引擎
窗口	- Spark***
** 在批处理映射和流映射中均支持。* *** 在批处理映射中有限支持。在流映射中不支持。有关 Spark 引擎上数据处理器转换支持的信息，请参阅 KB article。* **** 仅针对流映射支持。有关详细信息，请参阅《Data Engineering Streaming 用户指南》。*	

处理转换数据类型

转换可以处理数据类型特定函数或允许在不处理数据的情况下传递数据。 数据集成服务根据转换来处理一些数据类型，例如，Decimal、Timestamp with Timezone 和 Timestamp with Local Timezone。

十进制数据类型

可以使用小数数据类型读取数据并将其写入到平面文件和支持的数据库，如 Oracle、Microsoft SQL Server、IBM DB2 和 ODBC。

对于支持精度高达 38 位数的转换，精度为 1 至 38 位数，小数位数为 0 至 38。

支持小数数据类型的转换

以下转换支持精度高达 38 位数的小数数据类型且可对数据执行计算：

- 汇总器
- 数据屏蔽
- 表达式
- 筛选器
- Java
- 联接器
- 查找
- 规范器
- 等级
- 路由器
- 序列生成器
- 排序器
- 联合
- 更新策略

为小数数据类型提供传递支持的转换

某些转换只能通过转换传递精度高达 38 位数的小数数据， 但该转换无法对数据执行任何计算。 使用精度高达 38 位数的小数数据类型对转换执行计算时，数据集成服务会将数据类型作为双精度进行处理。

以下转换为精度高达 38 位数的小数数据类型提供传递支持：

- 数据处理器
- 层次结构到关系
- 宏
- REST Web 服务使用者转换
- SQL
- Web 服务使用者

不支持小数数据类型的转换

某些转换不支持小数数据类型，例如，数据质量转换。

以下数据质量转换适用列表不支持精度高达 38 位数的小数数据类型：

- 地址验证器
- 关联
- 大小写转换器
- 分类器
- 比较
- 合并
- 判定
- 键生成器
- 标签创建器
- 匹配
- 合并
- 解析器
- 标准创建器
- 加权平均值

对于不支持 38 位小数数据类型的转换，当小数数据类型的精度大于 28 位数时，数据集成服务会在高精度模式下将小数值转换为双精度值。

具有时区的时间戳

具有时区的时间戳数据类型是时间戳数据类型的一种变体，其中包含时区偏移或时区区域。

以下转换支持具有时区的时间戳数据类型：

- 汇总器
- 表达式
- 筛选器
- Java
- 联接器
- 查找
- 规范器

- 等级
- 路由器
- 序列生成器
- 排序器
- 联合
- 更新策略

传递支持是指您可以通过转换传递数据，但不能对具有时区的时间戳数据类型执行任何函数。

以下转换为具有时区的时间戳数据类型提供传递支持：

- 数据屏蔽
- 数据处理器
- 层次结构到关系
- 宏
- SQL

具有本地时区的时间戳

具有本地时区的时间戳受转换隐式支持，因为其功能等同于 Timestamp。

开发转换

构建映射时，添加转换并对转换进行配置，以根据业务目的处理数据。

完成以下任务以开发转换并将其合并到映射中：

1. 将不可重用转换添加到映射或 Mapplet 中。或者，创建可添加到多个映射或 Mapplet 中的可重用转换。
2. 配置转换。 每种类型的转换都有唯一一组可以配置的选项。
3. 如果转换可重用，请将其添加到映射或 Mapplet 中。
4. 将转换与映射或 Mapple 中的其他对象链接。

   将端口从上游对象拖至转换输入端口。 将输出端口从转换拖至下游对象上的端口。 某些转换使用您可以选择的预定义端口。

注意: 如果创建可重用转换，请先添加所需的输入和输出端口，然后再将转换链接到其他对象。 不能将端口添加到 Mapplet 或映射画布上的转换实例中。 要更新可重用转换上的端口，请从存储库项目打开转换对象，然后添加端口。

多组转换

转换可以包含多个输入和输出组。组是指定义一行传入或传出数据的端口组。

组类似于关系源或目标定义中的表。大多数转换都有一个输入组和一个输出组。不过，有些包含多个输入组、多个输出组或两者。组是一行数据进入或离开转换的表示形式。

所有多组转换都是主动转换。不能将多个主动转换或一个主动和一个被动转换连接到同一个下游转换或转换输入组。

有些多输入组转换要求集成服务在等待来自其他输入组的行时对输入组中的数据编块。编块转换指对传入数据进行编块的多输入组转换。以下转换为编块转换：

- 启用“输入可编块”属性的自定义转换
- 为未排序的输入配置的联接器转换

在保存或验证映射时，有些包含主动或编块转换的映射可能无效。

多组转换的规则和准则

在映射中连接转换时，对于连接多组转换必须遵循一些规则和准则。

对于多组转换，请遵循以下规则和准则：

- 可以将一个组连接到一个转换或目标。
- 可以将组中的一个或多个输出端口连接到多个转换或目标。
- 不能将某一转换中多个输出组的字段连接到另一转换的同一输入组。
- 不能将不同转换中多个输出组的字段连接到另一转换的同一输入组，除非源和转换之间的每个转换都是被动转换。
- 不能将某一转换中多个输出组的字段连接到另一转换的同一输入组，除非另一个转换是编块转换。
- 不能将输出字段连接到同一输入组中的多个输入字段，除非该组为规范器转换。

转换中的表达式

可以在某些转换的**表达式编辑器**中输入表达式。 表达式将修改数据或测试数据是否与条件相匹配。

创建使用转换语言函数的表达式。 转换语言函数类似于 SQL 函数，用于转换数据。

在使用来自输入端口或输入/输出端口的数据值的端口中输入表达式。例如，您有一个使用输入端口 IN_SALARY 的转换，该端口包含所有员工的薪酬数据。以后可以在映射中使用 IN_SALARY 列中的值。 还可以使用该转换计算总薪酬和平均薪酬。 Developer tool 要求您为每个计算的值创建一个单独的输出端口。

下表列出了可在其中输入表达式的转换：

转换	表达式	返回值
汇总器	基于通过转换传递的所有数据执行聚合计算。或者，可以为聚合计算中的记录指定筛选器，以排除某些类型的记录。例如，可以使用该转换得出分公司所有员工的总人数和平均薪酬。	端口的聚合计算结果。
表达式	基于单个行内的值执行计算。例如，根据特殊物品的价格和数量，您可以计算订单中该行物品的总采购价格。	端口的行级计算结果。

转换	表达式	返回值
筛选器	指定用于筛选通过该转换传递的行的条件。例如，如果要将具有未结余额的客户的客户数据写入到 BAD_DEBT 表中，可以使用筛选器转换筛选客户数据。	TRUE 或 FALSE（根据行是否满足指定条件）。 数据集成服务将返回 TRUE 的行通过该转换传递。 该转换将此值应用到通过其传递的每个行。
联接器	指定用于联接未排序源数据的高级条件。 例如，可以连接名字和姓氏主端口，然后将它们与全名详细信息端口匹配。	TRUE 或 FALSE（根据行是否满足指定条件）。 根据选定联接的类型，数据集成服务将行添加到结果集中，或者放弃该行。
等级	设置包含在某一等级中的行的条件。例如，可以设定就职于组织的前 10 位销售人员的等级。	端口的条件或计算结果。
路由器	根据组表达式将数据路由到多个转换。例如，使用该转换比较处于三个不同薪酬水平的员工的薪酬。这可以通过在路由器转换中创建三个组来实现。例如，为各个薪酬范围创建一个组表达式。	TRUE 或 FALSE（根据行是否满足指定的组表达式）。 数据集成服务通过该转换中的每个用户定义组传递返回结果为 TRUE 的行。 返回 FALSE 的行通过默认组传递。
更新策略	将行标记为更新、插入、删除或拒绝。如果要根据应用的部分条件控制对目标的更新，可使用该转换。例如，当邮寄地址更改时，可以使用更新策略转换将所有客户行标记为要进行更新。 或者，可以将不再就职于组织的人员的所有员工行标记为要拒绝。	更新、插入、删除或拒绝的数字代码。该转换将此值应用到通过其传递的每个行。

表达式编辑器

使用**表达式编辑器**构建类似 SQL 的语句。

可以手动输入表达式，或者使用点击式方法。 从点击式界面选择函数、端口、变量和运算符，以便在构建表达式时最大程度地减少错误。 在表达式中可以包含的最大字符数为 32,767。

表达式中的端口名称

可以在表达式中输入转换端口名称。

对于已连接的转换，如果在表达式中使用端口名称，则当您更改转换中的端口名称时，Developer 工具会更新该表达式。 例如，写入确定 Date_Promised 与 Date_Delivered 这两个日期之差的表达式。 如果您将 Date_Promised 端口名称更改为 Due_Date，Developer 工具会将表达式中的 Date_Promised 端口名称更改为 Due_Date。

注意: 可以将名称 Due_Date 传播到依赖于映射中的此端口的其他不可重用转换中。

将表达式添加到端口

可以将表达式添加到输出端口中。

1. 在转换中，选择端口并打开**表达式编辑器**。
2. 输入表达式。 使用“函数”和“端口”选项卡以及运算符键。

   注意: 无法在表达式中使用转义符。 如果表达式中包括转义符，Developer tool 可能会显示解析错误。

3. 或者，将注释添加到表达式中。

   使用注释指示符 -- 或 //。

4. 单击“验证”按钮以验证表达式。
5. 单击**确定**。
6. 如果表达式无效，修复验证错误并再次验证表达式。
7. 如果表达式有效，单击**确定**以关闭**表达式编辑器**。

表达式中的注释

可以在表达式中添加注释，以描述表达式或指定用于访问有关表达式的业务文档的有效 URL。

要在表达式中添加注释，请使用 -- 或 // 注释指示符。

表达式验证

需要验证表达式以运行映射或预览 Mapplet 输出。

在**表达式编辑器**中使用“验证”按钮以验证表达式。 如果不验证表达式，Developer 工具将在您关闭**表达式编辑器**时对表达式进行验证。 如果表达式无效，Developer 工具会显示一个警告。 可以保存无效的表达式，或者对其进行修改。

测试表达式

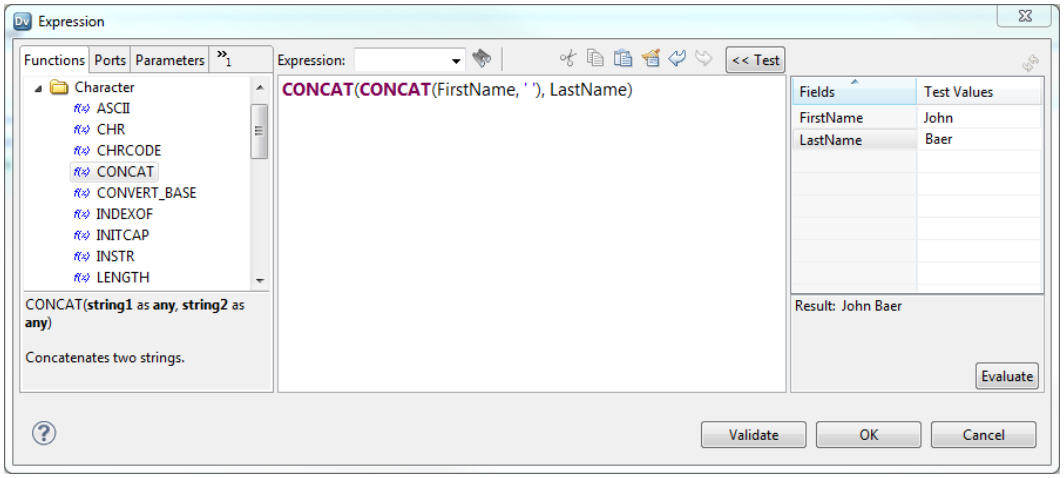
您可以测试一些在表达式编辑器中配置的表达式。 测试某个表达式时，需要输入示例数据，然后对该表达式求值。

在以下位置配置表达式时，可以测试表达式：

- 在表达式转换中的输出或变量端口中
- 将转换添加到映射后在表达式转换的映射输出视图中

例如，在配置连接名字、空格和姓氏的表达式之后，可以输入端口的示例数据，然后对该表达式求值，以便对结果进行验证。

下图显示了连接示例名字和姓氏的表达式的结果：



示例数据的数据格式字符串

测试一个使用数据类型为日期/时间或具有时区的时间戳的端口的表达式时，必须为使用所需日期格式字符串的端口输入示例数据。

要为数据类型为日期/时间的端口输入示例数据，请使用格式 MM/DD/YYYY HH24:MI:SS。 计算该表达式时，表达式编辑器将使用您在该表达式中指定的格式显示结果。 如果在表达式中省略了格式字符串，表达式编辑器将使用相同的格式 MM/DD/YYYY HH24:MI:SS 显示结果。

要为数据类型为具有时区的时间戳的端口输入示例数据，请使用格式 MM/DD/YYYY HH24:MI:SS TZR。 对该表达式进行求值时，表达式编辑器将使用格式 YYYY-MM-DD HH24:MI:SS.NS TZR 显示结果。

数据类型转换

多个表达式和汇总函数可能会生成数据类型不同于输入数据的数据。

例如，将两个精度为 18 位的小数数字相乘时，生成的数据类型可能是精度为 28 位的小数。

输入数据类型是精度为 38 位的小数时，某些运算的结果数据可能不适合所得到的数据类型。 这种情况下，用户可能会得到溢出异常。

以下函数可能需要进行数据类型转换，才能接受超出输入数据类型的数据大小：

- avg
- cume
- divide
- median
- movingavg
- movingsum
- multiply
- Percentile
- Sum

例如，输入数据是整数数据类型并且您使用乘法运算时，生成的数据类型可能是长整型数据类型。 类似地，输入数据类型是精度为 18 位的小数数据类型时，乘法运算的结果可能较大并且位于精度为 28 位的小数数据类型内。

局部变量

在汇总器、表达式和等级转换中使用局部变量可提高性能。可以在表达式中引用变量或使用它们临时存储数据。

可以使用变量完成以下任务：

- 临时存储数据。
- 简化复杂表达式。
- 存储来自以前行的值。
- 捕获来自存储过程的多个返回值。
- 比较值。
- 存储未连接的查找转换的结果。

临时存储数据和简化复杂表达式

在同一转换中输入多个相关表达式时，变量可提高性能。可以将组件定义为变量，而不必在转换中多次解析和验证相同的表达式组件。

例如，如果汇总器转换在计算总和和平均值以前使用相同的筛选器条件，则可以将此条件定义为变量，然后在两项汇总计算中重用该条件。

可以简化复杂表达式。如果汇总器在多个表达式中的计算相同，则可以通过创建变量来存储计算结果而提高性能。

例如，您可以创建以下表达式来查找相同数据的平均薪资和总薪资。

```
AVG( SALARY, ( ( JOB_STATUS = 'Full-time' ) AND (OFFICE_ID = 1000 ) ) )
SUM( SALARY, ( ( JOB_STATUS = 'Full-time' ) AND (OFFICE_ID = 1000 ) ) )
```

无需为两种计算输入相同的参数，可以为此计算中的每个条件创建一个变量端口，然后将表达式改为使用变量。

下表显示了如何使用变量简化复杂表达式和临时存储数据：

端口	值
V_CONDITION1	JOB_STATUS = ‘Full-time’
V_CONDITION2	OFFICE_ID = 1000
AVG_SALARY	AVG(SALARY, (V_CONDITION1 AND V_CONDITION2))
SUM_SALARY	SUM(SALARY, (V_CONDITION1 AND V_CONDITION2))

存储行中的值

您可以在转换中配置变量，以存储源行的数据。在转换表达式中可以使用变量。

例如，源文件包含以下行：

```
California
California
California
Hawaii
Hawaii
New Mexico
New Mexico
New Mexico
```

每行包含一个省/自治区/直辖市。您需要统计行数并对每个省/自治区/直辖市返回行数：

```
California,3
Hawaii    ,2
New Mexico,3
```

您可以配置一个汇总器转换来按省/自治区/直辖市组合源行，并统计每组中的行数。在汇总器转换中配置变量，以存储行计数。定义另一个变量来存储上一行的省/自治区/直辖市名称。

汇总器转换具有以下端口：

端口	端口类型	表达式	说明
省/自治区/直辖市	传递	n/a	省/自治区/直辖市的名称。源行按省/自治区/直辖市名称进行组合。汇总器转换针对每个省/自治区/直辖市返回一行。
State_Count	变量	IIF (PREVIOUS_STATE = STATE, STATE_COUNT +1, 1)	当前省/自治区/直辖市的行计数。在当前“省/自治区/直辖市”列的值与 Previous_State 列相同时，集成服务将对 State_Count 递增。否则，则将 State_Count 重置为 1。
Previous_State	变量	省/自治区/直辖市	上一行中“省/自治区/直辖市”列的值 当集成服务处理行时，会将“省/自治区/直辖市”值移到 Previous_State 中。
State_Counter	输出	State_Count	汇总器转换针对省/自治区/直辖市处理的行数。集成服务对于每个省/自治区/直辖市返回一次 State_Counter。

捕获来自存储过程的值

变量提供一种捕获来自存储过程的多列返回值的方式。

配置变量端口的准则

在转换中配置变量端口时，应考虑以下因素：

- **端口顺序。**集成服务按依赖关系计算端口。转换中端口的顺序必须与计算顺序匹配。输入端口、变量端口、输出端口。
- **数据类型。**所选的数据类型反映您输入的表达式的返回值。.
- **变量初始化。**集成服务设置变量端口的初始值，在变量端口中可创建计数器。

端口顺序

集成服务首先计算输入端口。接下来，集成服务计算变量端口，最后是输出端口。

集成服务按照以下顺序计算端口：

1. **输入端口。**集成服务首先计算所有输入端口，因为它们不依赖于任何其他端口。因此，可以按任何顺序创建输入端口。集成服务不会对输入端口排序，因为输入端口不引用其他端口。
2. **变量端口。**变量端口可引用输入端口和变量端口，但不引用输出端口。由于变量端口可引用输入端口，所以集成服务要在输入端口之后计算变量端口。变量可引用其他变量，所以变量端口的显示顺序与集成服务计算各个变量的顺序相同。

   例如，如果要计算建筑物的原始价值，然后调整进行折旧，则可以创建原始价值计算作为变量端口。此变量端口需要出现在调整折旧的端口之前。
3. **输出端口。**集成服务最后计算输出端口，因为输出端口可引用输入端口和变量端口。输出端口的显示顺序无关紧要，因为输出端口无法引用其他输出端口。务必使输出端口显示在端口列表的最底部。

数据类型

将端口配置为变量时，可以在其中输入任意表达式或条件。为此端口选择的数据类型反映您输入的表达式的返回值。如果通过变量端口指定条件，则任何数值数据类型将针对 TRUE（非零）和 FALSE（零）返回值。

变量初始化

集成服务不会将变量的初始值设置为空。

集成服务遵循以下准则设置变量初始值：

- 数值端口为零
- 字符串端口为空字符串
- 日期/时间端口为 01/01/0001

因此，使用变量作为计数器需要初始值。例如，可以使用以下表达式创建数值变量：

```
VAR1 + 1
```

此表达式将统计 VAR1 端口中的行数。如果将变量初始值设置为空，该表达式将始终计算为空。因此，应将初始值设置为零。

端口的默认值

所有转换均使用默认值，以确定集成服务如何处理输入空值和输出转换错误。

输入、输出和输入/输出端口都有系统默认值，有时可使用用户定义的默认值替代它们。默认值在不同类型的端口中有不同的功能：

- **输入端口。**空输入端口的系统默认值为空。默认值在转换中看起来为空白。如果输入值为空，则集成服务将使其保留为空。
- **输出端口。**输出转换错误的系统默认值为 ERROR。默认值在转换中看起来为 ERROR（“转换错误”）。如果出现转换错误，则集成服务将跳过该行。集成服务将在日志文件中记入 ERROR 函数跳过的所有输入行。

  以下错误为转换错误：

  - 数据转换错误，例如将数字传递给日期函数。
  - 表达式计算错误，例如除以零。
  - 调用 ERROR 函数。
- **传递端口。**空输入的系统默认值与输入端口相同 — 空。系统默认值在转换中看起来为空白。输出转换错误的默认值与输出端口相同。转换中不显示输出转换错误的默认值。

注意: Java 转换会根据 Java 转换端口类型将 PowerCenter® 数据类型转换为 Java 数据类型。空输入的默认值视 Java 数据类型有所不同。

下表显示了已连接转换中端口的默认值：

端口类型	默认值	集成服务行为	支持用户定义的默认值
输入、传递	空值	集成服务以空值形式传递所有输入空值。	输入、输入/输出
输出、传递	ERROR	对于输出端口转换错误，集成服务将调用 ERROR 函数。集成服务将跳过错误行，并在日志文件中写入输入数据和错误消息。	输出

变量端口不支持默认值。集成服务将根据数据类型初始化变量端口。

在集成服务遇到空输入值和输出转换错误时，可以替代部分默认值来更改其行为。

用户定义的默认值

对于已连接转换内支持的输入、传递和输出端口，可以使用用户定义的默认值替代系统默认值。

要替代端口的系统默认值，请遵循以下规则和准则：

- **输入端口。**如果不希望集成服务将空值视为 NULL，可以为输入端口输入用户定义的默认值。如果 NULL 传递至输入端口，集成服务则使用默认值来替换 NULL。
- **输出端口。**如果不希望集成服务跳过行或希望集成服务在日志中对跳过的行写入特定消息，可以为输出端口输入用户定义的默认值。如果在输出端口中定义默认值，则在输出端口发生转换错误时，集成服务会将行替换为默认值。
- **传递端口。**如果不希望集成服务将空值视为 NULL，可以为传递端口输入用户定义的默认值。在传递端口中，不能针对输出转换错误输入用户定义的默认值。

注意: 对于未连接的转换，集成服务将忽略用户定义的默认值。例如，如果通过表达式调用查找或存储过程转换，集成服务将忽略任何用户定义的默认值并应用系统默认值。

使用以下选项输入用户定义的默认值：

- **常量值。**使用任意常量（数值或文本），包括 NULL。
- **常量表达式。**可以在转换函数中使用常量参数。
- **ERROR。**生成转换错误。在映射日志或行错误日志中写入行和消息。
- **ABORT。**中止映射。

常量值

可以输入任意常量值作为默认值。常量值必须与端口数据类型匹配。

例如，数值端口的默认值必须是数值型常量。部分常量值包括：

```
0
9999
NULL
'Unknown Value'
'Null input data'
```

常量表达式

常量表达式是指使用转换函数（除汇总函数外）写入常量表达式的任何表达式。常量表达式中不能使用来自输入、输入/输出或变量端口的值。

部分有效的常量表达式包括：

```
500 * 1.75
TO_DATE('January 1, 1998, 12:05 AM','MONTH DD, YYYY, HH:MI AM')
ERROR ('Null not allowed')
ABORT('Null not allowed')
SESSSTARTTIME
```

在表达式内不能使用来自端口的值，因为集成服务在初始化映射时会为整个映射分配默认值。

以下示例使用了来自端口的值，所以是无效的：

```
AVG(IN_SALARY)
IN_PRICE * IN_QUANTITY
:LKP(LKP_DATES, DATE_SHIPPED)
```

注意: 不能从默认值表达式调用存储过程或查找表。

ERROR 和 ABORT 函数

对于输入和输出端口默认值及输入/输出端口的输入值，可使用 ERROR 和 ABORT 函数。当集成服务遇到 ERROR 函数时，它将跳过该行。当它遇到 ABORT 函数时，将中止映射。

用户定义的默认输入值

如果不希望集成服务将空值视为空，可以输入用户定义的默认输入值。

要替代空值，请完成以下任务之一：

- 将空值替换为常量值或常量表达式。
- 使用 ERROR 函数跳过空值。
- 使用 ABORT 函数中止映射。

下表汇总了集成服务如何处理输入和输入/输出端口的空值输入：

默认值	默认值类型	说明
空（显示空白）	系统	集成服务传递空值。
常量或常量表达式	用户定义	集成服务将空值替换为常量值或常量表达式。
ERROR	用户定义	集成服务将此情况视为转换错误： - 转换错误计数增加 1。 - 跳过该行，并向日志文件或行错误日志写入错误消息。 集成服务不会将这些行写入拒绝文件。
中止	用户定义	当集成服务遇到空输入值时，映射将中止。集成服务不会增加错误计数或将这些行写入拒绝文件。

替换空值

使用常量值或表达式以指定值替代端口的空值。

例如，如果输入字符串端口称为 DEPT_NAME，您希望将空值替换为字符串“UNKNOWN DEPT”，则可以将默认值设置为“UNKNOWN DEPT”。集成服务会将“UNKNOWN DEPT”传递到转换内的表达式或变量或数据流中的下一个转换，具体视转换而定。

例如，集成服务可将端口中的所有空值替换为字符串“UNKNOWN DEPT”。

```
DEPT_NAME    REPLACED VALUE
Housewares     Housewares
NULL           UNKNOWN DEPT
Produce        Produce
```

跳过空记录

当不希望将空值传递到转换中时，可使用 ERROR 函数作为默认值。例如，当 DEPT_NAME 的值为空时，您可能希望跳过某行。可以使用以下表达式作为默认值：

```
ERROR('Error. DEPT is NULL')
```

使用 ERROR 函数作为默认值时，集成服务将跳过具有空值的该行。集成服务会将 ERROR 函数跳过的所有行写入日志文件， 而不会将这些行写入拒绝文件。

```
DEPT_NAME     RETURN VALUE
Housewares    Housewares
NULL          'Error. DEPT is NULL' (Row is skipped)
Produce       Produce
```

以下日志显示了集成服务跳过具有空值的行的位置：

```
TE_11019 Port [DEPT_NAME]: Default value is: ERROR(<<Transformation Error>> [error]: Error. DEPT is NULL
... error('Error. DEPT is NULL')
).
CMN_1053 EXPTRANS: : ERROR: NULL input column DEPT_NAME: Current Input data:
CMN_1053   Input row from SRCTRANS: Rowdata: ( RowType=4 Src Rowid=2 Targ Rowid=2
  DEPT_ID (DEPT_ID:Int:): "2"
  DEPT_NAME (DEPT_NAME:Char.25:): "NULL"
  MANAGER_ID (MANAGER_ID:Int:): "1"
)
```

中止映射

当集成服务遇到空输入值时，可使用 ABORT 函数中止映射。

默认值验证

输入默认值时，Developer tool 工具将对其进行验证。

保存映射时，Developer tool 将验证默认值。如果输入的默认值无效，Developer tool 会将该映射标记为无效。

用户定义的默认输出值

可以创建用户定义的默认值来替代输出端口的系统默认值。

如果不希望集成服务跳过错误行或希望集成服务在日志中对跳过的行写入特定消息，可以为输出端口输入用户定义的默认值。当集成服务遇到输出转换错误时，可以输入默认值以完成以下函数：

- 使用常量值或常量表达式替换错误。集成服务不会跳过该行。
- 使用 ABORT 函数中止映射。
- 在日志中针对转换错误写入特定消息。

不能为输入/输出端口输入用户定义的默认输出值。

下表汇总了集成服务如何处理输出端口转换错误及转换中的默认值：

默认值	默认值类型	说明
转换错误	系统	出现转换错误时，如果您未替代默认值，集成服务将执行以下任务： - 转换错误计数增加 1。 - 跳过该行，并将错误和输入行写入会话日志文件或行错误日志，具体视会话配置而定。 集成服务不会将该行写入拒绝文件。
常量或常量表达式	用户定义	集成服务将使用默认值替换错误。 集成服务不会增加错误计数或向日志中写入消息。
中止	用户定义	映射中止，并且集成服务向日志中写入消息。 集成服务不会增加错误计数或将这些行写入拒绝文件。

替换错误

如果在出现转换错误时不希望集成服务跳过行，可使用常量或常量表达式作为输出端口的默认值。

例如，如果有某个数值输出端口名为 NET_SALARY，您希望在出现转换错误时使用常量值“9999”，可将默认值 9999 分配给 NET_SALARY 端口。如果在计算 NET_SALARY 的值时出现任何转换错误（例如除以零），集成服务将使用默认值 9999。

中止映射

当集成服务遇到空输入值时，可使用 ABORT 函数中止会话。

在映射日志中写入消息

如果您希望集成服务在映射日志中针对跳过的行写入特定消息，可以在输出端口中配置用户定义的默认值。系统默认值为 ERROR（“转换错误”），并且集成服务将在日志中对跳过的行写入消息“转换错误”。如果要写入其他消息，可以替换“转换错误”。

输出端口表达式中的 ERROR 函数

如果输入使用 ERROR 函数的表达式，则输出端口的用户定义的默认值可替代表达式中的 ERROR 函数。

例如，输入以下表达式以指示集成服务在遇到错误时使用值“Negative Sale”：

```
IIF( TOTAL_SALES>0, TOTAL_SALES, ERROR ('Negative Sale'))
```

以下示例显示了用户定义的默认值可如何替代表达式中的 ERROR 函数：

- **常量值或表达式。**常量值或表达式将替代输出端口表达式中的 ERROR 函数。

  例如，如果您输入“0”作为默认值，集成服务将替代输出端口表达式中的 ERROR 函数。遇到错误时，它将传递值 0。它不会跳过该行或在日志中写入“Negative Sale”。

- **ABORT.**ABORT 函数将替代输出端口表达式中的 ERROR 函数。

  如果使用 ABORT 函数作为默认值，出现转换错误时集成服务将中止会话。ABORT 函数将替代输出端口表达式中的 ERROR 函数。

- **错误。** If you use the ERROR function as the default value, the Integration Service includes the following information in the log:

  - 来自默认值的错误消息

- 输出端口表达式中 ERROR 函数中指示的错误消息
- 跳过的行

例如，可以使用以下 ERROR 函数替代默认值：

```
ERROR('No default value')
```

集成服务将跳过该行，并在日志中包括错误消息。

```
 TE_7007 Transformation Evaluation Error; current row skipped...
 TE_7007 [<<Transformation Error>> [error]: Negative Sale
... error('Negative Sale')
]
Sun Sep 20 13:57:28 1998
TE_11019 Port [OUT_SALES]: Default value is: ERROR(<<Transformation Error>> [error]: No default value
... error('No default value')
```

默认值的常规规则

创建默认值时，请遵循以下规则和准则：

- 默认值必须是空值、常量值、常量表达式、ERROR 函数或 ABORT 函数。
- 对于输入/输出端口，集成服务将使用默认值处理空输入值。输入/输出端口的输出默认值始终为 ERROR（“转换错误”）。
- 变量端口不使用默认值。
- 在汇总器和等级转换中，可以按端口将默认值分配到组中。
- 并不是所有转换中的所有端口类型都允许用户定义的默认值。如果端口不允许用户定义的默认值，将禁用默认值字段。
- 并不是所有转换都允许用户定义的默认值。
- 如果转换未连接到映射数据流，集成服务将忽略用户定义的默认值。
- 如果任何输入端口未连接，将假定其值为空，集成服务将对该输入端口使用默认值。
- 如果输入端口默认值包含 ABORT 函数且输入值为空，集成服务将立即停止映射。使用 ABORT 函数作为默认值可限制空输入值。输入端口中的第一个空值将停止映射
- 如果输出端口默认值包含 ABORT 函数且该端口发生任何转换错误，映射将立即停止。使用 ABORT 函数作为默认值可对转换错误强制执行严格规则。此端口的第一个转换错误将停止映射
- ABORT 函数、常量值和常量表达式将替代在输出端口表达式中配置的 ERROR 函数。

默认值验证

输入默认值时，Developer tool 工具将对其进行验证。

保存映射时，Developer tool 将验证默认值。如果输入的默认值无效，Developer tool 会将该映射标记为无效。

跟踪级别

配置转换时，可以设置数据集成服务写入日志中的详细信息量。

默认情况下，每个转换的跟踪级别都为“普通”。 需要对行为方式不符合预期的转换进行故障排除时，请将跟踪级别更改为“详细”设置。 希望日志中显示最小详细信息量时，请将跟踪级别设置为“简洁”。

在“高级”选项卡上配置以下属性：

跟踪级别

转换的日志中显示的详细信息量。

下表介绍了跟踪级别：

跟踪级别	说明
简洁	记录初始化信息、错误消息和已拒绝数据的通知。
普通	记录初始化信息和状态信息、遇到的错误以及由于转换行错误而跳过的行。汇总映射结果（但不是在单个行的级别）。 默认值为“普通”。
详细初始化	除了正常跟踪以外，还记录其他初始化详细信息、索引名称和使用的数据文件，以及详细的转换统计信息。
详细数据	除了详细初始化跟踪以外，还记录传递到映射的每一行。 并且标注在何处截断字符串数据以符合列的精度，并提供详细的转换统计信息。 如果配置了跟踪级别，块中所有行的行数据都会在处理转换时写入日志。

可重用转换

可重用转换是可以在多个映射或 Mapplet 中使用的转换。

例如，可创建一个表达式转换，用于计算在加拿大的销售增值税，以分析在该国的经营成本。 您可以创建可重用转换，而不是每次执行同样的工作。 需要将该转换合并到映射中时，可以将该转换的一个实例添加到映射中。 更改转换的定义时，其所有实例都将继承更改。

Developer 工具将每个可重用转换作为独立于使用该转换的任何映射或 Mapplet 的元数据进行存储。 Developer 工具会将可重用转换存储在项目或文件夹中。

将可重用转换的实例添加到映射中后，对转换进行更改可能会使映射无效，或者生成意外的数据。

可重用转换实例和继承的更改

将可重用转换添加到映射或 Mapplet 中时，将添加转换的实例。 转换的定义仍存在于映射或 Mapplet 的外部，而转换的实例则显示在映射或 Mapplet 内。

更改转换时，转换的实例将反映这些更改。 您无需在每个使用相同转换的映射中都更新该转换，而是可以更新一次可重用转换，该转换的所有实例都将继承此更改。 这些实例将继承对端口、表达式、属性和转换名称进行的更改。

编辑可重用转换

编辑可重用转换时，该转换的所有实例都将继承这些更改。 某些更改可能会导致使用可重用转换的映射无效。

可以在编辑器中打开转换以编辑可重用转换。 无法编辑映射中的转换的实例。 但可以编辑转换运行时属性。

如果对可重用转换进行了以下任意更改，则使用该转换的实例的映射可能无效：

- 删除转换中的一个或多个端口时，会将该实例与通过映射的部分或全部数据流断开连接。
- 更改端口数据类型时，无法将来自该端口的数据映射到使用不兼容数据类型的其他端口。

- 更改端口名称时，引用该端口的表达式将不再有效。
- 在可重用转换中输入无效的表达式时，使用该转换的映射将不再有效。 数据集成服务无法运行无效的映射。

可重用转换的编辑器视图

可在编辑器的视图中为可重用的 Java 转换定义属性并创建 Java 代码。

对于可重用的转换，提供了以下视图：

概览

请输入转换的名称和说明，然后创建和配置输入端口和输出端口。

高级

为转换设置高级属性。

不可重用转换

不可重用转换指您在特定映射中创建的转换。无法在任何其他映射中使用该转换。

例如，可以创建包含多个转换的映射。每个转换针对源数据执行计算。在映射结束时，创建不可重用汇总器转换以处理结果。创建不可重用转换后，可将端口从某一转换拖动到映射中的另一个转换并创建输入端口。

Developer 工具将不可重用汇总器转换存储为与该映射相关的元数据。

适用于不可重用转换的编辑器视图

在编辑器视图中定义不可重用的转换的属性。

对于不可重用的转换，将显示以下视图：

常规

输入转换的名称和说明。

端口

创建和配置输入端口和输出端口。

高级

为转换设置高级属性。

创建转换

可以创建在多个映射或 Mapplet 中重用的可重用转换。 或者，可以创建在映射或 Mapplet 中使用一次的不可重用转换。

要创建可重用转换，单击**文件** > **新建** > **转换**并完成向导。

要在映射或 Mapplet 中创建不可重用转换，请从“转换”选项板中选择转换，并将该转换拖至编辑器中。

某些转换需要在您创建转换时选择一种模式或执行其他配置。 例如，在创建查找转换时，必须选择一个数据对象作为查找源。

创建转换后，转换将显示在编辑器中。 一些转换包含预定义的端口和组。 其他转换为空。

第 2 章

转换端口

本章包括以下主题：

转换端口概览

转换端口是在映射或 Mapplet 中进行连接的各个数据列。转换接收来自输入端口的数据，并使用输出端口将数据发送到外部。输入/输出端口接收数据并按原样传递数据。

每个输入对象、输出对象、Mapplet 和转换都包含一组端口。输入对象用于提供数据，所以仅包含输出端口。输出对象用于接收数据，所以仅包含输入端口。Mapplet 同时包含输入端口和输出端口。转换包含输入端口、输出端口和输入/输出端口的组合，具体取决于转换本身及其应用程序。

动态端口会从上游转换收到一个或多个列。动态端口会收到新列或已更改的列，具体取决于数据流。

在映射中创建转换之后，创建端口并定义端口属性。通过端口将映射链接到目标和其他转换，这样便可完成该映射。通过传播端口属性，可以将已更改的属性传递给映射中的端口。

创建端口

创建某些转换时，不必手动创建所有端口。例如，可以创建查找转换并引用查找表。如果查看转换端口，可以看到转换针对您引用的表中的每列都有一个输出端口。无需定义这些端口。

请按照以下方法创建端口：

- **从其他转换中拖动端口。**从其他转换中拖动端口时，Designer 将创建一个具有相同属性的端口，并链接起两个端口。
- **在“端口”选项卡上单击“添加”按钮。**Designer 将创建一个可以配置的空端口。

配置端口

定义转换端口时，可定义端口属性。端口属性包括端口名称、数据类型、端口类型和默认值。

配置以下端口属性：

- **端口名称。**端口的名称。命名端口时，请遵循以下约定：

  - 以单字节或双字节字母或者单字节或双字节的下划线 (_) 开头。

  - 端口名称可以包含以下任意单字节或双字节字符：字母、数字、下划线 (_)、$、# 或 @。
- **}数据类型、精度和小数位数。**如果您计划输入表达式或条件，请验证数据类型是否与表达式的返回值匹配。
- **端口类型。**转换可包含输入、输出、输入/输出和变量端口类型的组合。
- **默认值。**为包含空值或输出转换错误的端口分配一个默认值。可以替代某些端口中的默认值。
- **说明。**端口的说明。
- **其他属性。**有些转换具有特定于该转换的属性，例如表达式或按属性组合。

链接端口

在映射中添加和配置输入、输出、转换和 Mapplet 对象后，请通过链接映射对象之间的端口来完成映射。

只有在连接满足链接验证和串联要求的情况下，Developer 工具才会创建连接。

可以使端口保持未连接状态。 数据集成服务会忽略未连接的端口。

在输入对象、转换、Mapplet 和输出对象之间链接端口时，可以创建以下链接类型：

- 一对一
- 一对多

可以手动或自动链接端口。

一对一链接

将输入对象或转换中的一个端口链接到输出对象或转换中的一个端口。

一对多链接

要将相同数据用于不同的目的时，可以将提供该数据的端口链接到映射中的多个端口。

可以通过以下方式创建一对多链接：

- 将一个端口链接到多个转换或输出对象。
- 将一个转换中的多个端口链接到多个转换或输出对象。

例如，如果要使用薪酬信息通过汇总器转换来计算某个银行支行的平均薪酬， 则可以使用配置的表达式转换中的相同信息来计算每个员工的月薪。

手动链接端口

您可以手动链接一个或多个端口。

将输入对象或转换中的端口拖动到输出对象或转换的端口。

使用 Ctrl 或 Shift 键可选择多个要链接到其他转换或输出对象的端口。 Developer 工具将从顶部的端口对开始链接端口。 它将链接所有满足验证要求的端口。

将一个端口拖动到空端口时，Developer 工具将复制该端口，并创建链接。

自动链接端口

自动链接端口时，可按位置或名称进行链接。

按名称自动链接端口时，可以指定要按其链接端口的前缀或后缀。 使用前缀或后缀可以指示端口在映射中的位置。

链接端口（按名称）

按名称链接端口时，Developer 工具会在名称相同的输入端口和输出端口之间添加链接。 如果要在不同的转换中使用相同的端口名称，请按名称进行链接。

您可以基于定义的前缀和后缀链接端口。 使用前缀或后缀可以指示端口在映射中的位置。 如果要在端口名称中使用前缀或后缀以区别端口在映射或 Mapplet 中的位置，则可以按名称加前缀或后缀来链接端口。

按名称链接不区分大小写。

1. 单击**映射** > **自动链接**。

   此时将显示**自动链接**对话框。
2. 在**源**窗口中选择要从中进行链接的对象。
3. 在**目标**窗口中选择要链接到的对象。
4. 选择**名称**。
5. （可选）单击**显示高级**，基于前缀或后缀链接端口。
6. 单击**确定**。

链接端口（按位置）

按位置进行链接时，Developer 工具会将每个输出端口链接到相对应的输入端口。 例如，将第一个输出端口链接到第一个输入端口，将第二个输出端口链接到第二个输入端口。 如果转换中的相关端口顺序相同，则创建转换时，请按位置链接端口。

1. 单击**映射** > **自动链接**。

   此时将显示**自动链接**对话框。
2. 在**源**窗口中选择要从中进行链接的对象。
3. 在**目标**窗口中选择要链接到的对象。
4. 选择**位置**，然后单击**确定**。

   Developer 工具会将每个输出端口链接到相对应的输入端口。 例如，将第一个输出端口链接到第一个输入端口，将第二个输出端口链接到第二个输入端口。

链接端口的规则和准则

链接端口时，需要应用一定的规则和准则。

连接映射对象时，请考虑以下规则和准则：

- 如果您尝试在两个映射对象之间链接端口时 Developer 工具检测到错误，将显示一个符号，指示无法链接端口。
- 遵循映射中的数据流逻辑。 可以链接以下类型的端口：
  - 接收端口必须是输入端口或输入/输出端口。
  - 源端口必须是输出端口或输入/输出端口。
  - 不能将输入端口链接到输入端口，或将输出端口链接到输出端口。
- 必须至少将输入组中的一个端口链接到上游转换。
- 必须至少将输出组中的一个端口链接到下游转换。
- 可以将来自一个主动转换或主动转换的一个输出组中的端口链接到另一个转换的输入组。
- 不能将主动转换和被动转换连接到同一个下游转换或转换输入组。
- 不能将多个主动转换连接到同一个下游转换或转换输入组。
- 不能将任意数量的被动转换连接到同一个下游转换、转换输入组或目标。
- 如果同一转换的两个输出组中的数据均已排序，则可以将这两个组中的端口链接到已配置用于排序数据的联接器转换。
- 只能链接数据类型兼容的端口。 链接之前，Developer 工具将验证其是否可以在这两种数据类型之间进行映射。 数据集成服务无法在数据类型不兼容的端口之间转换数据。
- 如果某些映射违反了数据流验证，则 Developer 工具会将其标记为无效。

传播端口属性

通过传播端口属性，可以将更改的属性通过映射传递给端口。

1. 在编辑器中，选择某个转换中的端口。
2. 单击**映射** > **传播属性**。

   此时将显示**传播属性**对话框。
3. 选择传播属性的方向。
4. 选择要传播的属性。
5. （可选）预览结果。
6. 单击**应用**。

   Developer 工具将传播端口属性。

相关性类型

传播端口属性时，Developer 工具将更新相关性。

Developer 工具可以更新以下相关性：

- 链接路径相关性
- 隐式相关性

链接路径相关性

链接路径相关性是指传播端口与其链接路径中的端口之间的相关性。

在链接路径中传播相关性时，Developer 工具将更新其前向链接路径中的所有输入和输入/输出端口，以及其后向链接路径中的所有输出和输入/输出端口。 Developer 工具将执行以下更新：

- 在传播端口的链接路径中，更新所有端口的端口名称、数据类型、精度、小数位数和说明。
- 对于已更改端口名称的传播端口，更新所有引用该端口的表达式或条件。
- 如果关联的端口名称发生更改，则更新动态查找转换中的关联端口属性。

隐式相关性

隐式相关性是指位于两个基于表达式或条件的端口之间的转换中的相关性。

可以将数据类型、精度、小数位数和说明传播到具有隐式相关性的端口。 还可以解析条件和表达式以确定传播端口的隐式相关性。 所有具有隐式相关性的端口都是输出端口或输入/输出端口。

包含条件时，Developer 工具将更新以下相关性：

- 链接路径相关性
- 在相同查找条件中用作传播端口的输出端口
- 与传播端口关联的动态查找转换中的关联端口
- 在相同联接条件中用作详细端口的主端口

包含表达式时，Developer 工具将更新以下相关性：

- 链接路径相关性
- 包含使用了传播端口的表达式的输出端口

由于 Developer 工具不会传播到同一转换中的隐式相关性， 所以您必须传播其他转换中的更改属性。 例如，更改查找条件中所用端口的数据类型并传播查找转换中的此更改后，Developer 工具不会将此更改传播到其他依赖于该查找转换中的条件的端口。

传播端口属性（按转换）

Developer tool 将为每个转换传播相关性和属性。

下表介绍了 Developer tool 为每个转换传播的相关性和属性：

转换	相关性	传播的属性
地址验证器	无。	无。该转换具有预定义的端口名称和数据类型。
汇总器	- 链接路径中的端口 - 表达式 - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称 - 数据类型、精度、小数位数
关联	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
大小写转换器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
分类器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
比较	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明

转换	相关性	传播的属性
合并器	无。	无。该转换具有预定义的端口名称和数据类型。
数据屏蔽	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
数据处理器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
判定	- 链接路径中的下游端口	- 端口名称、数据类型、精度、小数位数、说明
表达式	- 链接路径中的端口 - 表达式 - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称 - 数据类型、精度、小数位数
筛选器	- 链接路径中的端口 - 条件	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称
层次结构到关系	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
联接器	- 链接路径中的端口 - 条件 - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称 - 数据类型、精度、小数位数
键生成器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
标签创建器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
查找	- 链接路径中的端口 - 条件 - 关联端口（动态查找） - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称 - 端口名称 - 数据类型、精度、小数位数
匹配	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
合并	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
规范器	- 链接路径中的端口	- 端口名称
解析器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
等级	- 链接路径中的端口 - 表达式 - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称 - 数据类型、精度、小数位数
读取		
REST Web 服务使用者	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
路由器	- 链接路径中的端口 - 条件	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称
序列生成器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
排序器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
SQL	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明

转换	相关性	传播的属性
标准创建器	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
联合	- 链接路径中的端口 - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 数据类型、精度、小数位数
更新策略	- 链接路径中的端口 - 表达式 - 隐式相关性	- 端口名称、数据类型、精度、小数位数、说明 - 端口名称 - 数据类型、精度、小数位数
加权平均值	- 链接路径中的端口	- 端口名称、数据类型、精度、小数位数、说明
写入		

从 Excel 复制端口

可以在 Excel 中配置端口和端口属性，然后将其复制到 Developer tool 中的转换端口。端口属性包括列名称、数据类型、精度和小数位数。当需要开发或编辑包含许多端口的转换时，您可以执行此操作。

可以将元数据复制到以下转换类型：

- 汇总器
- 表达式
- 筛选器
- Java
- 联接器
- 查找
- 规范器
- 等级
- 读取
- 路由器
- 序列
- 排序器
- SQL
- 联合
- 更新策略
- Web 服务使用者
- 窗口
- 写入

在 Excel 中编辑转换

当需要编辑转换的一大部分内容时，您不必在 Developer tool 中更改每个值。相反，您可以将转换端口复制到 Excel，使用“自动填充”同时更改所有值，然后将转换端口**粘贴(替换)** 回到 Developer tool 中。

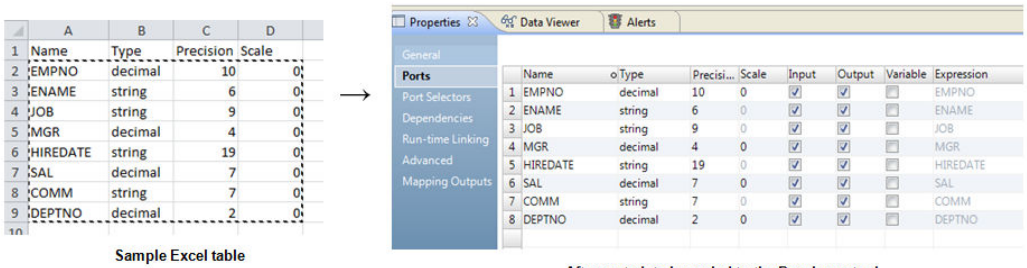
1. 在 Developer tool 的映射编辑器中，选择要从中复制端口的转换。
2. 要从 Developer tool 复制原始转换，请在“端口”中单击右键，然后单击**全选**。
3. 将端口复制到 Excel 电子表格。
4. 在 Excel 电子表格中进行更改。如果要更改元数据的一大部分内容，可以使用 Excel 中的“自动填充”功能。使用此功能时，您可以通过拖动填充柄，根据相邻单元格的值来填充数据。有关详细信息，请参阅“示例：在 Excel 中编辑转换” 页面上 63。
5. 从 Excel 复制元数据。
6. 要使用所做更改更新转换，请在“端口”中单击右键，然后单击**粘贴(替换)**。

将元数据复制到 Developer tool

可以在 Excel 中创建转换端口，然后将其复制到 Developer tool。

1. 在 Developer tool 中创建包含所需的转换的映射。
2. 在 Excel 中定义转换的元数据。
3. 从 Excel 复制元数据。
4. 要将元数据移动到 Developer tool 中的转换，请在“端口”中单击右键，然后单击**粘贴(替换)**。

   下图显示了一个示例 Excel 表以及将元数据复制到 Developer tool 之后生成的转换：



	A	B	C	D
1	Name	Type	Precision	Scale
2	EMPNO	decimal	10	0
3	ENAME	string	6	0
4	JOB	string	9	0
5	MGR	decimal	4	0
6	HIREDATE	string	19	0
7	SAL	decimal	7	0
8	COMM	string	7	0
9	DEPTNO	decimal	2	0

Sample Excel table

After metadata is copied to the Developer tool

注意: 将值复制到转换之前，必须先确认每个单元格中的值均有效。例如，字符串类型不能具有“0”以外的其他小数位数值。精度值不能是单词，类型值不能是数字。如果元数据不正确，将显示错误消息。

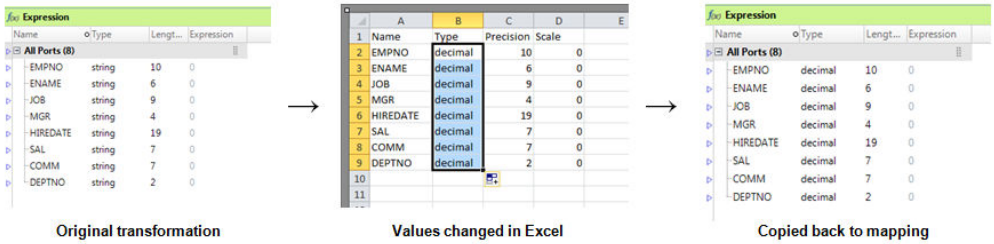
示例：在 Excel 中编辑转换

您要开发一个转换，并且需要将所有字符串数据类型更改为十进制。您可以在 Excel 中对所有字段进行全局更改并将字段复制到 Developer tool，而不必逐个更改字段。

1. 在“端口”中单击右键，单击**全选**，然后将元数据粘贴到 Excel 中。
2. 首先将数据类型值从“string”更改为“decimal”，然后使用填充柄自动更改列中的其余单元格。
3. 要使用所做更改更新转换，请从 Excel 复制元数据，在“端口”中单击右键，然后单击**粘贴(替换)**。

使用 Excel，您不必逐个更改字段。

下图显示了将转换移动到 Excel 中，使用“自动填充”更改某些值，然后将转换复制回 Developer tool 中的过程：



Original transformation **Values changed in Excel** **Copied back to mapping**

从 Excel 复制元数据的规则和准则

从 Excel 将元数据复制到 Developer tool 时，请考虑以下规则和准则：

- 将值复制到转换之前，必须先确认每个单元格中的值均有效。例如，字符串类型不能具有“0”以外的其他小数位数值。精度值不能是单词，类型值不能是数字。如果元数据不正确，将显示错误消息。
- 可将元数据复制到的位置取决于所更新的转换的类型。例如，在转换的“属性”视图中单击右键时，“粘贴(替换)”选项不可用。但是，在编辑器中直接在转换端口内单击右键时，此选项仍可用。

第 3 章

转换缓存

本章包括以下主题：

转换缓存概览

数据集成服务为映射中的汇总器转换、联接器转换、查找转换、等级转换和排序器转换分配高速缓存。数据集成服务将为汇总器转换、联接器转换、查找转换和等级转换创建索引缓存和数据缓存。数据集成服务将为排序器转换创建一个缓存。

您可以配置这些转换的缓存大小。缓存大小决定数据集成服务在映射运行开始时为每个转换缓存分配的内存量。

如果缓存大小大于计算机的可用内存，数据集成服务将无法分配足够内存，映射运行将失败。

如果缓存大小小于运行转换所需的内存量，数据集成服务将在内存中处理部分转换，并将溢出数据存储在缓存文件中。数据集成服务将缓存文件分页到磁盘时，会增加处理时间。为实现最佳性能，请配置适当的缓存大小，使数据集成服务可以在内存中处理完整的转换。

默认情况下，数据集成服务根据可分配的内存上限自动计算运行时的内存需求。在自动缓存模式下运行映射后，您可以调整转换的缓存大小。分析映射日志中的转换统计数据，以确定实现最佳性能所需的缓存大小，然后为转换配置特定的缓存大小。

缓存类型

汇总器转换、联接器转换、查找转换和等级转换需要使用索引缓存和数据缓存。数据集成服务在索引缓存中存储键值，在数据缓存中存储输出值。排序器转换需要一个单独的缓存。数据集成服务将排序键和要存储的数据存储在排序器缓存中。

下表介绍了数据集成服务在每个缓存中存储的信息类型：

映射对象	缓存类型和说明
汇总器	- 索引。 按照分组依据端口中的配置存储组值。 - 数据。 基于分组依据端口存储计算。
联接器	- 索引。 将具有唯一键的所有主行存储在联接条件中。 - 数据。 存储主源行。
查找	- 索引。 存储查找条件信息。 - 数据。 存储未存储在索引缓存中的查找数据。
等级	- 索引。按照分组依据端口中的配置存储组值。 - 数据。 根据分组依据端口存储等级信息。
排序器	- 排序器。 存储排序键和数据。

缓存文件

当您运行映射时，数据集成服务至少为每个汇总器转换、联接器转换、查找转换、等级转换和排序器转换创建一个缓存文件。如果数据集成服务无法在内存中运行转换，会将溢出数据写入缓存文件。

下表介绍了数据集成服务为不同映射对象创建的缓存文件类型：

映射对象	缓存文件
汇总器、联接器、查找和等级转换	数据集成服务会创建以下类型的缓存文件： - 为每个索引缓存和数据缓存创建一个表头文件 - 为每个索引缓存和数据缓存创建一个数据文件
排序器转换	数据集成服务会创建一个排序器缓存文件。

当您运行映射时，数据集成服务会在映射日志中写入一条消息，指示缓存文件名称和转换名称。映射完成后，数据集成服务会释放高速缓存，并且通常会删除缓存文件。在以下情况下，您可以在缓存目录中找到索引缓存文件和数据缓存文件：

- 配置查找转换以使用持久性缓存。
- 映射未成功完成。下次运行映射时，数据集成服务将删除现有缓存文件并创建新的缓存文件。

由于向缓存文件写入数据可能会降低映射性能，因此请配置适当的缓存大小以便在内存中运行转换。

缓存文件目录

对于汇总器转换、联接器转换、查找转换和等级转换，数据集成服务将在“缓存目录”属性中指定的目录中创建缓存文件。对于排序器转换，服务在“工作目录”属性中指定的目录中创建缓存文件。

如果数据集成服务进程找不到该目录，将使映射失败并向映射日志中写入一条消息，指示无法创建或打开缓存文件。数据集成服务可能会创建多个缓存文件。缓存文件的数量受缓存目录中的可用磁盘空间量的限制。

缓存大小

缓存大小决定数据集成服务在映射运行开始时为每个转换缓存分配的内存大小。您可以将转换缓存大小配置为使用自动缓存模式或是使用特定值。

自动缓存大小

默认情况下，转换的缓存大小设置为“自动”。数据集成服务将在运行时自动计算高速缓存需求。您定义服务可分配的最大内存量。

数据集成服务遵循以下准则来自动分配内存：

为需要更长处理时间的转换分配更多内存。

数据集成服务将为通常需要更长处理时间的转换分配更多内存。例如，数据集成服务会为排序器转换分配更多内存，因为排序器转换通常运行时间更长。

为数据缓存分配比索引缓存更多的内存。

汇总器转换、联接器转换、查找转换和等级转换需要使用索引缓存和数据缓存。数据集成服务在索引缓存和数据缓存间划分分配给转换的内存时，将为数据缓存分配更多内存。

排序器转换需要一个单独的缓存。服务会将分配给转换的所有内存分配给排序器缓存。

自动缓存大小的内存上限

您在 Administrator 工具中数据集成服务模块的“每个请求的内存上限”属性中定义数据集成服务可分配给转换缓存的最大内存量。

每个模块运行不同类型的请求，这些请求具有不同的内存要求。例如，映射和配置文件请求通常比 SQL 服务或 Web 服务请求需要更多高速缓存。您可以配置以下数据集成服务模块的“每个请求的内存上限”属性：

- 映射服务模块
- 剖析服务模块
- SQL 服务模块
- Web 服务模块

注意: 映射服务模块请求包括来自工作流内映射任务的映射和映射运行。

对于剖析服务模块，“每个请求的内存上限”定义了数据集成服务可为单个配置文件请求的每次映射运行分配的最大内存量。

对于其余模块，“每个请求的内存上限”的行为取决于数据集成服务的配置。该行为取决于数据集成服务上的“启动作业选项”属性和“内存大小上限”属性。

下表根据数据集成服务配置介绍了映射、SQL 服务和 Web 服务模块的“每个请求的内存上限”的行为：

数据集成服务配置	“每个请求的内存上限”的行为
在单独的本地或远程操作系统进程中运行作业，或“内存大小上限”为 0（默认值）	数据集成服务可为单个请求中所有使用自动缓存模式的转换分配的最大内存量（以字节为单位）。 为“每个请求的内存上限”定义的值仅影响使用自动缓存模式的转换。数据集成服务为配置了特定缓存大小的转换单独分配内存。请求所用的总内存可以超过“每个请求的内存上限”值。 例如，“每个请求的内存上限”设置为 800 MB。一个映射具有三个需要缓存的转换。您将其中两个转换配置为使用自动缓存模式，将第三个转换配置为使用总共 500 MB 的缓存大小。数据集成服务将为所有转换缓存分配总共 1,300 MB 内存。
在数据集成服务进程中运行作业，并且“内存大小上限”大于 0	数据集成服务可以为单个请求分配的最大内存量（以字节为单位）。 为“每个请求的内存上限”属性定义的值会影响所有转换。请求所用的总内存不可超过“每个请求的内存上限”值。

当您增加自动缓存模式所用的最大内存量时，会增加可用于模块所有请求的缓存大小上限。您可以增加最大内存量，以确保不会有缓存文件被分页到磁盘。但是，因为此值用于所有请求，所以数据集成服务为某些请求分配的内存可能会多于所需内存。

特定缓存大小

您可以为转换配置特定缓存大小。数据集成服务将在映射运行开始时向转换缓存分配指定的内存量。在调整缓存大小时配置特定值（以字节为单位）。

首次配置缓存大小时，请使用自动缓存模式。运行映射后，分析映射日志中的转换统计信息，以确定在内存中运行转换所需的缓存大小。将缓存大小配置为使用映射日志中指定的值时，可确保不会浪费分配的内存。但是，最佳缓存大小因源数据大小而异。请在后续映射运行后查看映射日志，以监视缓存大小的变化。如果您为可重用转换配置特定的缓存大小，请确认该缓存大小是否对于映射中每次使用转换时都是最佳大小。

要定义特定的缓存大小，请在 Developer tool 中的转换属性中配置缓存大小值。

通过数据集成服务增加缓存大小

数据集成服务会根据配置的缓存大小创建各个内存缓存。在某些情况下，因为需要更多高速缓存，数据集成服务可能会增加配置的缓存大小。

数据集成服务可能会因为以下原因之一而增加配置的缓存大小：

配置的缓存大小低于处理操作所需的最小缓存大小。

数据集成服务需要一定的最小内存量来对每个映射进行初始化。如果配置的缓存大小低于所需的最小缓存大小，数据集成服务将增加配置的缓存大小来满足最低要求。如果数据集成服务无法分配所需的最小内存量，映射将失败。

配置的缓存大小不是缓存页大小的倍数。

数据集成服务在缓存页中存储缓存的数据。缓存页面必须均匀放入缓存。例如，如果将缓存大小配置为 10 MB（1,048,576 个字节），缓存页大小为 10,000 个字节，则数据集成服务会将配置的缓存大小增加到 1,050,000 个字节，使其达到 10,000 个字节的页面大小的倍数。

增加配置的缓存大小后，数据集成服务会继续运行映射，并在映射日志中写入以下消息：

```
INFO: MAPPING,    TE_7212,    Increasing [Index Cache] size for transformation <transformation name> from
<configured cache size> to <new cache size>.
INFO: MAPPING,    TE_7212,    Increasing [Data Cache] size for transformation <transformation name> from
<configured cache size> to <new cache size>.
```

已分区缓存的缓存大小

如果您有分区选项，缓存分区功能将为每个运行汇总器转换、联接器转换、等级转换、查找转换或排序器转换的分区创建一个单独的缓存。在缓存分区过程中，每个分区在单独的缓存中存储不同的数据。当数据集成服务为这些转换使用缓存分区时，服务将在分区间划分所分配的缓存大小。

例如，您将转换的缓存大小配置为 100 MB。数据集成服务使用四个分区运行转换。服务将划分缓存大小值，使每个分区最多使用 25 MB 的缓存大小。

缓存大小优化

要获得最佳映射性能，请配置缓存大小以使数据集成服务可以在内存中运行完整转换。

要配置最佳缓存大小，请执行以下任务：

1. 将跟踪级别设置为详细初始化。
2. 在自动缓存模式下运行映射。
3. 在映射日志中分析缓存性能。
4. 为缓存大小配置特定值。

步骤 1. 将跟踪级别设置为详细初始化

在 Developer tool 中，将跟踪级别设置为详细初始化，以使数据集成服务可以将转换统计信息写入映射日志。转换统计信息会列出达到最佳性能所需的缓存大小。默认情况下，跟踪级别设置为普通。

使用以下方式之一将跟踪级别设置为详细初始化：

- 修改每个使用缓存的转换的高级属性。
- 如果计划第一次从 Developer tool 运行映射，请修改默认的映射配置属性。有关详细信息，请参阅*《Informatica Developer tool 指南》*。
- 如果计划第一次从命令行运行已部署的映射，请修改包含该映射的应用程序的高级属性。有关详细信息，请参阅*《Informatica Developer tool 指南》*。

步骤 2. 在自动缓存模式下运行映射

首次运行映射时，为转换缓存大小使用自动缓存模式。

您可以从 Developer tool 运行映射。也可以将映射添加到应用程序，然后将该应用程序部署到数据集成服务，从而可以从命令行运行该映射。

步骤 3. 分析缓存性能

在自动缓存模式下运行映射后，分析映射日志中的转换统计信息，以确定达到映射最佳性能所需的缓存大小。

当汇总器转换、联接器转换、查找转换或等级转换分页到磁盘时，映射日志将指出在内存中运行转换所需的索引缓存大小和数据缓存大小。例如，您运行一个名为 AGG_TRANS 的汇总器转换。映射日志包含以下文本：

```
CMN_1791, The index cache size that would hold [1098] aggregate groups of input rows for [AGG_TRANS], in memory, is [286720] bytes
CMN_1790, The data cache size that would hold [1098] aggregate groups of input rows for [AGG_TRANS], in memory, is [1774368] bytes
```

该日志显示索引缓存和数据缓存分别需要 286,720 个字节和 1,774,368 个字节才能在内存中运行转换而不分页到磁盘。

当排序器转换分页到磁盘时，映射日志会显示数据集成服务对源数据执行了多次传递。当数据集成服务必须分页到磁盘才能完成排序时，它会对数据执行多次传递。该消息将指出单次传递（即数据集成服务一次性读取数据并在内存中执行排序而不分页到磁盘）所需的字节数。

例如，您运行一个名为 SRT_TRANS 的排序器转换。映射日志包含以下文本：

```
SORT_40427,    Sorter Transformation [SRT_TRANS] required 2-pass sort (1-pass temp I/O: 13126221824 bytes). You may try to set the cache size to 14128 MB or higher for 1-pass in-memory sort.
```

日志显示排序器缓存需要 14,128 MB 才能使数据集成服务对数据执行一次传递。

步骤 4. 配置特定缓存大小

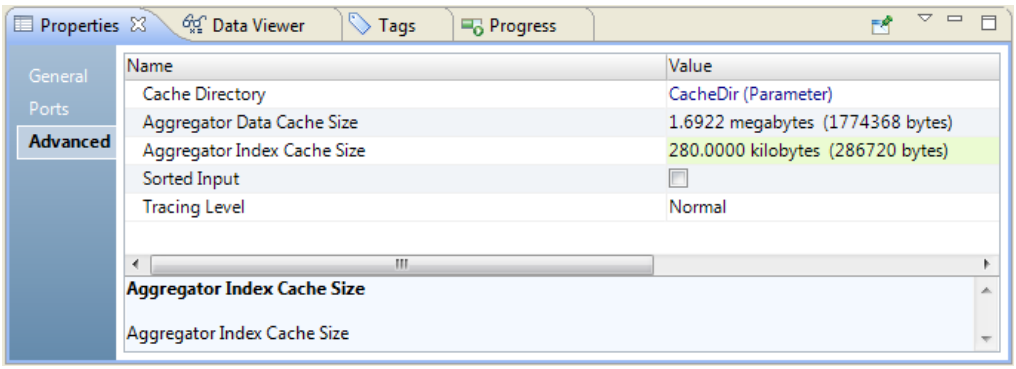
为实现最佳性能，请将转换缓存大小配置为使用映射日志中指定的值。在 Developer tool 中更新索引缓存和数据缓存大小转换属性。

1. 在 Developer tool 中，打开可重用或不可重用转换。
2. 根据以下转换类型找到缓存大小属性：

选项	说明
可重用的汇总器转换、联接器转换、等级转换或排序器转换	单击**高级**视图。
不可重用的汇总器转换、联接器转换、等级转换或排序器转换	单击**属性**视图中的**高级**选项卡。
可重用的查找转换	单击**运行时**视图。
不可重用的查找转换	单击**属性**视图中的**运行时**选项卡。

3. 输入映射日志为索引缓存和数据缓存大小建议的值（以字节为单位）。

   下图显示了一个不可重用的汇总器转换，它为索引缓存和数据缓存大小配置了特定值。



4. 单击**文件** > **保存**。

第 4 章

地址验证器转换

本章包括以下主题：

地址验证器转换概览

地址验证器转换是一种多组转换，可比较输入地址数据和地址引用数据。它决定地址的准确性，并修复地址中的错误。该转换将创建符合数据内容和结构的邮寄投递标准的记录。该转换还会在每个地址中添加状态信息。

地址验证器转换对地址数据执行以下操作：

- 该转换比较源数据中的地址记录和地址引用数据中的地址定义。

- 该转换生成详细状态报表，该报表有关每个输入地址的有效性、可投递状态和所含的任何错误或歧义的性质。
- 修复错误并完成部分地址记录。要修复地址，转换必须找到引用数据中地址的正向匹配。该转换将所需数据元素从地址引用数据复制到地址记录。
- 添加不以标准地址格式显示但有助于邮政投递的信息，如收件人地点信息和地理编码信息。
- 其以数据对象和邮件运营商要求的格式写入输出地址。在转换上选择输出端口时定义该格式。

地址引用数据

地址引用数据集描述国家邮件运营商能够识别的属于某个国家/地区的地址。在通过地址验证器转换执行地址验证之前，在域中的 Informatica 服务计算机上安装地址引用数据。可从 Informatica 购买并下载地址引用数据。

为源地址数据所标识的每个国家/地区安装地址引用数据文件。具有大型社群的国家/地区可能需要多个文件。此外，还可以安装数据文件来补充或扩充地址数据。邮件运营商可以使用数据浓缩来认证地址的准确性并加快邮件送达速度。

执行地址验证时，地址验证器转换会将每个输入记录与地址引用数据相比较。如果转换在地址引用数据中找到输入地址，则该转换可以使用正确且完整的地址数据来更新记录。如果购买了额外的引用数据集，转换还可以扩充地址数据。

使用 Developer 工具中的**首选项**窗口查看有关域中 Informatica 服务计算机上的地址引用数据文件的信息。

地址引用数据的类型

所选验证模式决定转换如何将输入地址与地址引用数据相比较。

地址验证器转换可以读取以下类型的地址引用数据：

地址代码查找数据

安装地址代码查找数据以从输入端口的代码值中检索部分地址或完整地址。地址的完整性取决于地址所属的国家/地区中的地址代码支持级别。要从输入地址读取地址代码，请在“离散值”端口组中选择国家/地区特定的端口。

您可以为以下国家/地区选择端口：

- 奥地利返回建筑物级别的地址。
- 德国。返回区域、行政区或街道级别的地址。
- 日本。返回唯一邮箱级别的地址。
- 南非。返回街道级别的地址。
- 韩国。返回唯一邮箱级别的地址。
- 塞尔维亚共和国。返回街道级别的地址。
- 英国。返回唯一邮箱级别的地址。

如果将地址验证器转换配置为在地址代码查找模式下运行，该转换将读取地址代码查找数据。

批处理数据和交互数据

安装批处理数据和交互数据可对一组地址记录执行地址验证。使用批处理数据和交互数据可以根据来自国家邮件运营商的当前邮政数据验证输入地址是否完全可投递以及是否完整。

如果将地址验证器转换配置为在批处理模式下运行，则该转换将为每个输入地址返回一个地址。如果将地址验证器转换配置为在交互模式下运行，则该转换将为每个输入地址返回一个或多个地址。

CAMEO 数据

安装 CAMEO 数据可将客户细分数据添加到住宅地址记录中。客户细分数据指示每个地址的居民的可能收入水平和生活方式首选项。

如果将地址验证器转换配置为在批处理模式或认证模式下运行，该转换将读取 CAMEO 数据。

认证数据

安装认证数据可验证地址记录是否满足邮件运营商规定的认证标准。如果地址包含的数据元素可以标识唯一的邮箱（如收件人地点数据元素），则该地址符合认证标准。如果地址符合认证标准，则邮件运营商收取的邮费会减少。

以下国家/地区规定的认证标准：

- 澳大利亚。根据地址匹配审批制度 (Address Matching Approval System, AMAS) 标准认证邮件。
- 加拿大。根据软件评估和识别程序 (Software Evaluation And Recognition Program, SERP) 标准认证邮件。
- 法国。根据国家地址管理服务 (National Address Management Service, SNA) 标准认证邮件。
- 新西兰。根据 SendRight 标准认证邮件。
- 美国。根据编码准确性支持系统 (Coding Accuracy Support System ,CASS) 认证邮件。

如果将地址验证器转换配置为在认证模式下运行，该转换将读取认证数据。

地理编码数据

安装地理编码数据可将地理编码添加到地址记录中。地理编码是纬度坐标和经度坐标。

如果将地址验证器转换配置为在批处理模式或认证模式下运行，该转换将读取地理编码数据。

建议列表数据

安装建议列表数据可找到部分地址记录的有效备用版本。配置地址验证映射以便实时逐个处理地址记录时，使用建议列表数据。地址验证器转换使用部分地址中的数据元素对建议列表数据执行重复项检查。该转换将返回在部分地址中包含该信息的任何有效地址。

如果将地址验证器转换配置为在建议列表模式下运行，该转换将读取建议列表数据。

补充数据

安装补充数据可将数据添加到地址记录，以帮助邮件运营商进行邮件投递。使用补充数据可添加有关包含该地址的地理或邮政区域的详细信息。在一些国家/地区，补充数据可提供邮政系统内邮箱的唯一标识符。

如果将地址验证器转换配置为在批处理模式或认证模式下运行，该转换将读取补充数据。

注意: 在国家/地区识别模式或解析模式下，转换不会读取地址引用数据。

相关主题：

- “ 地址验证器转换常规设置” 页面上 91

模式和模板

配置地址验证器转换时，可以选择转换执行的验证类型。转换将验证类型定义为模式。可以在**常规设置**选项卡上选择模式，也可以选择模式作为转换的高级属性。

可以在转换中创建端口模板。模板是一个或多个端口组的端口子集。可以使用模板对要在项目中使用的端口进行组织。

端口组和端口选择

地址验证器转换包含预定义的端口组，这些端口组中包含可以使用的输入端口和输出端口。 配置地址验证器转换时，浏览这些组，然后选择所需的端口。

选择与地址输入数据的结构对应的输入端口。选择包含项目所需地址数据的输出端口。

可以直接将输入端口和输出端口添加到转换中，也可以创建包含输入端口和输出端口的默认模型。直接将端口添加到转换中时，选定端口仅应用到该转换。将端口添加到默认模型时，选定端口会应用到您将来创建的地址验证器转换。

对于不希望地址验证器转换处理的列，还可以将传递端口添加到转换中。

地址验证器转换输入端口组

先浏览输入组并选择与输入数据的结构和内容对应的端口，然后才能将地址数据连接到转换上的输入端口。 浏览输出组，然后选择与您的数据要求匹配的端口。

地址验证器转换将显示基本模型和高级模型中的端口组。 您可以使用基本模型中的端口组定义大多数地址。 如果地址非常复杂，请使用高级模型中可用的其他端口。

注意: 仅从一个输入端口组中选择端口。

转换具有以下输入端口组：

离散值

使用“离散值”端口读取包含单个数据元素（如住宅门牌号、街道或邮政编码）完整信息的数据列。在基本模型和高级模型中查找“离散值”组。

混合

使用“混合”端口读取包含一个或多个数据元素信息的数据列。“混合”组将“离散值”组和“多行”组中的端口组合在一起。 使用“混合”端口可创建可以提交至邮件运营商的地址记录。 “混合”端口按邮件运营商的标准构建地址，并标识每一行的数据类型。 在基本模型和高级模型中查找“混合”组。

多行

使用“多行”端口读取包含多个数据元素的数据列。每个输入列对应一行地址。为了取得最佳效果，请按照邮政运营商要求的格式定义输入数据。选择“多行”端口以创建一组可打印的地址记录。

每个“多行”端口均表示可打印地址中的一行，例如以下街道数据行：

```
123 Main Street, Apartment 2
```

“多行”端口不会指定显示在每个地址行上的数据类型。 在基本模型和高级模型中查找“多行”组。

单行

使用“单行”端口可读取具有以下特征的单个数据列：该单个数据列包含直至省级别的全部地址元素，并且各元素之间不含分隔符。使用端口组中的“完整地址”端口来提交地址元素。端口组还包括一个可用于读取地址的国家/地区信息的“国家/地区”端口。在基本模型和高级模型中查找“单行”组。

地址验证器转换输出端口组

先确定所需的信息类型和输出地址将采取的结构，然后才能将地址验证器转换连接到其他转换或数据对象。

浏览输出组，然后选择与您的数据要求匹配的端口。

注意: 可以从多个输出组中选择端口，可以选择具有常见功能的端口。

转换具有以下预定义的输出组：

地址元素

将街道数据元素（如门牌号、公寓号和街道名称）写入各个端口。在基本模型和高级模型中查找地址元素组。

奥地利补充

向奥地利地址写入有助于邮政投递的数据，例如建筑物级别的邮政编码数据。在基本模型中查找“奥地利补充”组。

澳大利亚补充

向澳大利亚地址写入数据，以标识澳大利亚统计局为其分配地址的地理区域。在基本模型中查找“澳大利亚补充”组。

特定于澳大利亚

向澳大利亚地址写入数据，使地址符合澳大利亚邮政的地址匹配审批制度 (AMAS) 标准。在基本模型和高级模型中查找“特定于澳大利亚”组。

比利时补充

向比利时地址写入有助于邮政投递的数据。这些数据包括比利时国家统计局提供的区域和社区标识码。在基本模型中查找“比利时补充”组。

巴西补充

向巴西地址写入有助于邮政投递的数据，例如地理统计局 (IBGE) 的地区标识代码。在基本模型中查找“巴西补充”组。

CAMEO

生成可以在客户细分分析中使用的人口和收入摘要数据。在基本模型中查找 CAMEO 组。

特定于加拿大

向加拿大地址写入数据，使地址符合加拿大邮政的软件评估和识别程序 (SERP) 标准。在基本模型中查找“特定于加拿大”组。

瑞士补充

向瑞士地址写入有助于邮政投递的数据，例如扩展邮政编码数据。在基本模型中查找“瑞士补充”组。

捷克补充

向捷克共和国地址写入有助于邮政投递的数据，例如扩展邮政编码数据。在基本模型中查找“捷克补充”组。

联系人元素

写入个人或联系人数据，如姓名、称呼和职称。在高级模型中查找“联系人元素”组。

国家/地区

写入国家/地区名称或由 International Organization for Standardization (ISO) 定义的国家/地区代码。在基本模型和高级模型中查找国家/地区组。

德国补充

向德国地址写入有助于邮政投递的数据，例如行政区和地区代码数据。在基本模型中查找“德国补充”组。

西班牙补充

向西班牙地址写入有助于邮政投递的数据。在基本模型中查找“西班牙补充”组。

格式化的地址行

写入设置了格式以用于打印和发送电子邮件的地址。在基本模型和高级模型中查找“格式化的地址行”组。

法国补充

向法国地址写入有助于邮政投递的数据，例如来自法国国家统计及经济研究局 (INSEE) 的标识代码。在基本模型中查找“法国补充”组。

特定于法国

向法国地址写入数据，使地址符合法国邮政的国家地址管理服务 (SNA) 标准。在基本模型中查找“特定于法国”组。

地理编码

为地址生成地理代码数据，如经度和纬度坐标。在基本模型中查找地理编码组。

ID 元素

写入记录 ID 和交易密钥数据。在高级模型中查找 ID 元素组。

意大利补充

向意大利地址写入有助于邮政投递的数据。在基本模型中查找“意大利补充”组。

日本补充

向日本地址写入有助于邮政投递的数据，例如 Choumei Aza 代码。在基本模型中查找“日本补充”组。

韩国补充

向韩国地址写入有助于邮政投递的数据，例如可用于指定给定地址的当前和非当前版本的唯一标识符。在基本模型中查找“韩国补充”组。

末行元素

写入可以显示在国内地址最后一行的数据。在基本模型和高级模型中查找“末行元素”组。

特定于新西兰

向新西兰地址写入数据，使地址符合新西兰邮政的 SendRight 标准。在基本模型中查找“特定于新西兰”组。

波兰补充

向波兰地址写入有助于邮政投递的数据，例如区域部门 (TERYT) 数据。在基本模型中查找“波兰补充”组。

剩余

写入转换无法解析到其他端口的数据元素。在基本模型和高级模型中查找“剩余”组。

塞尔维亚补充

向塞尔维亚地址写入有助于邮政投递的数据，例如邮政编码后缀数据。在基本模型中查找“塞尔维亚补充”组。

俄罗斯补充

向俄罗斯地址写入有助于邮政投递的数据，例如地址的联邦信息地址系统标识符。在基本模型中查找“俄罗斯补充”组。

状态信息

生成每个输入和输出地址的质量的详细数据。在基本模型中查找状态信息组。

英国补充

向英国地址写入有助于邮政投递的数据，例如收件人地点数据和英国地形测量局数据。在基本模型中查找“英国补充”组。

特定于美国

向美国地址写入数据，使地址符合美国邮政服务的编码准确性支持系统 (CASS) 标准。在基本模型中查找“特定于美国”组。

美国补充

写入地理和人口数据，如美国地址的联邦信息处理标准 (FIPS) 代码。在基本模型中查找“美国补充”组。

XML

以地址验证软件库定义的 XML 结构写入地址记录数据。在高级模型中查找 XML 组。

南非补充

向南非地址写入有助于邮政投递的数据，例如国家地址数据库数据。在基本模型中查找“南非补充”组。

多实例端口

多种类型的地址数据可以在一个地址中出现多次。 当地址中包含多种形式的数据元素时，可以选择端口的多个实例。

多实例端口中最多可包含六个实例。 许多地址对其包含的每个数据元素使用一个端口实例。部分地址使用端口的其他实例。 较小地址集使用多个端口实例。

通常，端口的第一个实例是该端口识别的主要名称或最大区域。对于选定的任何端口，必须验证端口实例之间的关系。

街道填写端口示例

英国的地址记录可以包含两个街道名，其中一个街道是较大街道平面图的一部分。

下表包含使用两个“街道填写”端口的地址：

端口	数据
街道编号填写 1	1A
街道填写 1	THE PHYGTLE
街道填写 2	SOUTH STREET
区域名称 1	NORFOLK
邮政编码 1	NR25 7QE

在本示例中，“街道填写 1”中的街道数据与“街道填写 2”中的街道数据相关。街道编号填写 1 中的数据将引用街道填写 1 中的数据。

注意: 尽管“街道填写 1”指定了邮箱位置，但“街道填写 2”可能是更大的街道。

联系人端口示例

当联系人是同一住户的成员时，地址记录可包含多个联系人。

下表包含使用两个“联系人姓名”端口的地址：

端口	数据
联系人姓名 1	MR.JOHN DOE
联系人姓名 2	MS. JANE DOE
格式化的地址行 1	2 MCGRATH PLACE EAST
格式化的地址行 2	ST. JOHN'S NL A1B 3V4
格式化的地址行 3	CANADA

在本示例中，组织可以确定应用到“联系人姓名 1”或“联系人姓名 2”的优先级。地址验证器转换不会确定联系人数据的优先级。

如果设置打印输出的地址格式，则可以使用“格式化的地址行”端口的多个实例。最多可以选择 12 个“格式化的地址行”端口。

地址验证项目

可以在多种类型的项目中使用地址验证器转换。为每个项目类型创建具有不同端口的地址模板。

可以定义具有下列一个或多个目标的地址验证项目：

创建符合邮件运营商标准的格式化地址

可以为邮件营销活动准备大型地址记录集。如果使用邮件运营商偏好的格式创建地址，则您的邮件成本将显著减少。准备邮寄的地址时，请选择将格式化地址的每一行写入单个端口的输出端口。可以为联系人姓名、街道地址行以及区域和邮政编码行选择不同的端口。

根据收入和生活方式指标组织地址

可以将客户细分数据添加到住宅地址记录中。客户细分数据指示每个地址的居民的可能收入水平和生活方式首选项。从 CAMEO 输出组中选择端口，以便将客户细分数据添加到地址记录中。可以使用面向多个客户市场的邮件营销活动中的客户细分数据。

创建邮件运营商认证的地址

为澳大利亚邮政、加拿大邮政或美国邮政服务 (USPS) 准备记录集时，可以添加确认每个地址交付性的数据。

地址验证器转换可以生成报表以确认地址记录的完整性和准确性是否符合每个邮件运营商的数据标准。

创建符合法规要求的地址

您可以验证组织保留的地址记录是否符合行业监管或政府法规。选择将每个地址数据元素写入单独字段的输出端口。另外，选择地址验证状态端口，这些端口提供了有关输出数据准确性和完整性的详细信息。

完成部分地址

您可以输入部分地址并在引用数据中检索与部分地址匹配的有效完整地址。要完成部分地址，请将转换配置为在建议列表模式或交互模式下运行。您可通过单行形式在“完整地址”端口上键入输入地址。

改进地址的数据质量

与其他数据项目并行，您可以改进地址数据集的结构和常规数据质量。例如，数据集包含的列数可能超出您的需求，或者可能在多个列中包含相同类型的数据。可以减少数据集中的列数，还可以简化为不同类型的数据使用的列。

格式化的地址和邮件运营商标准

为邮件营销活动准备地址记录时，请创建与邮件运营商的格式设置标准匹配的可打印地址结构。

例如，USPS 为美国国内地址保留以下地址格式：

Line 1	**Person/Contact Data**	**JOHN DOE**
Line 2	Street Number, Street, Sub-Building	123 MAIN ST NW STE 12
Line 3	Locality, State, ZIP Code	ANYTOWN NY 12345

可以定义将地址的每一行写入单个端口的可打印地址格式。可以使用可识别每一行上的数据类型的端口，也可以使用可填充地址结构（无论每一行上的数据如何）的端口。

下表显示设置美国地址格式以用于打印的不同方式：

对于此地址	使用以下端口	或使用以下端口
JOHN DOE	收件人行 1	格式化的地址行 1
123 MAIN ST NW STE 12	收件人地址行 1	格式化的地址行 2
ANYTOWN NY 12345	特定于国家/地区的末行 1	格式化的地址行 3

当数据集包含不同类型的地址（例如办公地址和住宅地址）时，使用“格式化的地址行”端口。办公地址的联系人和组织数据可能需要三个地址行。地址验证器转换可确保仅在需要办公地址或住宅地址时，每个地址才使用“格式化的地址行”端口来正确设置格式。但是，“格式化的地址行”端口不标识其包含的数据类型。

当所有地址遵循同一种格式时，请使用“收件人行”、“收件人地址行”和“特定于国家/地区的末行”端口。“收件人行”、“收件人地址行”和“特定于国家/地区的末行”端口将根据信息类型分隔地址数据元素，使数据集更易于理解。

注意: 可以选择其他端口来处理此地址。本示例重点介绍为打印和交付而设置地址格式的端口。

人口统计学和地理数据

为邮件营销活动创建记录集时，可以添加多种类型的可能不会显示在地址中的数据。使用此数据查看邮件的人口统计学和地理分布。

例如，可以标识美国地址所属的选举区。如果目标国家/地区的邮件系统引用数据中包含坐标，则还可以生成纬度和经度坐标。

部分地址填写

使用建议列表模式或交互模式时，可以输入不完整的地址，并从引用数据中检索有效的完整地址。

如果对地址不确定，并且希望查看有效候选地址列表，请选择建议列表模式。如果对地址很确定，并且希望验证地址的完整形式，请选择交互模式。在这两种情况下，地址验证器转换都会搜索地址引用数据，并返回包含输入数据的所有地址。

将转换配置为在建议列表模式或交互模式下运行时，请注意以下规则和指导：

- 您可以在多个端口上定义输入地址，也可以在“完整地址”输入端口上输入所有地址元素。
- 在建议列表模式下配置转换时，请从“离散值”输入组选择端口。或者，选择“完整地址”端口，并选择性地从“多行”组中选择“国家/地区名称”端口。
- 建议列表模式和交互模式可以为每个输入地址返回多个地址。“最大结果计数”属性指定返回地址数的上限。如果匹配地址的数量大于“最大结果计数”值，“计数溢出”端口会返回额外地址的数量。
- Informatica 地址验证将建议列表模式称为快速填写模式。

地址验证器状态端口

地址验证器转换可在地址元素中写入其在输入端口和输出端口上读取和写入的状态信息。使用状态信息端口查看状态信息。

可以选择以下状态端口：

地址解析代码

描述地址中的非有效地址元素。在基本模型的“状态信息”端口组中选择端口。

地址类型

在邮件运营商能够识别多个地址格式的情况下指出地址类型。在基本模型的“状态信息”端口组中选择端口。

元素输入状态

描述输入地址元素与引用数据之间的相似程度。在基本模型的“状态信息”端口组中选择端口。

元素相关性

标识邮件运营商标识地址邮箱所需的地址元素。在基本模型的“状态信息”端口组中选择端口。

元素结果状态

描述地址验证对输入地址进行的任何更新。在基本模型的“状态信息”端口组中选择端口。

扩展元素结果状态

指出引用数据中的地址存在额外数据。该端口可包含有关地址验证对地址进行的更新的详细信息。在基本模型的“状态信息”端口组中选择端口。

地理编码状态

描述地址验证为地址返回的地理编码数据的类型。在基本模型的“地理编码”端口组中选择端口。

可邮寄得分

指示邮件运营商可将邮件送达地址的可能性。在基本模型的“状态信息”端口组中选择端口。

匹配代码

描述地址验证结果以及对输入地址的地址更正操作。在基本模型的“状态信息”端口组中选择端口。

结果百分比

用百分比形式表示输入地址与对应的输出地址之间的相似度。在基本模型的“状态信息”端口组中选择端口。

元素状态代码定义

“元素输入状态”、“元素相关性”、“元素结果状态”和“扩展元素结果状态”端口提供了有关输入和输出数据元素有效性的状态信息。 选择元素端口可查看地址有效性操作的结果。

这些代码包含以下信息：

- “元素输入状态”代码表示在输入地址数据和引用数据之间找到的匹配项质量。
- “元素相关性”代码标识目标国家/地区收件人地址所需的地址元素。
- “元素结果状态”代码描述处理过程中对输入数据所做的任何更改。
- “扩展元素结果状态”代码指示地址引用数据包含有关地址元素的其他信息。

每个端口均返回一个包含 20 个字符的代码，其中每个字符引用一个不同的地址数据元素。在元素端口上读取输出代码时，必须了解每个字符所引用的元素。这 20 个字符由 10 对代码组成。每一对中的两个代码都表示一种类型的地址信息。例如，返回代码中的第一个位置表示基本邮政编码信息。

注意：“地址解析代码”端口会基于与“元素状态”端口相同的地址元素返回一个包含 20 个字符的字符串。

下表介绍了每个位置的值所标识的地址元素：

位置	地址元素	说明	地址元素示例
1	邮政编码级别 0	基本邮政编码信息，例如五位数的邮政编码。	五位数的邮政编码 10118
2	邮政编码级别 1	其他邮政编码信息，例如 ZIP +4 编码的最后四位数。	0110（ZIP+4 编码 10118-0110）
3	区域级别 0	主要位置，例如城市或乡镇。	英国伦敦
4	区域级别 1	相关区域，郊区，乡村。	伦敦伊斯灵顿
5	省/自治区/直辖市级别 0	国家/地区的主要区域，例如美国的州名称、加大那的省名称、瑞士的州。	纽约州
6	省/自治区/直辖市级别 1	美国县名。	纽约州 Queens 县
7	街道级别 0	主要街道信息。	South Great George 大街
8	街道级别 1	相关街道信息。	South Great George 大街 George 拱廊
9	编号级别 0	与主要街道相关的建筑物编号或门牌号。	South Great George 大街 460 号
10	编号级别 1	与相关街道相关的建筑物编号或门牌号。	George 拱廊 81 号
11	快递服务级别 0	邮政信箱描述符和编号。	邮政信箱 111 号

位置	地址元素	说明	地址元素示例
12	快递服务级别 1	负责快递的邮局代码。	MAIN STN
13	建筑物级别 0	建筑物名称或编号。 不会标识门牌号。	Alice Tully 大厅
14	建筑物级别 1	其他建筑物名称或编号。	Starr 剧院 Alice Tully 大厅
15	子建筑物级别 0	公寓、房间或楼层名称或编号。	350 第五大街 80 号，80 层
16	子建筑物级别 1	与子建筑物级别 0 信息配对的公寓、房间或楼层信息。	80-18，其中 18 是房间编号，80 是楼层编号
17	组织级别 0	公司名称。	AddressDoctor® GmbH
18	组织级别 1	其他公司信息，例如总公司。	Informatica Corporation
19	国家/地区级别 0	国家/地区名称。	美国
20	国家/地区级别 1	区域。	美国维尔京群岛

如果端口名称包含编号后缀，则级别 0 表示端口编号 1 上的数据，级别 1 表示端口编号 2 至 6 上的数据。

在打印地址中，级别 0 信息可以位于级别 1 信息之前或之后。例如，邮政编码级别 1 位于邮政编码级别 0 之后，而区域级别 1 位于区域级别 0 之前。

地址解析代码输出端口值

“地址解析代码”是一个 20 个字符的字符串，其中，每个字符都代表不同的输入地址元素。字符的值描述地址中对应位的任何非有效地址元素。

下表介绍了地址解析代码端口值：

代码	说明
2	该地址元素是进行投递的必需元素，但不显示在输入地址中。地址引用数据中包含缺少的地址元素。 输出为 2 表示地址缺少地址元素，无法进行投递。
3	地址元素是超出有效地址范围的门牌号或街道编号。例如，地址元素包含在指定街道不存在的门牌号。建议列表模式将返回备用地址。
4	由于输入地址包含多个元素实例，地址验证无法确认或更正地址元素。
5	地址元素在当前地址中不明确，并且地址引用数据包含备选项。地址验证将输入元素复制到输出地址。 例如，地址元素是与地址中的有效区域不匹配的有效邮政编码。
6	地址元素与地址中的其他元素相矛盾。地址验证无法确定正确的地址元素。输出地址复制输入地址。

代码	说明
7	如果不多次更改地址，无法更正地址元素。地址验证可以更正地址，但更改次数表明地址不可靠。
8	数据不符合邮件运营商的验证规则。

元素输入状态输出端口值

“元素输入状态”是一个 20 个字符的字符串，其中，每个字符都代表一个不同的输入地址元素。字符的值代表对相关地址元素执行的处理类型。

在“状态信息”端口组中查找端口。

下表介绍了“元素输入状态”在批量、认证或建议列表模式下可在输出字符串的每个位置上返回的代码：

代码	说明
0	输入地址在当前位置不包含任何数据。
1	引用数据在当前位置不包含任何数据。
2	由于引用数据缺失，无法检查数据。
3	当前位置上的数据不正确。引用数据库表明数值或交付服务值不在引用数据所需的范围内。 在批处理和认证模式下，转换在当前位置上的输入数据未更正的情况下传递为输出。
4	当前位置上的数据与引用数据匹配，但包含错误。
5	当前位置上的数据与引用数据匹配，但转换对数据进行了更正或标准化。
6	当前位置上的数据与引用数据匹配，且不含任何错误。

下表介绍了“元素输入状态”在解析模式下可在输出字符串的每个位置上返回的代码：

代码	说明
0	输入地址在当前位置不包含任何数据。
1	转换已将当前位置上的元素移至输出地址的其他位置。
2	当前位置上的元素与引用数据值匹配，但转换对输出地址中的元素进行了标准化。
3	当前位置上的数据正确无误。

元素相关性输出端口值

“元素相关性”值用于指示邮政投递是否必须提供一个地址元素。在“状态信息”端口组中查找端口。

“元素相关性”值是一个包含 20 个字符的字符串，其中每个字符表示一种不同类型的地址数据。运行地址验证映射后，请查看端口输出以确定每个地址的必要地址元素。使用结果来确认您为地址数据选择了正确的输出端口。如果不为相关地址数据元素选择输出端口，该地址的输出将无效。

下表介绍了“元素相关性”在输出字符串的每个位置所返回的代码：

代码	说明
0	与到该地址的投递不相关。
1	与到该地址的投递相关。 国家邮政运营商不能投递到在输出字符串中的此位置没有数据的地址。

注意: 在批处理模式下，元素相关性值可用于匹配代码值为 Cx 或 Vx 的地址；在交互模式下，元素相关性值可用于匹配代码值为 Cx、Vx、I3 或 I4 的地址。其他评估代码（例如“元素输入状态”、“元素结果状态”、“扩展元素结果状态”和“地址解析代码”）将返回值，而不管匹配代码值如何。

元素结果状态输出端口值

“元素结果状态”是一个 20 个字符的字符串，其中，每个字符都代表一个不同的输入地址元素。每个字符的值都表示验证过程对地址元素所做的任何更新。

在“状态信息”端口组中查找端口。

下表介绍了“元素结果状态”端口值：

代码	说明
0	输出地址在当前位置不包含任何数据。
1	转换在引用数据中找不到当前位置上的数据。转换将输入数据复制到输出数据。
2	当前位置上的数据不会进行检查，但会进行标准化。
3	当前位置上的数据会进行检查，但与引用数据不匹配。引用数据表明数值数据不在有效范围内。转换会将输入数据复制到输出端口。 适用于批处理模式。
4	由于引用数据缺失，转换将输入数据复制到输出数据。
5	当前位置上的数据会进行验证，但由于引用数据中存在多个匹配项，不会进行更改。 适用于批处理模式。
6	数据验证删除了当前位置上的输入值。
7	当前位置上的数据会进行验证，但输入数据包含拼写错误。验证使用引用数据中的值更正了错误。

代码	说明
8	当前位置上的数据会进行验证，且使用引用数据中的值进行更新。 值 8 同时也意味着引用数据库包含有关输入元素的更多数据。例如，如发现街道名称或建筑物名称的完全匹配项，验证可能会添加建筑物编号或子建筑物编号。
9	当前位置上的数据会进行验证，但不会进行更改，并且交付状态不明。例如，DPV 值错误。
C	当前位置上的数据会进行验证和确认，但名称数据已过期。验证操作会更改名称数据。
D	当前位置上的数据会进行验证和确认，但已从外来名称更改为正式名称。
E	当前位置上的数据会进行验证和确认。但是，地址验证对字符大小写或语言进行了标准化。 如果该值与一种备选语言完全匹配，则地址验证可以更改语言。例如，在比利时地址中，地址验证可能会将“Brussels”更改为“Bruxelles”。
F	当前位置上的数据会进行验证和确认，但由于与引用数据完全匹配，不会进行更改。

输出字符串中的位置 19 和 20 与国家/地区数据相关。

下表介绍了验证操作可能会对位置 19 和 20 返回的值：

代码	说明
0	输出地址在当前位置不包含任何数据。
1	地址验证不会识别国家/地区数据。
4	地址验证会根据地址验证器转换中的“默认国家/地区”值标识国家/地区。
5	由于引用数据包含多个匹配项，地址验证无法确定国家/地区。
6	地址验证会根据脚本识别国家/地区。
7	地址验证会根据地址格式识别国家/地区。
8	地址验证会根据主要城镇数据识别国家/地区。
9	地址验证会根据省/市/自治区数据识别国家/地区。
C	地址验证会根据区域数据识别国家/地区。
D	地址验证会根据国家/地区名称识别国家/地区，但名称中包含错误。
E	地址验证会根据地址数据识别国家/地区，例如，根据 ISO 代码或国家/地区名称。
F	地址验证会根据地址验证器转换中设置的“强制国家/地区”值识别国家/地区。

扩展元素结果状态输出端口值

“扩展元素结果状态”是一个由 20 个字符组成的字符串，其中的每个字符都代表一个不同的输入地址元素。端口输出代码补充“元素输入状态”端口和“元素结果状态”端口中的状态数据。端口输出代码还可以指示引用数据中是否存在更多有关地址元素的信息。

在“状态信息”端口组中查找端口。

下表介绍了“扩展元素结果状态”端口值：

代码	说明
1	地址引用数据包含有关地址元素的额外信息。地址验证不需要额外信息。
2	地址验证更新了地址元素以解决数据错误或格式错误。地址验证未验证地址元素。
3	地址验证更新了地址元素以解决数据错误或格式错误。地址验证已验证了地址元素中的数字数据。
4	地址验证将地址元素移动到了其他字段以解决格式错误。
5	地址引用数据包含备用版本的地址元素，如首选区域名称。
6	地址验证未验证地址元素的所有部分。该元素包括地址验证无法验证的数据。
7	地址验证在地址中的错误位置发现了一个有效地址元素。地址验证将地址元素移动到了正确位置。
8	地址验证在错误的数据字段发现了一个有效的地址元素。地址验证将地址元素移动到了正确的字段。
9	地址验证根据邮件运营商验证规则生成了输出元素。
A	地址验证从不同类型的地址中发现符合当前位置的地址元素。地址验证选择了符合目标国家/地区的邮件运营商规则的输出地址元素。
B	地址验证无法确定元素相关性。地址验证为地址所指定的国家/地区返回默认值。
C	建议列表模式。地址验证可以为地址元素返回更多地址建议。要返回更多建议，可更新地址验证器转换的“最大结果计数”属性。
D	地址验证在地址元素中插入数值数据。
E	地址验证无法使用首选语言返回地址元素。地址验证使用默认语言返回元素。
F	地址代码查找模式。输入地址过期。

可邮寄得分输出端口值

“可邮寄得分”值表示对输出地址可交付性的评估。可邮寄得分可用于对地址的可交付性进行一般性的评判。在“状态信息”端口组中查找端口。

地址验证器转换在计算可邮寄得分时会考虑诸多因素。该转换的计算依据主要是地址的“匹配代码”值和“元素结果状态”值。影响可邮寄得分的其他因素包括地址值的邮政相关性和国家/地区引用数据的粒度。

“可邮寄得分”端口值对地址的可交付性进行评估。该得分并不能精确或最终判定地址的可交付性。

下表介绍了“可邮寄得分”输出代码：

值	摘要	说明
5	完全有信心	表示地址验证已检查并确认输入地址中的所有相关元素。
4	几乎肯定	表示下列任一情况： - 因缺少引用数据而无法检查一个或多个相关地址元素。其他地址元素已得到确认。 - 地址验证对一个或多个相关元素进行了极高可信度的更正。当地址验证在输入地址和引用数据之间找到单项匹配且差异度相当低时，便属于这种情况。
3	应该良好	表示地址验证已更正输入地址中的一个或多个相关元素。地址验证在输入地址和引用数据之间找到单项匹配，且差异度可接受。
2	机会较高	表示地址验证由于下列原因之一而无法更正或确认地址： - 地址验证在引用数据中找不到足够可信的候选匹配项。 - 地址验证找到了可信度类似的多个候选匹配项。 邮件运营商可能可以投递到该地址。
1	有风险	表示地址验证只能为输入地址找到部分引用数据匹配项。
0	不可交付	表示地址验证在引用数据中找不到地址的匹配项。输入地址缺少过多的元素，或者地址验证无法确认地址中的大部分元素。

匹配代码输出端口值

“匹配代码”值汇总了输入地址与引用数据的对比结果。该代码还汇总了转换对地址进行的任何更正。在“状态信息”端口组中查找端口。

下表介绍了“匹配代码”输出端口值：

代码	说明
A1	地址代码查找过程为输入代码发现了不完整地址或完整地址。
A0	地址代码查找过程没有为输入代码发现地址。
C4	已更正。已检查所有与邮政相关的元素。
C3	已更正。无法检查部分元素。
C2	已更正，但是由于缺少引用数据，送达状态未知。
C1	已更正，但是由于用户标准化产生了错误，送达状态未知。
I4	无法完全更正数据，但是存在一个与引用数据中的地址匹配的匹配项。
I3	无法完全更正数据，但是存在多个与引用数据中的地址匹配的匹配项。
I2	无法更正数据。批处理模式返回部分建议的地址。
I1	无法更正数据。批处理模式无法建议地址。

代码	说明
N7	验证错误。由于单行验证未解锁，因此未进行验证。
N6	验证错误。由于目标国家/地区不支持单行验证，因此未进行验证。
N5	验证错误。由于引用数据库已过期，因此未进行验证。
N4	验证错误。由于引用数据已损坏或格式化错误，因此未进行验证。
N3	验证错误。由于国家/地区数据无法解锁，因此未进行验证。
N2	验证错误。由于所需的引用数据库不可用，因此未进行验证。
N1	验证错误。由于国家/地区无法识别或不受支持，因此未进行验证。
Q3	建议列表模式。地址验证可以从地址引用数据中检索到一个或多个与输入地址对应的完整地址。
Q2	建议列表模式。地址验证可以合并输入地址的元素和地址引用数据的元素来生成完整地址。
Q1	建议列表模式。地址验证无法建议完整地址。要生成完整地址建议，请向输入地址中添加数据。
Q0	建议列表模式。用于生成建议的输入数据不足。
RB	通过缩写识别了国家/地区。识别 ISO 双字符和 ISO 三个字符的国家/地区代码。也可识别常见缩写，如“GER”代表德国。
RA	通过强制国家/地区属性识别了国家/地区。
R9	通过默认国家/地区属性识别了国家/地区。
R8	通过国家/地区名称识别了国家/地区。
R7	通过国家/地区名称识别了国家/地区，但是验证进程在国家/地区数据中发现了错误。
R6	通过领土数据识别了国家/地区。
R5	通过省/市/自治区数据识别了国家/地区。
R4	通过主要城镇数据识别了国家/地区。
R3	通过地址格式识别了国家/地区。
R2	通过语言字母识别了国家/地区。
R1	由于存在多个匹配，未识别国家/地区。
R0	未识别国家/地区。
S4	解析模式。地址已完全解析。
S3	解析模式。解析地址后有多个结果。

代码	说明
S1	解析模式。由于输入格式不匹配，出现解析错误。
V4	已验证。输入数据正确。地址验证已检查所有与邮政相关的元素，且输入完全匹配。
V3	已验证。输入数据正确，但部分或所有元素已标准化，或者输入中包含的名称或外来语已过期。
V2	已验证。输入数据正确，但是由于引用数据不完整，无法验证某些元素。
V1	已验证。输入数据正确，但是用户标准化对可送达性造成了负面影响。例如，邮政编码长度过短。

地理编码状态输出端口值

下表介绍了“地理编码状态”输出端口值。在“地理编码”端口组中查找此端口。

如果已为输入地址国家/地区安装了地理编码引用数据，请选择此端口。

值	说明
EGC0	由于地址的地理编码不可用，无法向输入地址附加地理编码。
EGC1-3	保留以供将来使用。
EGC4	地理编码在一定程度上精确到邮政编码级别。
EGC5	地理编码精确到邮政编码级别。
EGC6	地理编码精确到区域级别。
EGC7	地理编码精确到街道级别。
EGC8	地理编码精确到门牌号级别。地理编码可估测门牌号位置，且包含朝向邮箱所在街道一侧的偏移。
EGC9	地理编码精确到到达点或屋顶。
EGCA	地理编码精确到地片的中心。
EGCC	地理编码数据库损坏。
EGCN	找不到地理编码数据库。
EGCU	地理编码数据库未解除锁定。

注意: Informatica 不再为地片中心和屋顶级地理编码发布引用数据。

地址验证器转换常规设置

配置常规设置以设置地址验证所需的参数。

在**常规设置**视图中可以配置以下属性：

默认国家/地区

如果转换无法标识输入地址中的国家/地区目标，则指定其使用的地址引用数据集。如果数据中包含国家/地区信息，则选择无。

还可以将默认国家/地区设置为转换上的高级属性。

强制国家/地区

可选属性。使用默认国家/地区名称或缩写替换输入地址中的国家/地区名称或缩写。如果输入地址未指定国家/地区，转换会将默认国家/地区数据附加到地址。

行分隔符

指定单行地址内分隔数据字段的分隔符号。

还可以将行分隔符设置为转换上的高级属性。

大小写样式

设置输出数据的字符大小写样式。选择混合选项以遵循首字母大写的地址引用数据标准。选择保留选项以按地址引用数据使用的大小写样式写入地址。

还可以将大小写样式设置为转换上的高级属性。

模式

决定转换执行的验证类型。

选择以下选项之一：

模式类型	说明
地址代码查找	当提供地址代码作为输入时，从引用数据返回部分地址或完整地址。许多国家/地区支持使用地址代码表示地址的区域、街道、建筑物或唯一邮箱。
批处理	针对数据集中的记录执行地址验证。批量验证专注于地址完整度和送达性。批处理模式不会对质量不好的地址返回建议。批处理模式为默认模式。
已认证	针对数据集中的记录进行地址验证，以了解它们是否符合指定国家/地区的认证标准。认证标准要求每个地址标识唯一的邮箱。可以对澳大利亚、法国、新西兰、英国和美国的地址进行认证地址验证。
国家/地区识别	确定通信地址的目标国家/地区。在国家/地区识别模式下，转换不执行地址验证。
交互	完成不完整的有效地址。当不完整的输入地址与引用数据中的多个地址匹配时，转换将返回所有有效地址，最多为“最大结果计数”指定的上限。
解析	将数据解析到地址字段。在解析模式下，转换不执行地址验证。
建议列表	当输入地址包含片段信息时，从引用数据返回有效地址列表。当地址片段与引用数据中的多个地址匹配时，转换将返回所有有效地址，最多为“最大结果计数”指定的上限。

还可以将模式设置为转换上的高级属性。

首选项窗口中的地址验证属性

您可以查看地址验证引擎的属性和引擎在 Developer tool 中读取的地址引用数据文件。Developer tool 公开了数据集成服务用于运行地址验证映射的引擎的属性。Developer tool 列出了控制地址验证操作的内容管理服务的属性。

使用 Developer tool 中的**首选项**窗口查看属性。在**首选项**窗口中选择**内容状态**选项，以标识当前数据集成服务所使用的内容管理服务。要查看属性，请选择本地内容管理服务。

您可以查看以下属性：

地址验证数据

地址验证数据属性列出了当前内容管理服务可以向数据集成服务提供的引用数据类型。这些属性还指示引用数据所适用的国家/地区。

地址验证引擎

地址验证引擎属性包括证书组件最近更新的当前引擎版本以及数据预加载方法。

地址验证许可证

地址验证许可证属性包括当前内容管理服务可以向数据集成服务提供的引用数据的许可证信息。

地址验证数据属性

地址验证数据属性列出了当前内容管理服务可以向数据集成服务提供的引用数据类型。这些属性还包括引用数据所适用的国家/地区。

下表描述了在**内容状态**视图中选择内容管理服务时显示的数据属性：

属性	说明
国家/地区 ISO	地址引用数据文件所适用的国家/地区。该属性显示国家/地区的 ISO 三字符代码。
过期日期	当前文件的过期日期。Informatica 在过期日期发布一个较新的文件。可以在过期日期后使用当前的地址引用数据文件，但是该文件中的数据可能不再准确。
国家/地区类型	可以对数据执行的地址验证类型。 在**常规设置**选项卡上的**模式**选项中选择处理类型。如果选定模式与域上的地址数据文件不对应，则地址验证映射将失败。
解锁过期日期	许可证的过期日期。在解锁过期日期后，不能使用本文件的任何版本。 “地址验证许可证属性”视图上的“解锁过期日期”属性和“过期日期”属性代表相同的信息。
解锁开始日期	许可证在“国家/地区类型”属性标识的模式下和“国家/地区 ISO”属性标识的国家/地区中生效的日期。在解锁开始日期前，不能使用本文件的任何版本。

地址验证许可证属性

地址验证许可证属性包括当前内容管理服务可以向数据集成服务提供的引用数据的许可证信息。

下表描述了在内容状态视图中选择内容管理服务时显示的许可证属性：

属性	说明
解锁代码	该许可证代码为“代码类型”属性标识的模式解锁引用数据。Developer tool 显示代码的前四个字符，并屏蔽其他字符。
代码类型	可以对许可证指定的数据执行的地址验证模式。Informatica 为每个模式颁发一个许可证代码。许可证代码可以适用于一个或多个国家/地区。 在**常规设置**选项卡上的**模式**选项中选择处理类型。如果选定模式与域上的地址数据文件不对应，则地址验证映射将失败。
国家/地区列表	解锁代码为其解锁引用数据的国家/地区。 “国家/地区列表”属性包含每个国家/地区的一个或多个 ISO 三字符代码。
状态	许可证代码的状态。当许可证文件有效时，该属性返回“确定”。
过期日期	许可证的过期日期。 “地址验证数据属性”视图中的“过期日期”属性和“解锁过期日期”属性代表相同的信息。

地址验证引擎属性

地址验证引擎属性包括证书组件最近更新的当前引擎版本以及数据预加载方法。

下表描述了在内容状态视图中选择内容管理服务时显示的引擎属性：

属性	值
引擎版本	数据集成服务运行的地址验证引擎的版本。
CASS 版本	Informatica 最近更新了 CASS 认证组件的地址验证引擎的版本。使用该属性可在 CASS 认证报告中标识引擎版本。 该属性还包括该引擎支持的 CASS 认证周期。例如，引擎可能支持认证周期 N。
AMAS 版本	Informatica 最近更新了 AMAS 认证组件的地址验证引擎的版本。使用该属性可在 AMAS 认证报告中标识引擎版本。
SendRight 版本	Informatica 最近更新了 SendRight 认证组件的地址验证引擎的版本。使用该属性可在 SendRight 认证报告中标识引擎版本。
SERP 版本	Informatica 最近更新了 SERP 认证组件的地址验证引擎的版本。使用该属性可在 SERP 认证报告中标识引擎版本。
SNA 版本	Informatica 最近更新了 SNA 认证组件的地址验证引擎的版本。使用该属性可在 SNA 认证报告中标识引擎版本。
正在预加载方法	数据集成服务用来将引用数据库预加载到内存中的方法。内容管理服务属性指定数据集成服务预加载引用数据的国家/地区。可能的值为 MAP 和 LOAD。默认值为 MAP。 MAP 方法和 LOAD 方法二者均分配内存块并于之后将引用数据读入此块。但是，MAP 方法可在多个进程之间共享引用数据。

属性	值
缓存大小	数据集成服务针对服务未预加载的引用数据使用的数据缓存的大小。可能的值为 NONE、SMALL 和 LARGE。默认值为 LARGE。
内存使用量上限	地址验证引擎可以分配的内存兆字节数。默认值为 4096。
地址对象计数上限	数据集成服务可以同时运行的最大地址验证实例数。默认值为 3。
线程计数上限	地址验证可使用的最大线程数。默认值为 2。
最大结果计数	在建议列表模式下运行映射时，地址验证可以返回的最大地址数。默认值为 20。该属性的上限为 100。
当前日期	当前日期。Developer tool 返回适用于当前日期的属性值。
编写 XML BOM	指示数据集成服务是否在 GetConfig.xml 文件中写入字节顺序标记。可能的值为 ALWAYS、IF_NECESSARY 和 NEVER。默认值为 IF_NECESSARY。
XML 编码	标识地址验证引擎用于读取和写入数据的 XML 编码。

地址验证高级属性

配置高级属性以确定数据集成服务如何为地址验证器转换处理数据。

别名区域

决定地址验证是否将有效的区域别名替换为官方区域名称。

区域别名是备用区域名称，USPS 将其识别为可投递地址中的元素。配置地址验证器转换时可以使用该属性验证认证模式下的美国地址记录。

下表介绍了别名区域选项：

选项	说明
关闭	禁用“别名区域”属性。
官方	将任何备选区域名称或区域别名替换为官方区域名称。默认选项。
保留	保留有效的备选区域名称或区域别名。如果输入区域名称无效，地址验证将使用官方名称替换该名称。

别名街道

确定地址验证是否会使用官方街道名称替换街道别名。

街道别名是备用街道名称，USPS 将其识别为可交付地址中的元素。配置地址验证器转换时可以使用该属性验证认证模式下的美国地址记录。

下表介绍了别名街道选项：

选项	描述
关闭	不应用属性。
官方	用官方街道名称替换所有备用街道名称或街道别名。默认选项。
保留	保留有效的备用街道名称或街道别名。如果输入街道名无效，地址验证将使用官方名称替换该名称。

大小写样式

指定转换应用到输出地址数据的字符大小写样式。

下表介绍了“大小写样式”选项：

选项	说明
分配参数	使用定义的参数设置大小写样式。
小写	以小写字母写入输出地址。
混合	尽可能采用目标国家/地区当前使用的大小写样式。
保留	应用地址引用数据中使用的大小写样式。 默认选项。
无更改	不对地址应用大小写样式。 **注意: 无更改**选项不保证输入地址与输入地址的大小写匹配。如果地址验证将某个地址元素替换为引用数据中的元素，该元素将沿用引用数据所使用的大小写。
大写	以大写字母写入输出地址。

还可以在**常规设置**选项卡上配置大小写样式。

参数的用法

可以使用以下参数之一指定大小写样式：

- LOWER. 以小写字母写入输出地址。
- MIXED. 尽可能采用目标国家/地区当前使用的大小写样式。
- NATIVE. 应用地址引用数据中使用的大小写样式。 默认选项。 与**保留**选项匹配。
- NOCHANGE. 不对地址应用大小写样式。
- UPPER. 以大写字母写入输出地址。

以大写字母输入参数值。

来源的国家/地区

标识发送地址记录的国家/地区。

从列表中选择一个国家/地区。 默认情况下，该属性为空。

国家/地区类型

确定“完整地址”或“格式化的地址行”端口输出数据中国家/地区名称或缩写的格式。转换将以选定国家/地区的标准格式写入国家/地区名称或缩写。

下表介绍了“国家/地区类型”选项：

选项	Country
ISO 2	ISO 双字符国家/地区代码
ISO 3	ISO 三个字符的国家/地区代码
ISO #	ISO 三位数国家/地区代码
缩写	（保留以供将来使用）
CN	加拿大
DA	（保留以供将来使用）
DE	德国
EN	英国（默认）
ES	西班牙
FI	芬兰
FR	法国
GR	希腊
IT	意大利
JP	日本
HU	匈牙利
KR	大韩民国
NL	荷兰
PL	波兰
PT	葡萄牙
RU	俄罗斯
SA	沙特阿拉伯
SE	瑞典

默认国家/地区

指定当地址记录未标识目标国家/地区时转换使用的地址引用数据集。

从列表中选择一个国家/地区。 如果地址记录包含国家/地区信息，则使用默认选项。默认值为“无”。

还可以在**常规设置**选项卡上配置默认国家/地区。

参数的用法

可以使用参数以指定默认国家/地区。创建参数时，请为国家/地区输入 ISO 3166-1 alpha-3 代码作为参数值。输入参数值时，请使用大写字符。例如，如果所有地址记录包括国家/地区信息，请输入 NONE。

双地址优先级

确定要验证的地址的类型。 如果输入地址记录包含多个有效地址数据类型，则设置该属性。

例如，当地址记录包含邮政信箱元素和街道元素时使用该属性。 地址验证读取包含所指定的地址数据类型的数据元素。 地址验证会忽略地址中的任何不兼容数据。

下表介绍了有关双地址优先级属性的选项：

选项	说明
投递服务	验证地址中的投递服务数据元素，如邮政信箱元素。
邮政管理	验证当地邮政运营商所需的地址元素。 默认选项。
街道	验证地址中的街道数据元素，如建筑物编号元素和街道名称元素。

元素缩写

确定转换是否返回地址元素的缩写形式。如果地址引用数据包含缩写，可将转换设置为返回缩写形式。

例如，美国邮政服务 (USPS) 包含很多街道和区域名称的简写形式和详细形式。HUNTSVILLE BROWNSFERRY RD 的简写形式为 HSV BROWNS FRY RD。当街道或区域值超出 USPS 规定的最大字段长度时，可以选择元素缩写属性。

默认情况下，将清除此选项。将该属性设置为“开启”可返回缩写的地址值。如果在分支模式中使用转换，该属性将返回缩写形式的区域名称和区域代码。如果在认证模式中使用转换，该属性将返回缩写形式的街道名称、区域名称以及区域代码。

执行实例

指定数据集成服务在运行时试图为当前转换创建的线程数。如果您替代包含转换的映射上的“最大并行数”运行时属性，则数据集成服务则会考虑“执行实例”值。“执行实例”默认值为 1。

数据集成服务会考虑多个因素来确定分配给转换的线程数。首要因素为“执行实例”值以及映射和域中关联应用程序服务上的值。

在计算用于转换的线程数时，数据集成服务读取以下值：

- 数据集成服务上的*最大并行数*值。默认值为 1。
- 您在映射级别设置的任何*最大并行数*值。默认为“自动”。
- 转换上的*执行实例*值。默认值为 1。

如果您替代映射级别的“最大并行数”值，则数据集成服务会尝试使用属性间的最低值来确定线程数。

如果您使用映射级别的默认“最大并行数”值，则数据集成服务会忽略“执行实例”值。

计算要创建的线程数时，数据集成服务还会考虑内容管理服务上的*地址对象计数上限*属性。*地址对象计数上限*属性决定了可以在映射中并发运行的最大地址验证实例数。*地址对象计数上限*属性值必须大于或等于数据集成服务上的*最大并行数*值。

执行实例属性的规则和准则

设置执行实例数量时，请注意以下规则和准则：

- 多个用户可能会在数据集成服务上运行并发映射。要计算正确的线程数，请将服务可以访问的中央处理单元数量除以并发的映射数量。
- 在 PowerCenter 中，*AD50.cfg* 配置文件指定了可以在映射中并发运行的最大地址验证实例数。
- 如果使用默认的“执行实例”值和默认的“最大并行数”值，则转换操作不可分区。
- 如果设置的“执行实例”值大于 1，地址验证器转换会从被动转换更改为主动转换。

灵活范围扩展

设置“要扩展的范围”属性时，对地址验证器转换返回的地址数强制使用实际限制。 将转换配置为在建议列表模式下运行时，可以设置“要扩展的范围”属性和“灵活范围扩展”属性。

“要扩展的范围”属性确定当输入地址不包含门牌号数据时，转换返回地址建议的方式。 如果输入地址不包含上下文数据（如完整邮政编码），“要扩展的范围”属性会生成大量非常相似的地址。 “灵活范围扩展”属性会限制“要扩展的范围”属性为一个单独地址所生成的地址数量。 如果将“要扩展的范围”属性设置为“全部”，则将“灵活范围扩展”属性设置为“开启”。

下表介绍了有关“灵活范围扩展”属性的选项：

选项	说明
打开	地址验证限制“要扩展的范围”属性添加到建议列表中的地址的数量。 默认选项。
关闭	地址验证不限制“要扩展的范围”属性添加到建议列表中的地址的数量。

注意: 地址验证器转换以不同的方式为其返回到建议列表中的每个地址应用“灵活范围扩展”属性。 该转换对列表中的扩展地址的数量强制使用固定限制。 该转换在计算要包含到列表中的扩展地址的数量时，还会考虑“最大结果计数”属性设置。

地理编码数据类型

确定地址验证器转换如何计算地址的地理编码数据。地理编码是纬度坐标和经度坐标。

转换返回的地理编码结果取决于安装的地理编码引用数据。有关地理编码引用数据的信息，请联系 Informatica。

选择以下地理编码选项之一：

到达点

返回建筑物或地片入口的经纬度坐标。默认选项。

您可以为以下国家/地区的地址选择到达点选项：

澳大利亚、奥地利、加拿大、克罗地亚、丹麦、爱沙尼亚、芬兰、法国、德国、匈牙利、意大利、拉脱维亚、列支敦士登、立陶宛、卢森堡、墨西哥、摩纳哥、荷兰、挪威、波兰、斯洛伐克、斯洛文尼亚、瑞典、瑞士和美国。

如果指定了到达点地理编码，但地址验证器转换无法为地址返回地理编码，则转换会返回内插式地理编码。

标准

返回建筑物或地片入口的估计纬度和经度坐标。估计的地理编码也称为内插式地理编码。

地址验证器转换使用引用数据中的最近可用地理编码来估计地址的地理编码。

注意: Informatica 不再为地片中心或屋顶级地理编码发布引用数据。

参数的用法

可以利用参数指定地理编码类型。输入 ARRIVAL_POINT 或 NONE。要返回标准地理编码，请输入 NONE。

以大写字母输入参数值。

全局最大字段长度

确定地址中任意行上的最大字符数。 如果地址验证器转换写入其中包含的字符多于您指定的字符的输出地址行，则转换会缩写行上的地址元素。

使用该属性控制地址中的行长度。 例如，SNA 标准要求地址在任意行上包含的字符数不多于 38。 如果您将地址生成为 SNA 标准，请将“全局最大字段长度”设置为 38。

默认值为 1024。

参数的用法

可以使用参数指定最大地址数。 要设置参数值，请输入 0 到 1024 之间的整数。

全局首选描述符

确定地址验证器转换写入输出数据的建筑物描述符、子建筑物描述符以及街道描述符的格式。当目标国家/地区的地址引用数据包含一个或多个数据元素的一系列描述符时，请选择一个描述符。

下表介绍了有关属性的选项：

选项	说明
数据库	返回引用数据库为地址中的元素指定的描述符。如果数据库未指定地址描述符，转换会将输入值复制到输出地址。 数据库为默认值。
长	返回描述符的完整形式，如 *Street*。
保留输入	将描述符从输入地址复制到输出地址。 如果输入描述符不是有效的描述符版本，转换会返回引用数据库中的有效对等描述符。
短	返回描述符的缩写形式，如 *St*。

输入格式类型

介绍了包含未填充输入数据中所含信息的最常见类型。将输入数据连接到“完整地址”或“格式化的地址行”端口时，请使用“输入格式类型”属性。选择最能说明映射源数据中信息的选项。

选择以下选项之一：

- 全部
- 地址

- 组织
- 联系人
- 组织/联系人
  地址包括组织信息和联系人信息。
- 组织/部门
  地址包括组织信息和部门信息。

默认为“全部”。

带国家/地区的输入格式

指定输入是否包含国家/地区数据。如果将输入数据连接到“完整地址”或“格式化的地址行”输入端口，并且该数据包含国家/地区信息时，请选择该属性。

默认情况下，将清除此选项。

行分隔符

指定用于指示格式化地址中的换行符的分隔符号。

选择以下选项之一：

- 分配参数以标识行分隔符
- 回车符 (CR)
- 逗号
- 换行符(LF)
- 无
- 分号
- 制表符
- Windows 换行符(CRLF)

默认值为分号。

还可以在**常规设置**选项卡上配置行分隔符。

参数的用法

可以使用参数指定行分隔符。 参数值区分大小写。 以大写字符输入参数值。

输入以下值之一：

- CR
- CRLF
- COMMA
- LF
- PIPE
- SEMICOLON
- SPACE
- TAB

匹配备选项

确定地址验证是否识别输入地址中的备用地名，例如别名或历史名称。该属性适用于街道、区域和省/自治区/直辖市数据。

注意：“匹配备选项”属性不会保留已验证地址中的备用名称。

下表介绍了“匹配备选项”选项：

选项	说明
全部	识别所有已知的备用街道名和地名。默认选项。
仅存档	仅识别历史名称。例如，地址验证将“Constantinople”验证为“Istanbul”的历史版本。
无	不识别备用街道名或地名。
仅别名	仅识别别名和外来语地名。例如，地址验证将“Londres”识别为“London”的外来语地名。

匹配扩展存档

确定地址验证是否返回过期日本地址的唯一收件人地点代码。

日本的地址引用数据文件在相应邮箱的当前地址旁边包括过期或已停用的地址。选择“匹配扩展存档”属性时，地址验证将返回每个地址当前版本的收件人地点代码。地址验证还会向“扩展元素结果状态”端口中写入值，以指示输入地址过期。

要从地址引用数据中检索当前地址，请输入地址代码作为输入元素。

下表介绍了“匹配扩展存档”选项：

选项	说明
关闭	不应用属性。
打开	返回过期日本地址当前版本的地址代码。

“匹配扩展存档”属性对日本使用补充数据和地址代码查找数据。要在地址验证中应用此属性，请将转换配置为在地址代码查找模式下运行。

匹配范围

确定地址验证期间转换与地址引用数据匹配的数据量。

下表介绍了“匹配范围”选项：

选项	说明
全部	验证所有选定端口。默认选项。
收件人地点	除“街道”选项验证的数据外，还将验证建筑物和子建筑物地址数据。

选项	说明
区域	验证省/自治区/直辖市、区域和邮政编码数据。
街道	除“区域”选项验证的数据外，还将验证街道地址数据。

最大结果计数

确定地址验证在建议列表模式下可以返回的最大地址数。

可以将数量上限设置为介于 1 到 100 之间。 默认值为 20。

注意: 建议列表模式将对地址引用数据执行地址检查，并返回可能与输入地址匹配的地址的列表。 以建议列表模式验证地址时，地址验证将首先返回最佳匹配项。

参数的用法

可以使用参数指定最大地址数。 要设置参数值，请输入 0 到 100 之间的整数。

模式

确定转换所执行的地址分析类型。还可以在转换的**常规设置**选项卡上配置模式。

相关主题:

- “ 地址验证器转换常规设置” 页面上 91

优化级别

确定转换如何匹配输入地址数据和地址引用数据。该属性定义转换在更新地址记录之前必须在输入数据与引用数据之间找到的匹配类型。

下表介绍了“优化级别”选项:

选项	说明
精优化	该转换在执行验证前解析街道信息中的建筑物编号和门牌号。 此外该转换将根据输入端口结构严格验证输入地址元素。 精优化选项将执行最快速的地址验证，但是返回的结果可能没有其他选项的结果准确。
标准	转换先解析输入数据中的多种地址信息类型，然后再执行验证。 选择标准选项后，如果转换可以将多个输入值与引用数据匹配，则转换将更新地址。 默认为标准。
泛优化	转换使用标准解析设置，并在输入数据中执行其他解析操作。 选择泛优化选项后，如果转换可以将至少一个输入值与引用数据匹配，则转换将更新地址。 泛优化选项会增加映射运行时间。

参数的用法

可以利用参数指定优化级别。 输入 NARROW、STANDARD 或 WIDE。 以大写字母输入参数值。

输出格式类型

描述转换在“完整地址”或“格式化的地址行”输出端口上写入的最常见信息类型。 选择最能说明输出端口上所需数据的选项。

选择以下选项之一：

- 全部
- 地址
- 组织
- 联系人
- 组织/联系人
  地址包括组织信息和联系人信息。
- 组织/部门
  地址包括组织信息和部门信息。

默认为“全部”。

带国家/地区的输出格式

确定转换将国家/地区标识数据写入“完整地址”输入端口还是“格式化地址行”输出端口。

默认情况下，将清除此选项。

首选语言

决定当引用数据集包含多种语言的数据时，地址验证器转换返回的地址元素所使用的语言。您可以为比利时、加拿大、中国、芬兰、中国香港、爱尔兰、以色列、中国澳门、瑞士和中国台湾的地址设置首选语言。

地址验证器转换可以采用以下语言返回地址数据：

- 地址引用数据中的地址的默认语言。默认语言是每个地址所属的地区主要使用的语言。
- 地址引用数据对某个地址支持的任何其他语言。例如，比利时引用数据包含采用佛兰德语、法语和德语的地址元素。

地址引用数据可能包含单个地址元素的数据，也可能包含采用多种语言的完整地址的数据。例如，地址验证可以采用英语返回爱尔兰的所有地址元素，还可以采用爱尔兰语返回街道、区域和省份信息。此外，引用数据可能为属于某个国家/地区的不同区域的地址指定不同的默认语言。例如，在瑞士引用数据中，默认语言因地区而异，其中包括法语、德语和意大利语。

下表总结了“首选语言”属性可供选择的选项：

选项	说明
数据库	使用地址引用数据指定的语言返回每个地址。地址引用数据集可能会为一个国家/地区中不同区域的地址指定不同的语言。 “数据库”为默认选项。
备选 1、备选 2、备选 3	使用备选语言从引用数据返回地址元素。备选语言取决于地址所属的国家/地区。

选项	说明
英语	当引用数据包含英语数据时，采用英语返回地址元素。使用地址所属区域的默认语言返回其他地址元素。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

注意: 地址引用数据集可能包含某些使用非默认语言的地址元素，但其他项目并非如此。如果转换找不到采用属性指定语言的元素，将使用默认语言返回该元素。

设置首选语言选项时，请验证“首选脚本”属性指定的字符集是否与所需的输出地址数据兼容。

比利时地址的多语言支持

下表介绍了可为比利时地址指定的语言：

选项	说明
数据库	默认值。使用地址所属地区的主要语言返回地址。该语言可能是弗拉芒语、法语或德语。
英语	如果地址引用数据包含英语信息，则使用英语返回省份、区域和街道信息。 使用地址所属地区的主要语言返回其他地址元素。
备选 1	使用佛兰芒语返回省份、区域和街道信息。
备选 2	使用法语返回省份、区域和街道信息。
备选 3	使用德语返回省份、区域和街道信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

针对加拿大地址的多语言支持

下表介绍了您可以为加拿大地址指定的语言：

选项	说明
数据库	默认值。使用英语返回除魁北克以外的所有省份的地址。 使用法语返回魁北克地址。
英语	使用英语返回所有地址。
备选 1	使用英语返回所有地址。

选项	说明
备选 2	使用法语返回魁北克地址。 在除魁北克以外的省份，转换将使用法语返回街道描述符、方向信息和省份名称，并使用英语返回其他地址元素。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

中国地址的多语言支持

下表介绍了您可以为中国地址指定的语言：

选项	说明
数据库	默认值。使用中文返回所有地址信息。
英语	返回街道描述符和街道方向值的英语版本。使用中文返回所有其他地址信息。 英语地址元素将忽略转译元素（例如“shi”）。
备选 1	使用数据库语言返回所有地址信息。
备选 2	使用数据库语言返回所有地址信息。
备选 3	使用数据库语言返回所有地址信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

选择首选语言时，请考虑以下规则和准则：

- 若要使用中文返回地址，请选择“数据库”、“备选 1”、“备选 2”或“备选 3”。

  若要使用中文字符集返回地址，请将“首选脚本”属性设置为“数据库”。
- 要使用英语返回街道描述符和街道方向信息，请选择“英语”。

  要使用拉丁字符集或 ASCII 字符集返回地址，请将“首选脚本”属性设置为值 LATIN 或 ASCII。
- 如果选择 LATIN 或 ASCII 值作为首选脚本并选择“数据库”作为首选语言，则地址验证将使用拼音返回地址数据。

芬兰地址的多语言支持

下表介绍了您可以为芬兰地址指定的语言：

选项	说明
数据库	默认值。使用芬兰语返回所有地址信息。
备选 1	使用数据库语言返回所有地址信息。

选项	说明
备选 2	使用瑞典语返回街道、区域和省份信息。使用芬兰语返回所有其他信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

香港地址的多语言支持

下表介绍了您可以为香港地址指定的语言：

选项	说明
数据库	默认值。使用中文返回所有地址信息。
英语	使用英语返回所有地址信息。
备选 1	使用数据库语言返回所有地址信息。
备选 2	使用英语返回所有地址信息。
备选 3	使用数据库语言返回所有地址信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

为香港选择首选语言时，请考虑以下规则和准则：

- 若要使用中文字符集返回地址，请将“首选脚本”属性设置为“数据库”。
- 要使用拉丁字符集或 ASCII 字符集返回地址，请将“首选脚本”属性设置为值 LATIN 或 ASCII。
- 输入数据的语言会影响对香港地址进行的“保留输入”选项的操作。当输入数据使用 7 位 ASCII 字符并包含英语描述符时，地址验证会将输入语言标识为英语。

爱尔兰地址的多语言支持

下表介绍了您可以为爱尔兰地址指定的语言：

选项	说明
数据库	默认值。使用英语返回所有地址信息。
英语	使用英语返回所有地址信息。
备选 1	使用英语返回所有地址信息。

选项	说明
备选 2	使用爱尔兰语返回街道、区域和郡信息。使用英语返回所有其他地址信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

以色列地址的多语言支持

下表介绍了您可以为以色列地址指定的语言：

选项	说明
数据库	默认值。使用希伯来语返回所有地址信息。
英语	使用英语返回所有地址信息。
备选 1	使用希伯来语返回所有地址信息。
备选 2	使用英语返回所有地址信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

选择首选语言时，请考虑以下规则和准则：

- 若要使用希伯来语字符集返回地址，请将“首选脚本”属性设置为“数据库”。
- 要使用拉丁字符集或 ASCII 字符集返回地址，请将“首选脚本”属性设置为值 LATIN 或 ASCII。
- 如果选择拉丁字符集作为首选脚本并选择希伯来语作为首选语言，地址验证则会将希伯来语地址音译为拉丁字符。若要以拉丁字符集获得最佳结果，请选择英语作为首选语言。

澳门地址的多语言支持

下表介绍了您可以为澳门地址指定的语言：

选项	说明
数据库	默认值。使用中文返回所有地址信息。
备选 1	使用数据库语言返回所有地址信息。
备选 2	使用葡萄牙语返回所有地址信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

- 若要使用中文字符集返回地址，请将“首选脚本”属性设置为“数据库”。

- 要使用拉丁字符集或 ASCII 字符集返回地址，请将“首选脚本”属性设置为值 LATIN 或 ASCII。
- 输入数据的语言会影响对澳门地址进行的“保留输入”选项的操作。当输入数据使用 7 位 ASCII 字符并包含葡萄牙语描述符时，地址验证会将输入语言标识为葡萄牙语。

瑞士地址的多语言支持

下表介绍了您可以为瑞士地址指定的语言：

选项	说明
数据库	默认值。使用地址所属地区的主要语言返回地址。 例如，地址验证使用德语返回一个苏黎世地址并使用法语返回一个日内瓦地址。
英语	如果引用地址数据库包含英语信息，则使用英语返回区域和州信息。 使用地址所属地区的主要语言返回其他地址元素。 地址验证使用英语为某些区域（如日内瓦和苏黎世）返回区域信息。
备选 1	使用德语返回州和区域信息。
备选 2	使用法语返回州和区域信息。
备选 3	使用意大利语返回州和区域信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

注意: 地址验证还将以您配置的备选语言返回比尔市地址的街道信息。

台湾地址的多语言支持

下表介绍了您可以为台湾地址指定的语言：

选项	说明
数据库	默认值。使用中文返回所有地址信息。
英语	使用英语返回所有地址信息。
保留输入	使用输入语言返回地址信息。如果引用数据包含使用输入语言的地址信息，地址验证则会保留该语言。 如果地址验证在输入地址中检测到多种受支持的语言，则使用数据库语言返回地址。如果地址验证无法使用输入语言返回某个元素，则使用数据库语言返回该元素。

选择首选语言时，请考虑以下规则和准则：

- 要使用中文字符集返回地址，请将“首选脚本”参数设置为“数据库”。
- 要使用拉丁字符集或 ASCII 字符集返回地址，请将“首选脚本”参数设置为值 LATIN 或 ASCII。
- 输入数据的语言会影响对台湾地址进行的“保留输入”选项的操作。当输入数据使用 7 位 ASCII 字符并包含英语描述符时，地址验证会将输入语言标识为英语。

首选脚本

确定地址验证器转换用于输出数据的字符集。

下表介绍了有关属性的选项：

选项	说明
ASCII（简化）	使用 ASCII 字符返回地址。
ASCII（扩展）	使用 ASCII 字符返回地址，并扩展地址中的任何特殊字符。例如，Ö 音译为 OE。
数据库	使用地址引用数据为默认语言使用的字符集来返回地址。 数据库为默认值。
拉丁文	使用拉丁文字符集返回地址。
拉丁文（备用）	使用备用拉丁字符集返回地址。 例如，指定拉丁文以使用国语罗马字表记法返回韩国地址。指定拉丁文（备用）以使用旧式的 ISO/TR 11941 音译法返回韩国地址。
邮政管理	使用地址的本地邮政服务首选的脚本返回地址。
邮政管理（备用）	使用地址的本地邮政服务批准作为替代脚本的脚本返回地址。
保留输入	采用输入地址所使用的字符集返回地址数据。

该转换可以对包含采用多种语言和字符集的数据的数据源进行处理。 该转换会将所有输入数据转换为 Unicode UCS-2 字符集，并以 UCS-2 格式处理数据。 处理数据后，该转换会将每个地址记录中的数据转换为在属性中指定的字符集。 该过程称为音译。

音译转换字符以供处理时，可以使用字符集中每个字符的数值表示。 字符没有等效的数值表示时，音译也可以按照发音转换字符。 如果地址验证器转换无法将字符映射到 UCS-2，则会将该字符转换为空格。

注意: 如果更新了该转换的首选语言或首选脚本，请确认所选择的语言和字符代码可以兼容。

要扩展的范围

确定地址验证器转换如何为未指定门牌号的街道地址返回建议地址。 当转换以建议列表模式运行时使用该属性。

地址验证器转换以建议列表模式读取部分或不完整的街道地址。 转换将地址与地址引用数据进行比较，并向最终用户返回所有类似地址。 如果输入地址不包含门牌号，则转换可以为街道返回一个或多个门牌号建议。 “要扩展的范围”属性确定转换如何返回地址。

转换可以在单个地址中返回有效门牌号范围，或者可以为每个有效门牌号返回一个单独的地址。 转换还可以为街道上从最低到最高门牌号范围中的每个门牌号返回一个地址。

下表介绍了有关属性的选项：

选项	说明
全部	地址验证为街道上可能门牌号范围中的每个门牌号返回一个建议地址。
无	地址验证将返回一个标识街道有效范围内最低和最高门牌号的地址。
仅包含有效项目	地址验证为地址引用数据识别为可交付地址的每个门牌号返回一个建议地址。

注意: 建议列表模式可以使用地址中的其他元素指定街道编号有效范围。 例如，邮政编码可能标识包含地址邮箱的城市街区。 地址验证器转换可以使用邮政编码标识街区上的最低和最高有效门牌号。

如果转换无法确定实际限制之内的门牌号范围，则建议地址的编号会增大到不可用的大小。 要限制“要扩展的范围”属性生成的地址编号，请将“灵活范围扩展”属性设置为“开”。

标准化无效地址

决定地址验证过程是否将不可投递地址中的数据值标准化。该属性应用于返回范围为 I1 到 I4 的匹配代码状态的地址记录。

将数据标准化后，可提高下游数据进程返回准确结果的可能性。例如，对于两个以相同格式表示通用地址元素的地址记录，重复分析映射可能会返回更高的匹配得分。

地址验证可以标准化以下地址元素：

- 街道后缀元素，如“路”和“大街”。
- 前置和后置方向元素，如“北”、“南”、“东”和“西”。
- 投递服务元素，如“邮政信箱”。
- 子建筑物元素，如“公寓”、“楼层”和“房间”。
- 州或省/自治区/直辖市名称。 标准化将返回名称的缩写形式。

下表介绍了有关属性的选项：

选项	说明
关闭	地址验证不会更正数据错误。默认选项。
打开	地址验证更正数据错误。

参数的用法

您可以分配一个参数来指定数据错误的标准化策略。输入 OFF 或 ON 作为参数值。 以大写字母输入值。

跟踪级别

设置包括在日志中的详细信息量。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

认证报表

您可以生成一个报表，描述在认证模式下提交验证的地址的状态。该报表可验证地址集是否满足邮件运营商规定的认证标准。

地址验证器转换将根据以下标准生成报表：

地址计算机审核系统 (AMAS)

澳大利亚邮政规定 AMAS 认证标准。

编码准确性支持系统 (CASS)

USPS 规定 CASS 认证标准。

SendRight

新西兰邮政规定 SendRight 认证标准。

软件评估和识别程序 (SERP)

加拿大邮政规定 SERP 认证标准。

注意: 可以使用地址验证器转换将法国地址记录认证为国家地址管理服务 (SNA) 认证标准。但是，不能在地址验证器转换中为 SNA 地址认证生成报表。

提供符合邮件运营商认证标准的邮件时，需提交认证报表与地址集。该报表包含有关组织的数据。可在配置地址验证器转换时输入此数据。转换过程会向报表文件中添加组织数据。

AMAS 报表字段

配置 AMAS 报表时，提供向澳大利亚邮政提交认证地址记录集的组织的相关信息。保存或打印报表，并在报表中包含向澳大利亚邮政提交的地址记录。

使用**报表**视图输入信息。

下表介绍了您输入的信息：

字段	说明
报表文件名	地址验证创建的报表的名称和位置。默认情况下，地址验证器将在数据集成服务计算机的 `bin` 目录中创建报表。 要将报表文件写入数据集成服务计算机上的其他位置，请输入文件路径和文件名。可以输入完全限定的路径或相对路径。 相对路径使用 `bin` 目录作为其根目录。 指定的目录必须存在才能运行地址验证映射。
地址列表名称	向澳大利亚邮政提交的地址记录集的名称。
列表处理器名称	提交地址记录集的组织的名称。
列表管理器/所有者的名称	组织中地址数据的管理者或所有者的名称。
电话号码	提交地址记录集的组织的联系人电话号码。
地址	提交地址记录集的组织的地址。

相关主题：

- “定义认证报表” 页面上 115

CASS 报表字段

配置 CASS 报表时，请提供与将已验证的地址记录集提交到 USPS 的组织有关的信息。 保存或打印报表，并在报表中包含提交到 USPS 的地址记录。

使用**报表**视图输入信息。

下表介绍了您输入的信息：

字段	说明
报表文件名	地址验证创建的报表的名称和位置。默认情况下，地址验证器将在数据集成服务计算机的 `bin` 目录中创建报表。 要将报表文件写入到数据集成服务计算机上的其他位置，请输入文件路径和文件名。可以输入完全限定的路径或相对路径。相对路径使用 `bin` 目录作为其根目录。 指定的目录必须存在才能运行地址验证映射。
列表名称/ID	提交给邮件运营商的地址列表的名称或标识号码。
列表处理器名称	执行地址验证的组织的名称。
名称/地址	执行地址验证的组织的邮寄名称和地址。

注意: 对于美国地址，地址验证器转换将从批处理和交互式引用数据文件读取有关 CASS 引用数据文件的元数据。要生成 CASS 报告，转换必须读取当前的 CASS 引用数据文件以及当前的批处理和交互式引用数据文件。

相关主题：

- “定义认证报表” 页面上 115

SendRight 报告

配置 SendRight 报表时，应提供向新西兰邮政提交认证地址记录集的组织的相关信息。 保存或打印该报表，并在报表中加入您提交到新西兰邮政的地址记录。

使用**报表**视图输入信息。

下表介绍了您输入的信息：

字段	说明
客户名称	提交地址记录集的组织的名称。
客户 NZP 编号	提交地址记录集的组织的新西兰邮政账号。 如果某个邮售商店代表该组织提交记录，请输入该邮售商店运输标识 (TPID) 号。
客户数据库	包含地址记录集的文件的名称。 地址验证器转换在您于内容管理服务上指定的位置创建报表。 使用管理员工具设置该位置。
客户地址	提交地址记录集的组织的地址。

相关主题：

- “定义认证报表” 页面上 115

SERP 报表字段

配置 SERP 报表时，提供向加拿大邮政提交认证地址记录集的组织的相关信息。 保存或打印报表，并在报表中包含向加拿大邮政提交的地址记录。

使用**报表**视图输入信息。

下表介绍了您输入的信息：

字段	说明
报表文件名	地址验证创建的报表的名称和位置。默认情况下，地址验证器将在数据集成服务计算机的 `bin` 目录中创建报表。 要将报表文件写入到数据集成服务计算机上的其他位置，请输入文件路径和文件名。可以输入完全限定的路径或相对路径。相对路径使用 `bin` 目录作为其根目录。 指定的目录必须存在才能运行地址验证映射。
客户 CPC 号	加拿大邮政公司向执行地址验证的组织签发的客户编号。
客户名称/地址	执行地址验证的组织的名称和地址。

相关主题：

- “定义认证报表” 页面上 115

配置地址验证器转换

使用地址验证器转换验证和改进邮寄地址数据的质量。

地址验证器转换将读取地址引用数据。 验证 Developer 工具是否可以访问所需的地址引用数据文件。

1. 打开转换。
2. 单击**常规设置**视图，然后配置常规属性。
3. 单击**模板**视图以添加输入端口和输出端口。
4. 单击**报表**视图以生成邮政服务地址认证的报表。
5. 单击**高级**视图以配置高级地址验证属性。
6. 连接输入端口和输出端口。

   注意: 将不希望地址转换验证的输入端口与**传递**输入端口组连接。

将端口添加到地址验证器转换

使用**模板**视图将端口添加到地址验证器转换中。

1. 单击**模板**视图。

2. 展开模板。
   - 选择**基本模型**模板添加常见地址字段。
   - 选择**高级模型**模板添加专用地址字段。
3. 展开与输入数据的格式对应的输入端口组。 输入端口组为**离散值**、**多行**和**混合**。
4. 选择输入端口。

   提示: 按 CTRL 键选择多个端口。
5. 右键单击端口，然后选择**将端口添加到转换**。
6. 展开包含所需字段的输出端口组。
7. 右键单击端口，然后选择**将端口添加到转换**。
8. 要为不希望验证的列添加传递端口，请单击**端口**选项卡，选择**传递**输入端口组，然后单击**新建**。

创建用户定义的模板

创建模板以便对计划重用的地址端口进行分组。

通过从基本模板和高级模板中选择端口来创建自定义模板。 创建后续地址验证器转换时，可以选择自定义模板。

注意: 模板不是存储库对象。 模板驻留在用于创建模板的计算机上。

1. 选择**模板**视图。
2. 单击**新建**。
3. 键入模板的名称。
4. 展开**基本模型**或**高级模型**模板，然后选择所需端口。
5. 单击**确定**。

定义地址验证器模型

地址验证器模型将定义地址验证器转换的默认输入端口和输出端口。

地址验证器转换不包含默认输入端口和输出端口。 但是，可以定义模型以指定地址验证器转换使用的输入端口和输出端口。

注意: 模型不是存储库对象。 模型驻留在用于创建模型的计算机上。

要定义地址验证器模型，请执行以下步骤：

1. 选择**模板**视图。
2. 展开**基本模型**或**高级模型**模板，然后选择所需端口。
3. 选择**使用选定端口创建默认 AV 模型**。
4. 要重置模型并删除所有端口，请选择**清除默认 AV 模型**。

定义认证报表

在地址验证器转换中定义认证报表时，请配置**常规设置**和**报表**视图上的选项。

1. 在**常规设置**视图上，将**模式**选项设置为认证。
2. 在**报表**视图上，选择要生成的报表类型。可以选择以下报表类型：

选项	说明
AMAS 报表	包含澳大利亚邮政处理记录集所需的信息。
CASS 报表	包含 USPS 处理记录集所需的信息。
SendRight 报告	包含新西兰邮政处理记录集所需的信息。
SERP 报表	包含加拿大邮政处理记录集所需的信息。

3. 为每个选定的报表类型输入报表详细信息。

将报表文件提交给邮件运营商，并提交使用地址验证器转换验证的地址记录的列表。

相关主题：

- “AMAS 报表字段” 页面上 111
- “CASS 报表字段” 页面上 112
- “SendRight 报告” 页面上 112
- “SERP 报表字段” 页面上 113

地址验证器转换在非本地环境中

非本地环境中的地址验证器转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。有限支持。

地址验证器转换在 Blaze 引擎上

支持地址验证器转换，但存在以下限制：

- 地址验证器转换无法生成认证报表。
- 当配置为在“交互”模式或“建议列表”模式下运行时，地址验证器转换通不过验证。

地址验证器转换在 Spark 引擎上

支持地址验证器转换，但存在以下限制：

- 地址验证器转换无法生成认证报表。

- 当配置为在“交互”模式或“建议列表”模式下运行时，地址验证器转换通不过验证。
- 使用地址验证器转换运行的映射会忽略转换中**执行实例**高级属性上的值。

地址验证器转换在流映射中

在 Spark 引擎上流映射与批处理映射具有相同的处理规则。

Databricks Spark 引擎上的地址验证器转换

支持地址验证器转换，但存在以下限制：

- 地址验证器转换无法生成认证报表。
- 当配置为在“交互”模式或“建议列表”模式下运行时，地址验证器转换通不过验证。
- 使用地址验证器转换运行的映射会忽略转换中**执行实例**高级属性上的值。

第 5 章

汇总器转换

本章包括以下主题：

汇总器转换概览

配置一个汇总器转换来执行汇总计算，例如，求一组数据的平均值与和。 可以使用汇总器转换来删除重复行。 汇总器转换是主动转换。

汇总器转换与表达式转换不同：汇总器转换可以对一组数据执行计算， 而表达式转换针对每一行返回结果。

例如，您可以计算组织内每个部门的员工平均薪资。 首先按部门编号配置组。 然后，配置一个表达式来计算平均薪资并针对每个唯一部门编号返回结果。

使用转换语言来创建汇总表达式。

数据集成服务在汇总缓存中读取和存储数据时，将执行汇总计算。 您可以通过对输入数据进行排序来提高性能。 如果对输入数据进行了排序，数据集成服务将不创建缓存。

动态映射中的汇总器转换

可以在动态映射中使用汇总器转换。可以在转换中配置动态端口，并引用生成的端口。

可执行以下任务以在动态映射中配置汇总器转换：

将动态端口或生成的端口作为“分组依据”列引用。

可以将动态端口或生成的端口作为“分组依据”列包括在内。如果将某个动态端口指定为“分组依据”列，即表示您将该动态端口中所有生成的端口都指定为“分组依据”列。因此，如果您将父动态端口指定为“分组依据”列，则无法将生成的端口指定为“分组依据”列。如果引用生成的端口，但该端口在运行时不存在，映射将失败。可以参数化“分组依据”列以指示在运行时要进行分组的列。

在汇总表达式中引用生成的端口。

可以在汇总表达式中包括生成的端口。如果该端口在运行时不存在，映射将失败。无法在汇总表达式中引用动态端口。

在输出端口中创建汇总表达式。

无法在生成的端口或动态端口中创建汇总表达式。汇总表达式不能是动态表达式。

在汇总表达式中参数化值。

可以在汇总表达式中包括参数， 但无法在其中使用表达式参数或端口参数。

开发汇总器转换

创建汇总器转换时，需要定义要对每个行运行的表达式。 定义一个分组依据端口列表，以按其返回结果。

要创建汇总器转换，请执行以下步骤：

1. 定义转换并创建端口。
2. 在变量或输出端口上配置汇总器转换。
3. 定义一组端口，以按其返回汇总结果。

汇总器转换端口

汇总器转换具有多个端口类型，您可以使用这些端口类型来配置组并汇总表达式。

“汇总器转换端口”视图具有以下字段：

名称

端口的名称。

类型

端口数据类型。

精度

字段长度。

小数位数

数值数据小数点右侧的位数。

输入

指示数据来自上游转换。

输出

指示端口返回表达值的值。 表达式可包含非汇总表达式和条件子句。 可以创建多个汇总输出端口。

传递

指示端口是返回未更改数据的输入-输出端口。

变量

指示端口可以存储要在表达式中使用的值或计算。 变量端口不能是输入端口或输出端口。 变量端口仅在该转换内传递数据。

表达式

用于汇总行或行组的表达式。

默认值

包含无效值或空值的端口的默认值。

输入规则

一组规则，用于根据端口名称或数据类型对要在转换中包括或排除的端口进行筛选。 可在定义动态端口时配置输入规则。

汇总表达式

在汇总器转换的变量端口或输出端口中配置汇总表达式。 如果端口为输入端口、传递端口或动态端口，则无法输入表达式。

汇总表达式可包含条件子句和非汇总函数。 汇总函数还可以包括嵌套在另一汇总函数内的汇总函数，例如：

```
MAX( COUNT( ITEM ))
```

汇总表达式的结果因转换中的分组依据端口而异。 例如，以下汇总表达式将查找已售项目的总数量：

```
SUM( QUANTITY )
```

但是，如果使用同一表达式，并按 ITEM 端口分组，则数据集成服务会按项目返回总数量。

可以在任何输出端口中创建汇总表达式，并在转换中使用多个汇总端口。

汇总函数

在汇总器转换中配置汇总函数。可以将一个汇总函数嵌套在另一汇总函数中。

转换语言包括以下汇总函数：

- ANY
- AVG
- COLLECT_LIST
- COLLECT_MAP
- COUNT
- FIRST

- LAST
- MAX(Date)
- MAX (Number)
- MAX (String)
- MEDIAN
- MIN (Date)
- MIN (Number)
- MIN (String)
- PERCENTILE
- STDDEV
- SUM
- VARIANCE

如果在汇总器转换的表达式中使用某个端口但未在汇总函数中使用该端口，则数据集成服务将使用该端口中的最后一行来处理表达式。

例如，创建一个包含端口 COMMISSIONS 和 SALARY 的汇总器转换。端口 SALARY 为分组端口。

您可能在输出端口中使用以下表达式：

```
SUM(COMMISSIONS)
```

数据集成服务处理汇总器函数并在输出端口中返回端口 COMMISSIONS 中的值总和。

您可能将该表达式修改为以下表达式：

```
SUM(COMMISSIONS) + COMMISSIONS
```

为了处理该表达式，数据集成服务将返回端口 COMMISSIONS 中的值总和，并将端口 COMMISSIONS 中最后一行的值与输出端口中的返回值相加。

对于不同的输出端口，您可能使用以下表达式：

```
SUM(COMMISSIONS) + SALARY
```

为了处理该表达式，数据集成服务将返回端口 COMMISSIONS 中的值总和，并将端口 SALARY 中最后一行的值与输出端口中的返回值相加。请注意，端口 SALARY 的每一行中的值均相同，因为 SALARY 端口为分组端口。

嵌套汇总函数

可以在汇总器转换的不同输出端口中包含多个单层函数或多个嵌套函数。

汇总器转换中不能同时包含单层函数和嵌套函数。 因此，如果汇总器转换的任意输出端口中包含单层函数，则不能在该转换的任何其他端口中使用嵌套函数。 将单层函数和嵌套函数包含在同一汇总器转换中时，Developer 工具会将映射或 Mapplet 标记为无效。 如果需要同时创建单层函数和嵌套函数，请创建单独的汇总器转换。

汇总表达式中的条件子句

在汇总表达式中使用条件子句可减少汇总中使用的行数。 条件子句可以是计算结果为 TRUE 或 FALSE 的任何子句。

例如，使用以下表达式计算超过其季度配额的员工的佣金总额：

```
SUM( COMMISSION, COMMISSION > QUOTA )
```

分组依据端口

您可以定义要进行汇总的行组，而不是对所有输入数据运行汇总。例如，可以计算公司销售总额，也可以查找按区域分组的公司销售总额。

要为汇总表达式定义组，请在汇总器转换中选择相应的输入、输入/输出、输出和变量端口。可以选择多个分组依据端口来为每个唯一的组合创建新组。然后数据集成服务会为每个组执行定义的汇总。

对值进行分组时，数据集成服务会针对每个组生成一行。如果不对值进行分组，则数据集成服务会针对所有输入行返回一行。数据集成服务返回每个组的最后一行，其中包含汇总结果。您可以指定返回特定行。例如，如果您使用 FIRST 汇总器函数，数据集成服务将返回第一行。

在汇总器转换中选择多个分组依据端口时，数据集成服务通过端口顺序来确定分组顺序。组顺序会影响结果。对分组依据端口进行排序可确保分组恰当。选择组中的端口后，可以更改端口顺序。

例如，可以创建一个名为 Price_Out 的输出端口。Price_Out 的表达式为 SUM (Qty * Price)。将 Store_ID 和 Item 定义为分组依据端口。转换将按照门店返回每个项目的总价格。

输入行可能包含以下数据：

Store_ID	Item	Qty	Price
101	battery	3	2.99
101	battery	1	3.19
101	battery	2	2.59
101	AAA	2	2.45
201	battery	1	1.99
201	battery	4	1.59
301	battery	1	2.45

数据集成服务对以下唯一的组执行汇总计算：

Store_Id	Item
101	battery
101	AAA
201	battery
301	battery

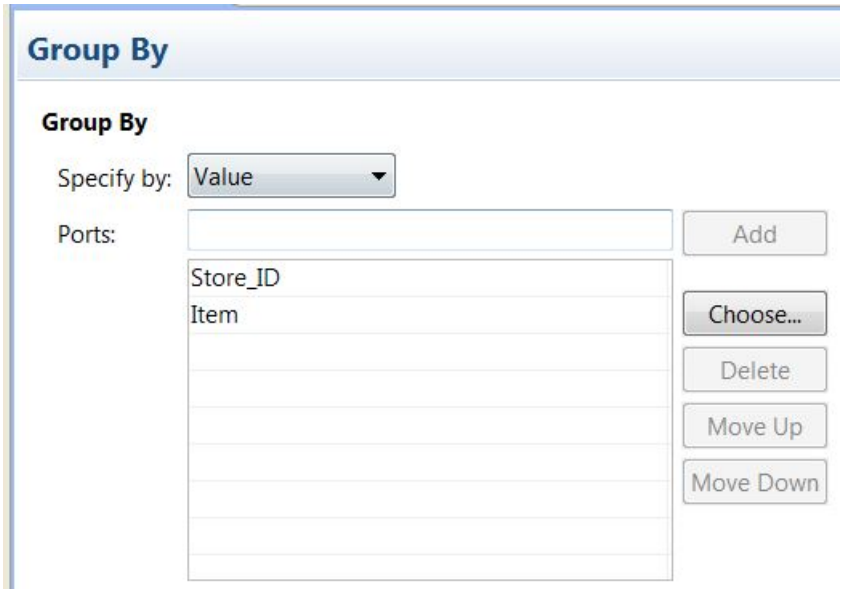
数据集成服务从最后一行返回 Store_ID、Item、Qty 和 Price，其中包含按照门店划分的每个项目的总和 (Price * Qty)。

Store_ID	Item	Qty	Price	Price_Out
101	battery	2	2.59	17.34
101	AAA	2	2.45	4.90
201	battery	4	1.59	8.35
301	battery	1	2.45	2.45

配置分组依据端口

在转换**属性**视图的**分组依据**选项卡中定义分组依据端口。

下图显示了“分组依据”选项卡：



“分组依据”选项卡包含以下选项：

指定依据

选择**值**或**参数**。 选择**值**可使用端口名称。 选择**参数**可使用端口列表参数。

添加

接受手动键入的端口名称。 单击**添加**之前必须键入有效的端口名称。

选择

单击**选择**可选择要添加到组中的端口。 Developer tool 提供一个来自转换的端口列表，您可从中进行选择。

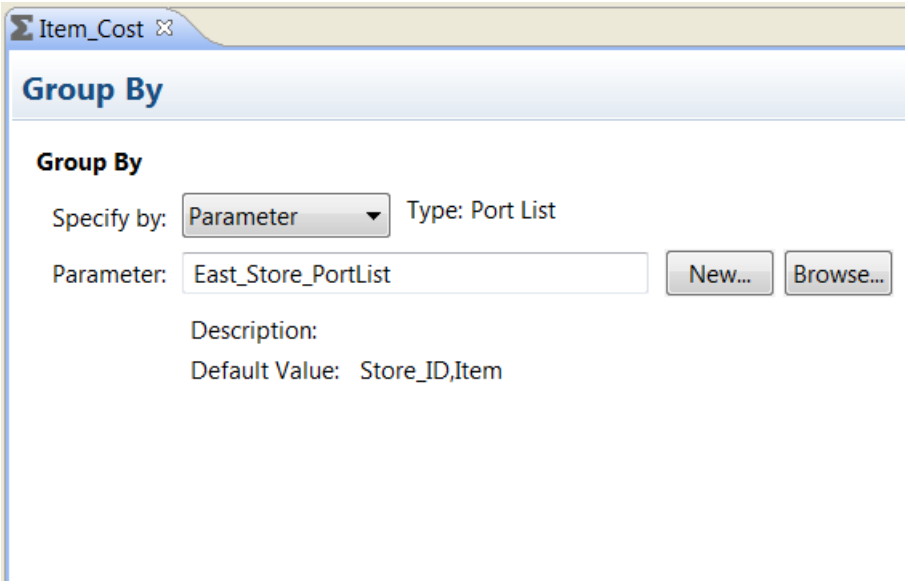
“上移”和“下移”

可以更改组中的端口顺序。 选择端口名称，然后单击其中一个移动按钮，可在排序顺序中将其上移或下移。

分组依据参数

可以配置一个端口列表参数，其中包含要加入到组中的一个或多个端口。 通过在转换中从端口列表选择端口，可以创建端口列表参数。

下图显示了在您使用参数来标识组中的端口时出现的**分组依据**选项卡：



可以浏览端口列表参数，也可以单击**新建**创建端口列表参数。 如果选择创建端口列表参数，则可在转换中从端口列表选择端口。

分组依据端口的默认值

分组依据端口包含空值时，数据集成服务不创建组。 可为组中的每个端口定义默认值以替换任何空输入值。 然后，数据集成服务可以将行数包含在汇总总数中。

非汇总表达式

可在分组依据端口中使用非汇总表达式来修改或替换组。

例如，如果要在分组之前替换“AAA battery”，可以使用以下表达式创建名为 CORRECTED_ITEM 的分组依据输出端口：

```
IIF( ITEM = 'AAA battery', battery, ITEM )
```

汇总器缓存

当您运行使用汇总器转换的映射时，数据集成服务将在内存中创建索引缓存和数据缓存来运行该转换。如果数据集成服务需要的空间大于内存缓存中的可用空间，它会将溢出数据存储在缓存文件中。

数据集成服务为汇总器转换创建以下缓存：

- 索引缓存，按照分组依据端口中的配置存储组值。
- 数据缓存，按照分组依据端口存储计算。

数据集成服务不使用高速缓存运行具有已排序端口的汇总器转换。您无需配置使用已排序端口的汇总器转换的高速缓存。

汇总器转换的已排序输入

您可以通过已排序输入选项提高汇总器转换的性能。

使用已排序输入时，数据集成服务假定所有数据已按组排序，并在读取某个组的行时执行汇总计算。 必要时，数据集成服务会将组信息存储在内存中。 要使用已排序输入选项，必须将已排序数据传递至汇总器转换。如果使用已排序输入，则汇总器转换会提供已排序输出。

如果不使用已排序输入，则数据集成服务将在读取时执行汇总计算。 由于数据未排序，因此，数据集成服务将存储每个组的数据，直至读取整个源，以确保所有汇总计算准确无误。

例如，如果选择已排序输入选项，则汇总器转换将具有 STORE_ID 和 ITEM 分组依据端口。 在向汇总器传递以下数据后，数据集成服务将在发现组“201/电池”后对“101/电池”组中的三行执行汇总。

STORE_ID	ITEM	QTY	PRICE
101	'battery'	3	2.99
101	'battery'	1	3.19
101	'battery'	2	2.59
201	'battery'	4	1.59
201	'battery'	1	1.99

如果使用已排序的输入并且没有预先对数据进行正确排序，数据集成服务将使映射运行失败。

已排序的输入条件

某些条件可防止使用已排序的输入。

出现以下任一情况时，您无法使用已排序输入：

- 汇总表达式包含嵌套汇总函数。
- 源数据为数据驱动。

出现其中任一情况时，数据集成服务将如未使用已排序输入一样处理该转换。

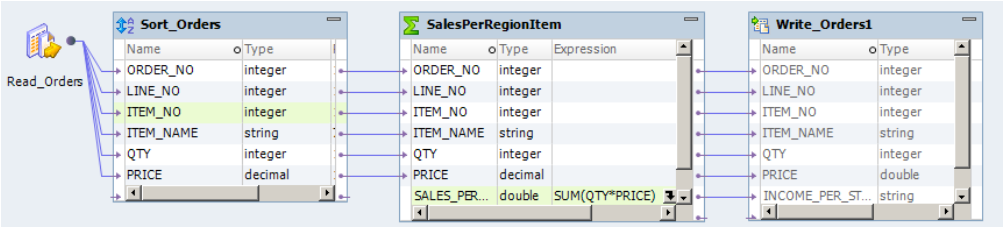
对汇总器转换中的数据进行排序

要使用已排序输入，可以将已排序数据传递到汇总器转换。

您必须按汇总器分组依据端口在汇总器转换中显示的顺序对数据进行排序。

对于关系和平面文件输入，请在将映射中的数据传递至汇总器转换之前，使用排序器转换对其进行排序。如果没有转换更改已排序数据的顺序，则可以在汇总器之前将排序器转换放置在映射中的任意位置。汇总器转换中“分组依据”列的顺序必须与其在排序器转换中显示的顺序相同。

以下映射显示了一个排序器转换，该排序器转换配置为按 ITEM_NO 对源数据进行升序顺序排序：



排序器转换对数据进行如下排序：

ITEM_NO	ITEM_NAME	QTY	PRICE
345	Soup	4	2.95
345	Soup	1	2.95
345	Soup	2	3.25
546	Cereal	1	4.49
546	Cereal	2	5.25

使用已排序输入后，汇总器转换将返回以下结果：

ITEM_NAME	QTY	PRICE	INCOME_PER_ITEM
Cereal	2	5.25	14.99
Soup	2	3.25	21.25

汇总器转换高级属性

配置有助于确定数据集成服务如何为汇总器转换处理数据的属性。

为汇总器转换配置以下高级属性：

缓存目录

数据集成服务创建索引缓存文件和数据缓存文件的目录。请确认该目录存在并含有足够的磁盘空间可用于缓存文件。

输入多个以分号分隔的目录，以提高缓存分区期间的性能。缓存分区会为处理转换的每个分区创建一个单独的缓存。

默认值为 CacheDir 系统参数。您可以为此属性配置另一个系统参数或用户定义的参数。

数据缓存大小

数据集成服务在映射运行开始时为转换的数据缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

索引缓存大小

数据集成服务在映射运行开始时为转换的索引缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

已排序输入

指示输入数据已按组预先排序。仅在映射将已排序数据传递到汇总器转换时选择此选项。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

相关主题：

- “缓存大小” 页面上 67

创建可重用汇总器转换

可创建可重用汇总器转换以供在多个映射或 Mapplet 中使用。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择汇总器转换。
4. 单击**下一步**。
5. 输入转换的名称。
6. 单击**完成**。

   此时，转换将显示在编辑器中。
7. 单击**新建**按钮向转换添加端口。
8. 编辑端口以设置名称、数据类型和精度。
9. 确定每个端口的类型：输入、输出、传递或变量。
10. 单击“表达式”字段为输出端口配置汇总表达式。您可以选择要用于定义汇总表达式的端口和参数。
11. 单击**高级**视图并编辑转换属性。

创建不可重用汇总器转换

可在映射或 Mapplet 中创建不可重用汇总器转换。

1. 在映射或 Mapplet 中，将“转换”选项板中的汇总器转换拖动到编辑器中。

   此时，转换将显示在编辑器中。
2. 在**属性**视图中，编辑转换名称和说明。
3. 在**端口**选项卡中，单击**新建**按钮将端口添加到转换。
4. 编辑端口以设置名称、数据类型和精度。
5. 确定每个端口的类型：输入、输出、传递或变量。

6. 配置输出端口的汇总表达式。
7. 在**高级**视图中，编辑转换属性。

汇总器转换提示

可以使用提示以更有效地使用汇总器转换。

使用已排序输入以减少汇总缓存的使用。

已排序输入可减少映射运行期间缓存的数据量并提高性能。请将此选项与排序器转换配合使用，以将已排序数据传递到汇总器转换。

限制连接的输入/输出或输出端口。

限制连接的输入/输出端口或输出端口的数量可降低汇总器转换在数据缓存中存储的数据量。

在汇总数据之前对其进行筛选。

如果您在映射中使用筛选器转换，请在执行汇总器转换之前执行该转换，以减少不必要的汇总。

只有已排序的汇总器转换提供已排序输出。

如果您使用未排序输入并要生成已排序输出，请先使用汇总器转换再使用排序器转换。

汇总器转换故障排除

您可以对汇总器转换进行故障排除。

我选择了已排序输入，但是映射花费的时间与之前相同。

出现以下任一情况时，您无法使用已排序输入：

- 汇总表达式包含嵌套汇总函数。
- 源数据为数据驱动。

出现其中任一情况时，数据集成服务将如未使用已排序输入一样处理该转换。

包含汇总器转换的映射会降低性能。

数据集成服务可能对磁盘进行分页。您可以通过增加转换属性中的索引和数据缓存大小来提高性能。

非本地环境中的汇总器转换

非本地环境中的汇总器转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。

- Databricks Spark 引擎。有限支持。

Blaze 引擎上的汇总器转换

Blaze 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

在下列情况下，映射验证会失败：

- 转换包含有状态变量端口。
- 转换在表达式中包含不受支持的函数。

汇总函数

如果在汇总器转换的表达式中使用某个端口但未在汇总函数中使用该端口，则运行时引擎可能会使用该端口中的任何行来处理表达式。

运行时引擎使用的行可能不是端口中的最后一行。处理是分布式的，因此运行时引擎可能无法确定端口中的最后一行。

数据缓存优化

汇总器转换的数据缓存已经过优化，可使用可变长度来存储通过汇总器转换的 binary 和 string 数据类型。系统对最大为 8 MB 的记录启用此优化。如果记录大小大于 8 MB，则会禁用可变长度优化。

使用可变长度在数据缓存中存储通过汇总器转换的数据时，系统会将汇总器转换优化为使用已排序输入，并在运行时映射中的汇总器转换之前插入传递排序器转换。

要查看排序器转换，请查看优化的映射或查看 Blaze 验证环境中的执行计划。

在数据缓存优化期间，汇总器转换的数据缓存和索引缓存设置为“自动”。排序器转换的排序器缓存设置为与汇总器转换的数据缓存相同的大小。要配置排序器缓存，必须为汇总器转换配置数据缓存的大小。

Spark 引擎上的汇总器转换

Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

在下列情况下，映射验证会失败：

- 转换包含有状态变量端口。
- 转换在表达式中包含不受支持的函数。

汇总函数

如果在汇总器转换的表达式中使用某个端口但未在汇总函数中使用该端口，则运行时引擎可能会使用该端口中的任何行来处理表达式。

运行时引擎使用的行可能不是端口中的最后一行。处理是分布式的，因此运行时引擎可能无法确定端口中的最后一行。

数据缓存优化

无法将转换的数据缓存优化为使用可变长度存储数据。

流映射中的汇总器转换

流映射具有不适用于批处理映射的其他处理规则。

映射验证

在下列情况下，映射验证会失败：

- 流管道中包含多个汇总器转换。
- 流管道中包含汇总器转换和等级转换。
- 汇总器转换在查找转换的上游。
- 汇总器转换与配置了不相等查找条件的被动查找转换处于同一流管道中。

Databricks Spark 引擎上的汇总器转换

Databricks Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

在下列情况下，映射验证会失败：

- 转换包含有状态变量端口。
- 转换在表达式中包含不受支持的函数。

汇总函数

如果在汇总器转换的表达式中使用某个端口但未在汇总函数中使用该端口，则运行时引擎可能会使用该端口中的任何行来处理表达式。

运行时引擎使用的行可能不是端口中的最后一行。处理是分布式的，因此运行时引擎可能无法确定端口中的最后一行。

数据缓存优化

无法将转换的数据缓存优化为使用可变长度存储数据。

第 6 章

关联转换

本章包括以下主题：

关联转换概览

关联转换会处理来自匹配转换的输出数据。 它会在分配给不同匹配群集的重复记录之间创建链接，从而使这些记录可以在数据合并和主数据管理操作中相互关联。

关联转换会为一组关联记录中的每行生成一个 **AssociationID** 值，并将 ID 值写入输出端口。

整合转换会读取来自关联转换的输出。 使用整合转换可基于具有公用关联 ID 值的记录创建一个主记录。

关联转换会在输入端口上接受字符串和数值数据值。 如果添加其他数据类型的输入端口，则转换会将端口数据值转换为字符串。

AssociationID 输出端口会写入整数数据。 如果转换是在早期版本的 Informatica Data Quality 中配置的，则该转换可以在 AssociationID 端口上写入字符串数据。

示例：关联匹配转换输出

下表包含三条记录，这些记录可能标识的是同一个人：

ID	名称	地址	城市	省/自治区/直辖市	邮政编码	SSN
1	David Jones	100 Admiral Ave.	纽约	NY	10547	987-65-4321
2	Dennis Jones	1000 Alberta Ave.	新泽西	NY	-	987-65-4321
3	D. Jones	Admiral Ave.	纽约	NY	10547-1521	-

在匹配转换中定义的重复项分析操作不会将所有三条记录视为彼此重复，原因如下：

- 如果对姓名和地址数据定义了重复搜索，则记录 1 和 3 会被视为重复，而记录 2 将被省略。
- 如果对姓名和社会保障号数据定义了重复搜索，则记录 1 和 2 会被视为重复，而记录 3 将被省略。
- 如果对所有三个属性（姓名、地址和社会保障号）定义了重复搜索，则匹配转换可能不会将其中任何一条记录视为匹配项。

关联转换会将来自不同匹配群集的数据链接在一起，以便具有相同群集 ID 的记录会获得一个公用的 AssociationID 值。 在本示例中，所有三条记录都会获得一个相同的 AssociationID，如下表中所示：

ID	名称	地址	城市	省/自治区/直辖市	邮政编码	SSN	姓名和地址群集 ID	姓名和 SSN 群集 ID	关联 ID
1	David Jones	100 Admiral Ave.	纽约	NY	10547	987-65-4320	1	1	1
2	Dennis Jones	1000 Alberta Ave.	新泽西	NY	-	987-65-4320	2	1	1
3	D. Jones	Alberta Ave.	纽约	NY	10547-1521	-	1	2	1

可以在整合转换中合并重复的记录数据。

内存分配

可以设置关联转换使用的最小缓存量。 默认设置为 400,000 字节。

在**高级**选项卡的**缓存文件大小**属性中设置值。

默认值表示转换所使用的最小内存量。 根据关联的端口数量，关联转换会尝试获取此默认值的倍数。 转换使用以下公式来获取缓存：

（关联端口数 + 1）x 默认缓存大小

例如，如果配置七个关联端口，则转换会尝试为缓存分配 320 万字节或 3.05 MB。

如果更改了默认设置，则转换不会尝试获取更多内存。

注意: 如果输入的缓存值低于 65536，则关联转换将以 MB 为单位读取值。

关联转换高级属性

关联转换包含确定高速缓存行为和跟踪级别的高级属性。

您可以配置以下高级属性：

缓存文件目录

指定数据集成服务将当前转换的临时数据写入到的目录。当输入数据量超出可用系统内存时，数据集成服务会将临时文件写入该目录。数据集成服务会在运行映射后删除这些临时文件。

您可以在该属性上输入目录路径，也可以使用参数标识目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。默认值为 CacheDir 系统参数。

缓存文件大小

确定数据集成服务对转换的输入数据排序使用的系统内存量。

在对数据排序之前，数据集成服务会分配您指定的内存量。如果排序操作生成的数据较多，数据集成服务会将超出的数据写入缓存文件目录。如果排序操作所需的内存超出系统内存和文件存储可以提供的内存，映射会失败。

转换以字节为单位读取值。默认值为 400,000 字节。最大值为 2,147,483,647 字节。可以使用参数指定缓存文件大小。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

第 7 章

离散记录异常转换

本章包括以下主题：

离散记录异常转换概览

离散记录异常转换是一种主动转换，可读取数据质量处理的输出，并标识需要手动查看的记录。离散记录异常转换属于多组转换。

配置离散记录异常转换以分析标识记录中的数据质量问题的过程的输出。包含需要进一步查看的数据质量问题的记录属于异常。

离散记录异常转换从其他转换或其他映射中的数据对象接收输入。离散记录转换的输入必须包含一个或多个质量问题端口，可接收数据质量问题的文本说明。离散记录异常转换的输入还可以包含该转换可用来确定每个记录的数据质量的数字记录得分。在异常转换中设置得分阈值的上界和下界，以根据记录得分将记录的质量分为正常和离散两类。离散记录异常转换将异常和关联的质量问题写入到离散记录表中。

例如，组织需要在将一些邮件发送给客户之前对客户地址进行验证。开发人员创建一个映射，以使用标签创建器转换根据引用表验证客户的城市、省/自治区/直辖市和邮政编码。标签创建器转换验证字段，并根据结果在每一行中添加记录得分。标签创建器转换还为每个有错的记录添加描述质量问题的文本。标签创建器转换为每个异常添加质量问题文本，如城市无效或邮政编码为空。离散记录异常转换将需要手动查看的客户记录写入离散记录表中。数据分析师可在 Analyst 工具中查看和更正离散记录。

离散记录异常输出记录类型

离散记录异常检查输入记录得分，以确定记录质量。 其将记录返回到不同的输出组。

异常转换根据每个记录得分标识以下类型的记录：

正常记录

得分大于或等于阈值上界的记录。 正常记录有效，并且无需查看。 例如，如果将阈值上界配置为 90，则得分为 90 或 90 以上的任何记录都无需查看。

离散记录

得分小于阈值上界但大于或等于阈值下界的记录。 离散记录是指需要在 Analyst 工具中查看的异常。 例如，如果阈值下界为 40，则任何得分为 40 到 90 的记录均需要手动查看。

已拒绝记录

得分小于阈值下界的记录。 已拒绝记录无效。 默认情况下，异常转换会从数据流删除已拒绝记录。对于本例，任何得分等于或低于 40 的记录都是已拒绝记录。

注意: 如果质量问题字段为 NULL，则记录不是异常。 如果任何质量问题包含文本或者包含空字符串，则记录是异常。 如果字段没有错误，则验证质量问题端口是否包含空值。 如果质量问题端口为空白而不是空值，则异常转换会将每个记录标记为异常。 如果用户需要在 Analyst 工具中更正问题，则用户无法按照数据质量问题筛选异常。

如果记录的得分小于 0 或大于 100，则该行无效。 数据集成服务会记录一条说明行无效的错误消息，并会跳过记录而不进行处理。

如果不将记录得分作为输入连接到异常转换，则转换将包含质量问题的所有记录写入到离散记录表中。

如果在映射任务中包含离散记录异常转换，则可以在同一个工作流中配置人工任务，以包括手动查看异常。 人工任务在工作流中的映射任务结束时启动。 人工任务需要用户访问 Analyst 工具以解析质量问题。 用户可以更新数据并更改离散记录表中每个记录的质量状态。

离散记录异常管理流程

异常转换从数据质量转换接收记录得分，并创建包含不同数据质量级别的记录的表。必须配置数据质量转换以查找质量问题并为每一行提供记录得分。

可以在一个映射中配置数据质量转换，也可以针对数据质量流程的不同阶段创建映射。

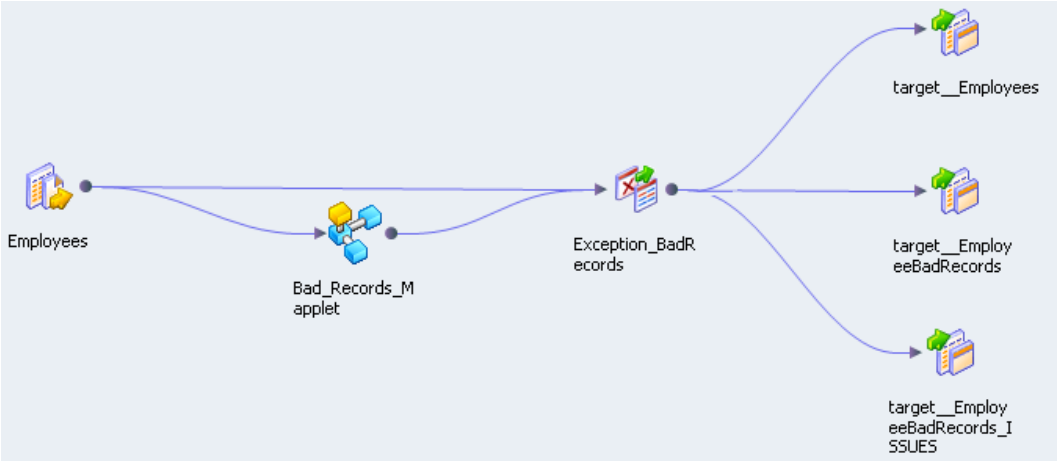
完成以下离散记录异常管理任务：

1. 在 Developer 工具中，根据定义的数据质量问题定义可生成源数据的得分值的转换。定义可返回描述源数据质量的文本的转换。例如，可以配置标签创建器转换，以对照引用表检查源数据，然后为每次比较写入描述性标签。可以在判定转换中定义 IF/THEN 规则，以检查数据字段。可以定义执行不同数据质量操作的多个转换和 Mapplet。
2. 配置异常转换以分析其从数据质量操作接收的记录得分。配置转换以基于记录中的得分值将记录写入数据库表。可以为正常记录、离散记录、质量问题和已拒绝记录创建单独的表。
3. 为可能包含错误数据的每个输入端口分配质量问题端口。
4. 或者，为正常记录和离散记录配置目标数据对象。将异常转换输出端口连接到映射中的目标数据对象。
5. 为离散记录创建目标数据对象。选择生成离散记录表并将其添加到映射中。生成离散记录表时，Developer 工具还会生成质量问题表。将质量问题表添加到映射中。
6. 将映射添加到工作流中。
7. 配置人工任务以便为用户分配离散记录的手动查看。用户可在 Analyst 工具中查看和更新离散记录。

离散记录异常映射

创建标识离散记录异常的映射后，配置映射以基于记录中的数据质量将记录写入一个或多个数据库目标。

下图显示了离散记录异常映射示例：



映射包含以下对象：

数据源

包含要进行数据质量分析的记录的 Employees 数据源。

Mapplet

Bad_Records_Mapplet 包含检查质量问题和记录得分并将其添加到源记录中的转换。规则是分析数据并查找质量问题的转换。例如，可以在引用表中加入一个用于比较输入数据的标签创建器转换。视结果而定，可以将标签创建器转换配置为将质量问题作为行中的附加列返回。可以配置一个使用 IF、THEN、ELSE 语句的判定转换，以检查数据并将质量问题和记录得分应用到输入数据。

异常转换

异常转换确定哪些记录写入到数据目标中（包括离散记录表和问题表）。

正常记录表

异常转换将所有正常质量的记录写入 target_Employees 表。

离散记录表

异常转换将所有离散质量的记录写入 target_EmployeeBadRecords 表。离散记录需要手动查看。

问题表

异常转换将质量问题写入 target_EmployeeBadRecords_ISSUES 表。在 Analyst 工具中查看离散记录时，用户界面将质量问题链接到离散记录。

或者，异常转换可以将已拒绝记录写入到已拒绝记录表中。在转换的**配置**视图中，必须选择为已拒绝记录创建单独的输出组。

离散记录异常质量问题

质量问题是文本字符串，描述可导致记录得分低的数据质量问题的类型。 离散记录异常转换接收与包含低记录得分的每个源行相关联的质量问题。 可以配置确定质量问题和记录得分的不同类型的转换。

例如，可以创建一个检查电话号码的判定转换。 判定转换生成电话号码的记录得分和质量问题。

下面的判定策略标识判定转换中长度错误的电话号码：

```
IF LENGTH(Phone_Number) > 10 THEN
  Score:=50
  Phone_Quality_Issue:='Phone num too long'
ELSEIF  LENGTH(Phone_Number) < 10 THEN
  Score:=50
  Phone_Quality_Issue:=' Phone num too short'
ELSE
  Score:=90
ENDIF
```

配置异常转换时，必须将 Phone_Quality_Issue 与 Phone_Number 端口关联。 这些端口来自不同的输入组。

异常转换读取判断转换生成的得分，并将得分为“50”的记录分配给输出端口的离散记录组。 其将 Phone_Quality_Issue 写入到输出端口的 Issues 组。

人工任务

配置包含异常转换的工作流时，在映射任务中加入一个映射。可以在同一个工作流中添加人工任务。人工任务需要一位或多位用户在 Analyst 工具中更正异常记录。

映射任务标识源数据中包含未解决的数据质量问题的记录。数据分析师可使用 Analyst 工具解决问题和更新每条记录的数据质量状态。

配置人工任务时，将创建一个或多个任务实例以及一个或多个任务步骤。任务实例表示用户必须处理的数据集。任务步骤表示用户在其任务实例中对记录执行的工作类型。可以创建多个任务实例，这样在 Analyst 工具中，不同的用户可以处理不同部分的数据。

在 Analyst 工具中，用户可以使用以下方法之一更新离散记录的状态：

- 如果记录有效，用户将更新表元数据，以确认数据库中永久性存储的记录。
- 如果记录无效，用户将更新表元数据，以便在工作流后面的阶段从数据库中删除记录。
- 如果未确认记录状态，用户将更新表元数据，以使记录返回到工作流，以便在映射任务中进一步进行处理。

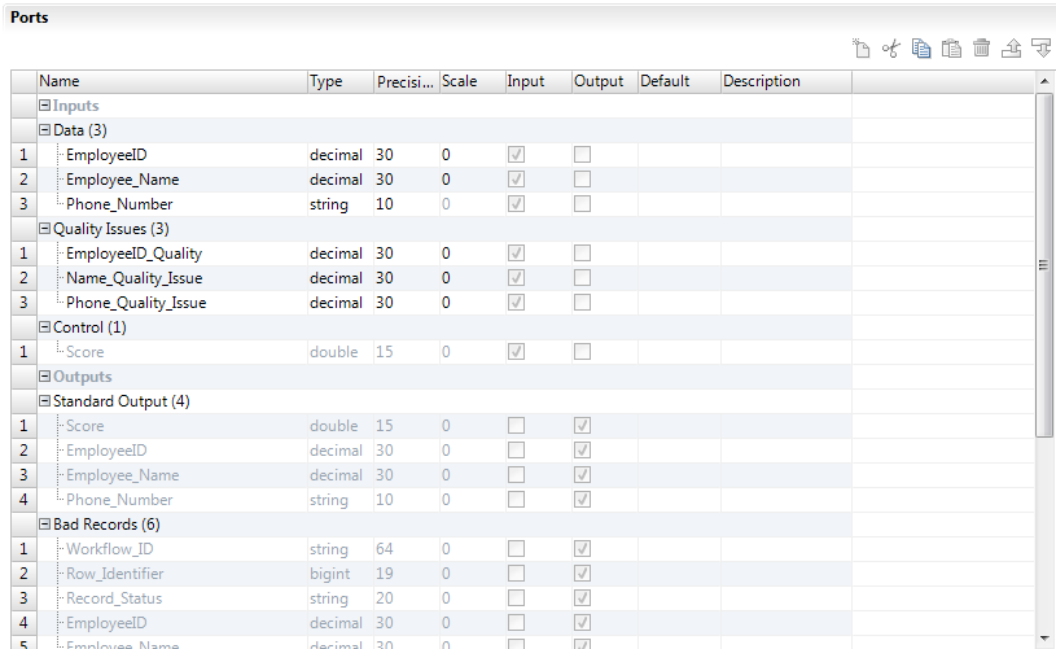
有关人工任务的详细信息，请参阅《Informatica Developer 工作流指南》。

离散记录异常端口

在离散记录异常转换的**端口**选项卡上配置输入和输出端口。

离散记录异常转换包含输入和输出端口组。

下图显示了**端口**选项卡：



离散记录异常转换输入端口

离散记录异常转换包含分别用于数据、质量问题和记录得分的输入组。

离散记录异常转换具有以下输入组：

数据

源数据字段。

质量问题

包含描述记录质量问题的端口。 质量问题端口包含字符串，例如“Excess_Characters”或“Bad_Data_Format”。 每个记录中可以有多个质量问题。 转换不会将质量问题组中的端口与源数据字段关联，直至您在**问题分配**视图中将问题分配给数据端口。

控制

记录得分。 异常转换分析记录得分，以确定输入行是否属于异常。 如果未连接得分端口，则当质量问题端口中包含数据时，异常转换会将行标识为异常。

离散记录异常转换输出

离散记录异常转换具有多个输出组。

离散记录异常转换具有以下输出组：

标准输出

无需检查数据质量问题的正常质量记录。

标准输出组中的每个记录都包含一个代表记录的数据质量的得分端口。

离散记录

需要检查数据质量问题的异常。

离散记录组中的每个记录都包含一个工作流 ID、一个行标识符和一个记录状态端口。

问题

离散记录组中记录的质量问题。质量问题指您查看离散记录时 Analyst 工具显示的元数据元素。

问题组中的每个记录都包含一个工作流 ID 和一个行标识符端口，用以标识问题所属的离散记录行。

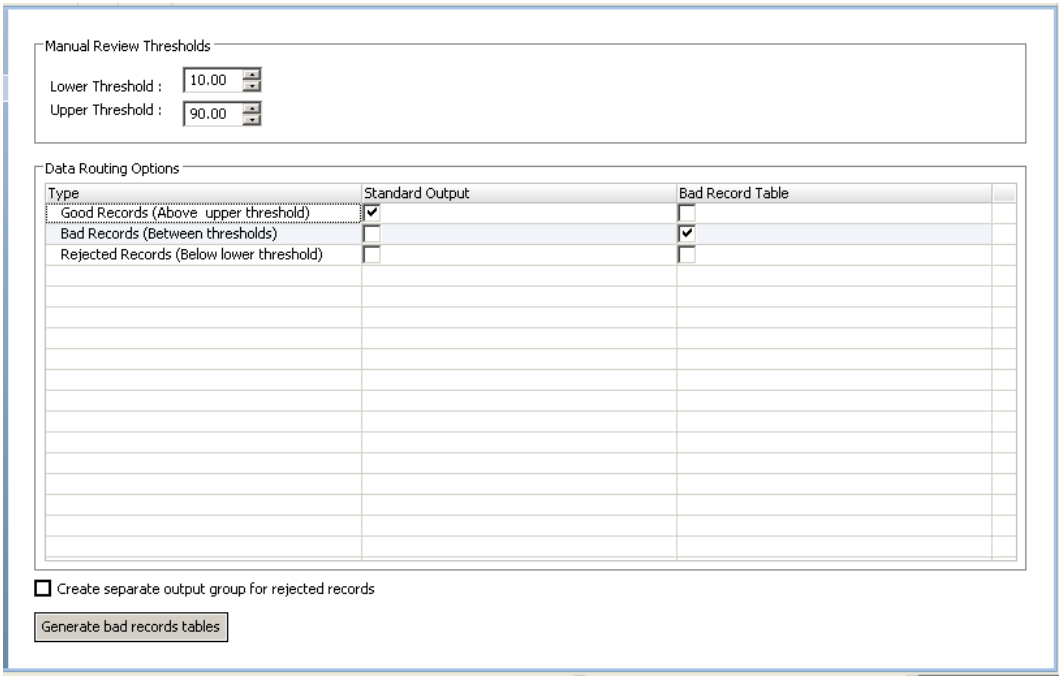
已拒绝记录

包含可从数据库中删除的记录的可选组。已拒绝组中的每个记录在得分端口中都包含一个得分的低记录。

离散记录异常配置视图

配置视图指定转换用于识别正常记录和离散记录的阈值上界和下界。**配置**视图同时还识别得分高于或低于阈值的记录的目标表。

下图显示异常转换**配置**视图：



可以在**配置**视图中配置以下属性：

阈值下界

离散记录得分范围的下限。转换将得分低于阈值下界的记录视为已拒绝记录。

阈值上界

离散记录得分范围的上限。转换将得分高于或等于阈值上界的记录视为正常记录。

数据路由选项

输出记录的类型。在默认配置中，转换会将正常记录写入标准输出，将离散记录写入离散记录表。默认情况下，转换不会将已拒绝记录写入数据库表。

标准输出

转换写入到标准输出端口中的记录类型。默认类别是正常记录。

离散记录表

转换写入离散记录输出端口的记录的类型。默认类别是离散记录。

为已拒绝记录创建单独的输出组

为已拒绝记录创建单独的输出组。默认情况下，不选中此选项。

生成离散记录表

创建数据库表以存储离散记录数据。选择该选项后，异常转换将创建数据库表，将数据对象添加到模型存储库，并将该对象的实例添加到映射画布。可以为映射中的异常转换实例生成离散记录表。在生成离散记录表的过程中，Developer tool 还会创建问题表，以存储记录的描述性元数据。

注意: Developer tool 向离散记录表中的每个列名添加 12 个字符的后缀。如果使用 Oracle 数据库，源列名不能超过 18 个字符。

生成离散记录表和问题表

向映射中添加转换时，可以生成离散记录表和问题表。Developer 工具会将这些表添加到模型存储库。

1. 单击**生成离散记录表**以生成表。

   此时将显示**创建关系数据对象**对话框。
2. 浏览数据库连接。选择与数据库的连接，以包含该表。
3. 输入离散记录表的名称。Developer 工具会将您输入的名称应用至离散记录表和问题表。

   Developer 工具会将以下字符串附加到问题表名称：

   _ISSUE

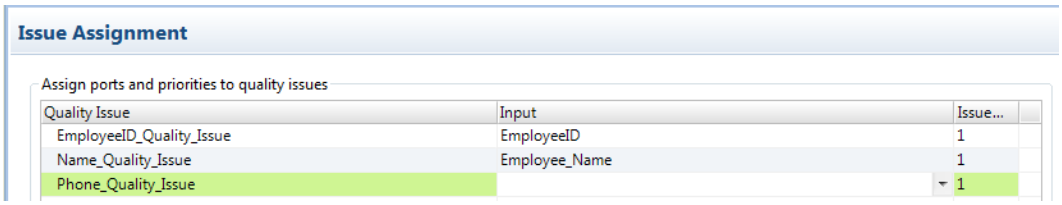
   如果连接到 Oracle 数据库，离散记录表名称不能超过 24 个字符。
4. 输入模型存储库中离散记录数据对象的名称。
5. 单击**完成**。

   Developer 工具会将表添加到映射画布和模型存储库中。

离散记录异常问题分配

必须为数据质量问题分配端口和优先级。

下图显示了**问题分配**视图：



问题分配视图包含以下字段：

质量问题

您在质量问题输入组中定义的每个质量问题端口都显示在**质量问题**列中。

输入

输入列包含您在**问题分配**视图中分配给质量问题的数据端口。将输入端口与各个质量问题端口关联。包含离散质量数据的各个输入端口必须至少有一个对应的质量问题端口，用以指出问题的类型。例如，可以为 Phone_Quality_Issue 选择 Phone_Number 端口。可以将一个端口分配给多个质量问题。

问题优先级

在将同一个输入端口分配给多个质量问题时，问题优先级确定哪个质量问题最重要。如果为一个输入字段显示多个质量问题，则数据集成服务将应用优先级最高的质量问题。如果输入端口存在多个质量问题并且这些问题的优先级相同，则数据集成服务将应用位于列表最顶端的质量问题。输入 1 到 99 之间的优先级，其中 1 代表最高优先级。

定义问题优先级，以便在 Analyst 工具中筛选记录。

将端口分配给质量问题

分配用来与各个质量问题关联的端口。Developer 工具将在问题输出组中为您在**问题分配**视图中添加的每个关联创建端口。

1. 对于每个质量问题，单击**输入**字段以显示输入端口列表。
2. 选择要与质量问题关联的输入端口。

   可以为多个问题选择同一个端口。
3. 单击**问题**列并选择质量问题的优先级。

异常转换高级属性

配置可确定数据集成服务如何处理异常转换的数据的属性。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

配置离散记录异常转换

配置离散记录异常转换时，配置输入端口和可能在每个端口中出现的质量问题。 定义用于确定数据质量的阈值上界和下界。 配置写入异常和拒绝记录的位置。

1. 创建可重用或不可重用离散记录异常转换。
   - 要创建可重用转换，请选择**文件** > **新建** > **转换**，然后选择离散记录异常转换。
   - 要创建不可重用转换，请打开映射，并将异常转换添加到映射画布。 从向导中选择离散记录异常转换。
2. 单击**下一步**，或单击**完成**。

   如果单击**下一步**，则可在创建转换之前更新默认阈值和数据路由选项。
3. 配置输入端口。

- 如果创建可重用转换，请选择**端口**选项卡，并为要连接到转换的数据添加端口。
- 如果创建不可重用转换，请将其他对象添加到映射画布，并将输入端口拖动到转换中。

4. 选择**配置**视图。
5. 配置得分阈值的上界和下界。
6. 在**数据路由选项**部分，配置标准输出和异常表属性，以设置转换写入每个记录类型的位置。

   配置写入正常记录、离散记录和已拒绝记录的位置。 可以将这些记录写入标准输出或写入离散记录表中。
7. 打开**问题分配**视图。 将数据质量问题分配给数据端口。

   将优先级分配给每个问题。 如果端口包含多个问题的值，则转换将显示优先级最高的问题。
8. 选择用以生产离散记录表的选项。 输入数据库连接和表名称信息。表必须来自于默认架构。
   - 在生成离散记录表时，为该记录生成一个表，并为与这些记录相关联的数据质量问题额外生成一个表。 转换将在模型存储库中创建数据库对象。
9. 将转换输出端口连接到一个或多个数据目标。将输出端口连接到与您在**配置**视图上设置的输出选项相对应的数据对象。
   - 如果要创建可重用转换，请将转换添加到映射并连接输出端口。
   - 如果创建不可重用转换，该转换会将端口连接到离散记录表。 将输出端口连接到任何其他数据目标。

离散记录异常映射示例

组织执行数据项目以查看新客户数据。 组织需要验证客户联系人数据是否有效。 下面的示例说明如何定义离散记录异常转换，该转换从执行客户记录的数据质量分析的 Mapplet 接收记录。

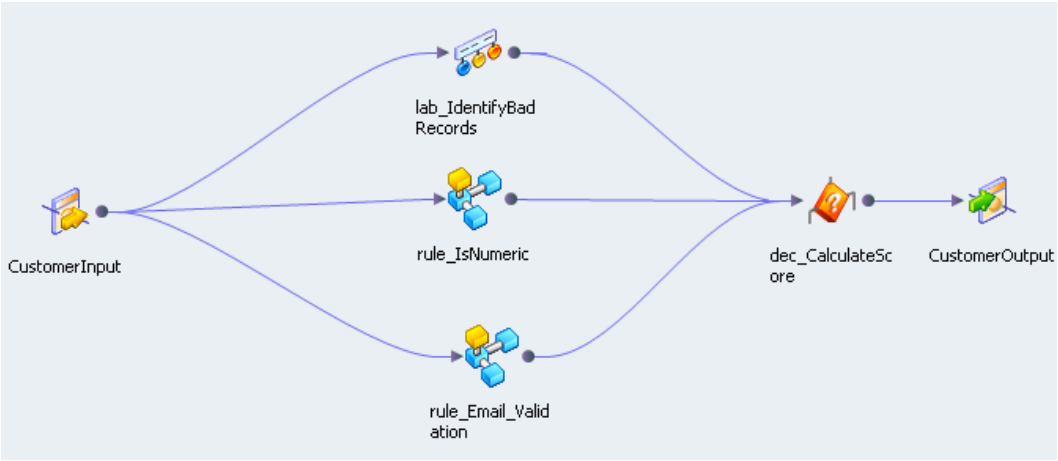
创建一个包含数据质量转换的 Mapplet，用以计算客户数据的格式和准确性。 Mapplet 包含可根据数据质量分析的结果生成记录得分的转换。 该转换还根据分析结果定义数据的质量问题。

离散记录异常 Mapplet

创建包含数据质量转换的 Mapplet，以检查特定字段的值。转换检查引用表和内容集，以确定记录中的字段是否有效。转换根据结果为每个记录应用记录得分。异常转换从 Mapplet 接收记录，并根据记录得分将每个记录路由到相应的输出。

Mapplet 由标签创建器转换、判定转换和表达式转换组成。

下图显示了 Mapplet 中的对象：



Mapplet 执行以下任务：

- 标签创建器转换验证输入端口中的区域、省/自治区/直辖市、国家/地区代码、邮政编码和收到的邮政代码数据。对于每个端口，该转换都包含一个策略。这些策略将源数据与引用表相比较，并标识无效值。
- 表达式转换 Mapplet 验证电话号码是否为数字，并验证该号码是否包含 10 位数字。
- 标签创建器转换和表达式转换 Mapplet 验证电子邮件地址是否有效。表达式转换验证电子邮件字符串的结构。标签创建器转换对照国际 IP 地址后缀的引用表检查 IP 地址。
- 判定转换从该转换和 Mapplet 接收输出。其计算客户联系人记录的整体记录得分。

创建包含 Mapplet 的离散记录异常映射。离散记录异常映射包括一个异常转换，该转换将异常写入到离散记录数据库表中。数据分析师可使用 Analyst 工具研究和更新离散记录表中的异常记录。

离散记录异常示例输入组

异常转换有三个输入组。 该转换有一个接收源数据的数据组。 有一个质量问题组，该组接收由数据质量转换发现的数据质量问题。 还有一个包含行的记录得分的控制组。

下图显示了异常转换中的输入组：

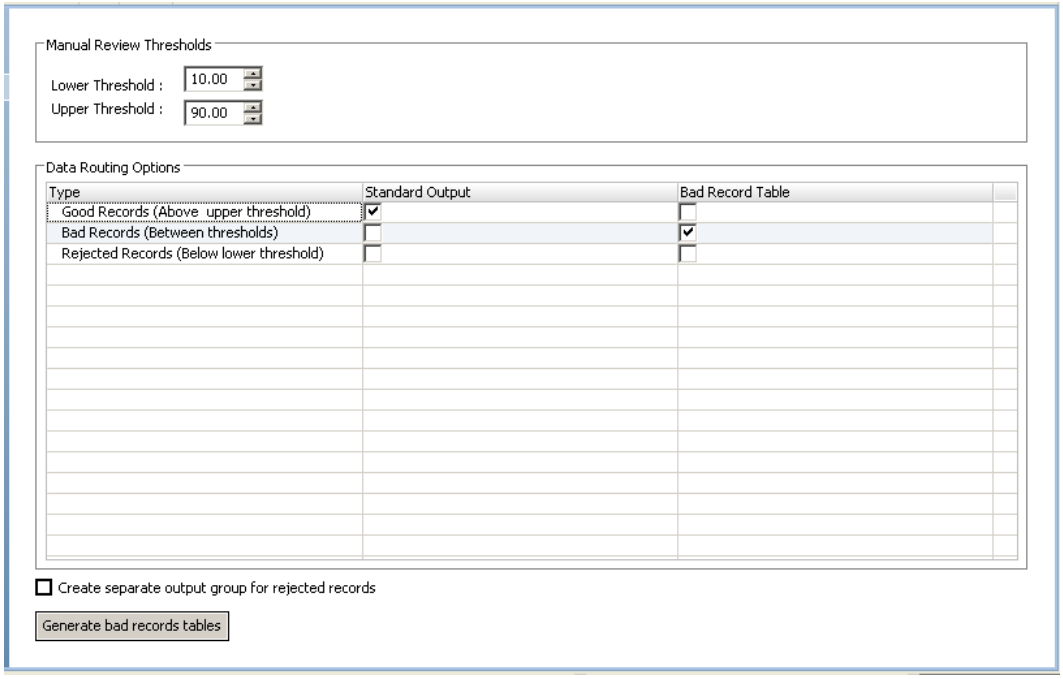
	Name	Type	Precision	Scale	Input	Output
	Inputs					
	Data (11)					
1	CUST_ID	decimal	20	0	☑	☐
2	COMPANY	string	100	0	☑	☐
3	CONTACT	string	100	0	☑	☐
4	TITLE	string	100	0	☑	☐
5	ADDR1	string	100	0	☑	☐
6	ADDR2	string	100	0	☑	☐
7	ADDR3	string	100	0	☑	☐
8	ADDR4	string	30	0	☑	☐
9	COUNTRY	string	20	0	☑	☐
10	PHONE	string	50	0	☑	☐
11	EMAIL	string	100	0	☑	☐
	Quality Issues (7)					
1	CompanyStatus	string	30	0	☑	☐
2	LocalityStatus	string	30	0	☑	☐
3	ProvinceStatus	string	30	0	☑	☐
4	CountryStatus	string	30	0	☑	☐
5	ZipStatus	string	30	0	☑	☐
6	PhoneStatus	string	30	0	☑	☐
7	EmailStatus	string	30	0	☑	☐
	Control (1)					
1	Score	double	15	0	☑	☐

离散记录异常示例配置

在**配置**视图上定义阈值上界和下界。 标识转换写入正常记录、离散记录和已拒绝记录的位置。

接受路由正常记录、离散记录和问题的默认配置。

下图显示异常转换**配置**视图：



下表介绍了配置设置：

选项	设置
阈值下界	10
阈值上界	90
正常记录	标准输出
离散记录	离散记录表
已拒绝记录	-

单击**生成离散记录**表以创建离散记录和问题表。

离散记录异常示例映射输出

向映射添加写入转换，并将标准输出端口连接到数据对象。该映射还包含您在**配置**视图上创建的离散记录数据库对象和问题数据库对象。

离散记录表

离散记录表包含记录得分介于阈值上界和下界之间的异常。

下图显示了异常转换所返回的离散记录：

Name: exc_BadRecords.Bad_Output_DI

	Workflow_ID	Row_Identifier	Record_Status	CUST_ID	COMPANY	CONTACT	TITLE	ADDR1	ADDR2
1	DummyWor...	0	INVALID	7121657	WAITROSE	MRS LACI WINI...	DIRECTOR OF IS	380-394 N END...	FULHAM
2	DummyWor...	1	INVALID	7121649	WAITROSE	MR NICHOLAS...	SENIOR PROJE...	EATON CNTR C...	EATON
3	DummyWor...	2	INVALID	7121647	VARSHEE SUPE...	MRS REUVEN C...	MANAGER	834 LONDON RD	THORNTON HE
4	DummyWor...	3	INVALID	1002138	VICTORY SUPE...	MR RAY ARIAS	COMPUTER SPE...	22 FITCHBURG...	AYER
5	DummyWor...	4	INVALID	1002137	VICTORY SUPE...	MR NEVILLE DE...	SR PROJ MANA...	21 TIMPANY BLV	GARDNER
6	DummyWor...	5	INVALID	1002109	USDA FOREST...	MR MICHEAL F...	COMPUTER SPE...	1621 N KENT S...	ROSSLYN
7	DummyWor...	6	INVALID	1002026	US ARMY CORP...	MR ROBERT EV...	PROGRAMMER...	20 MASS AVE NW	WASHINGTON
8	DummyWor...	7	INVALID	1002062	US NAVY	MR JOHN WELCH	COMPUTER SPEC	DATA PROCESS...	OAK HARBOUR
9	DummyWor...	8	INVALID	1062004	TRICOR INDUS...	MR MICHAEL G...	V.P.	8181 PROFESSI...	LANDOVER
10	DummyWor...	9	INVALID	1001921	THE CORNER S...	MR DENIS LEE	MANAGER ($$...	102 MID STR S...	REDHILL
11	DummyWor...	10	INVALID	7121217	THE CORNER S...	MRS TESSI SAN...	PROGRAMMER	192 BATTERSE...	BATTERSEA

离散记录表包括源记录中的所有字段。离散记录还包括以下字段：

Workflow_ID

包括异常转换的工作流的名称。工作流包含异常转换映射任务和用以查看问题的人工任务。如果异常转换不在工作流中，则 Workflow_ID 包含 DummyWorkflowID。

Row_Identifier

标识各个行的唯一编号。

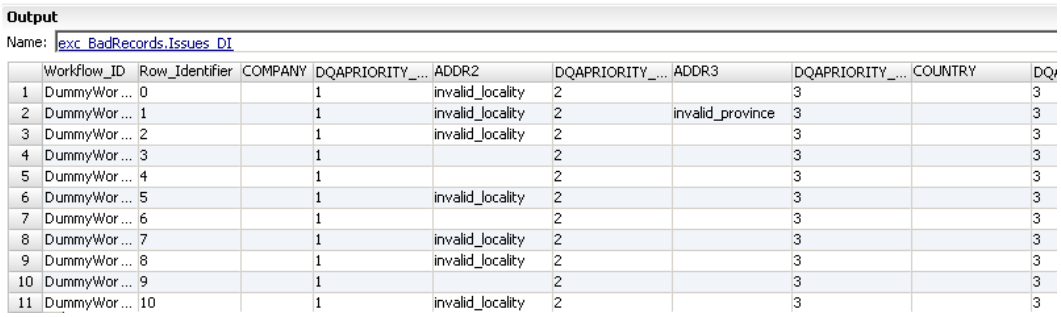
Record_Status

Analyst 工具的记录状态。离散记录表中的每个记录接收的都是无效状态。在 Analyst 工具中更新记录时可以维护记录状态。

问题表

对于离散记录表中的每一行，在问题表中都包含一个对应行。 每一行都包含数据质量分析针对源记录所发现的问题。

下图显示了问题表中的列：



Output

Name: exc_BadRecords.Issues_DI

	Workflow_ID	Row_Identifier	COMPANY	DQAPRIORITY_...	ADDR2	DQAPRIORITY_...	ADDR3	DQAPRIORITY_...	COUNTRY	DQ
1	DummyWor...	0		1	invalid_locality	2		3		3
2	DummyWor...	1		1	invalid_locality	2	invalid_province	3		3
3	DummyWor...	2		1	invalid_locality	2		3		3
4	DummyWor...	3		1		2		3		3
5	DummyWor...	4		1		2		3		3
6	DummyWor...	5		1	invalid_locality	2		3		3
7	DummyWor...	6		1		2		3		3
8	DummyWor...	7		1	invalid_locality	2		3		3
9	DummyWor...	8		1	invalid_locality	2		3		3
10	DummyWor...	9		1		2		3		3
11	DummyWor...	10		1	invalid_locality	2		3		3

问题表包含以下列：

Workflow_ID

标识创建了记录的工作流。 工作流包含异常转换映射任务和用以查看问题的人工任务。

Row_Identifier

标识数据库表中的记录行。 行标识符标识离散记录表中的哪一行与问题表中的行对应。

问题字段名称

字段名称是指可能有质量问题的字段的名称。 如果字段包含一个错误，则列值为质量问题文本。 在上图中，ADDR2 字段名称包含 invalid_locality 质量问题。

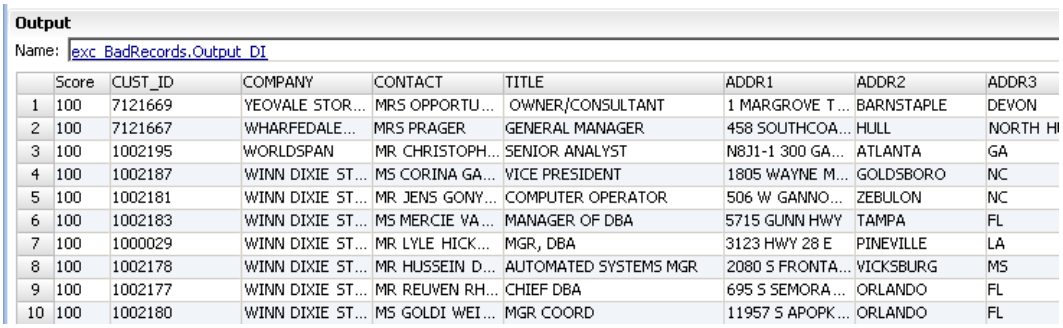
DQAPriority

问题优先级。 如果同一个字段出现多个问题，则在问题字段名称中显示优先级最高的问题。

正常记录表

正常记录表中每个记录的记录得分都大于阈值上界。 在本例中，阈值上界为 90。

下图显示了异常转换所返回的正常记录：



Output

Name: exc_BadRecords.Output_DI

	Score	CUST_ID	COMPANY	CONTACT	TITLE	ADDR1	ADDR2	ADDR3
1	100	7121669	YEOVALE STOR...	MRS OPPORTU...	OWNER/CONSULTANT	1 MARGROVE T...	BARNSTAPLE	DEVON
2	100	7121667	WHARFEDALE...	MRS PRAGER	GENERAL MANAGER	458 SOUTHCOA...	HULL	NORTH HI
3	100	1002195	WORLDSPAN	MR CHRISTOPH...	SENIOR ANALYST	N8J1-1 300 GA...	ATLANTA	GA
4	100	1002187	WINN DIXIE ST...	MS CORINA GA...	VICE PRESIDENT	1805 WAYNE M...	GOLDSBORO	NC
5	100	1002181	WINN DIXIE ST...	MR JENS GONY...	COMPUTER OPERATOR	506 W GANNO...	ZEBULON	NC
6	100	1002183	WINN DIXIE ST...	MS MERCIE VA...	MANAGER OF DBA	5715 GUNN HWY	TAMPA	FL
7	100	1000029	WINN DIXIE ST...	MR LYLE HICK...	MGR, DBA	3123 HWY 28 E	PINEVILLE	LA
8	100	1002178	WINN DIXIE ST...	MR HUSSEIN D...	AUTOMATED SYSTEMS MGR	2080 S FRONTA...	VICKSBURG	MS
9	100	1002177	WINN DIXIE ST...	MR REUVEN RH...	CHIEF DBA	695 S SEMORA...	ORLANDO	FL
10	100	1002180	WINN DIXIE ST...	MS GOLDI WEI...	MGR COORD	11957 S APOPK...	ORLANDO	FL

正常记录表的记录包含记录得分和源数据字段。

第 8 章

大小写转换器转换

本章包括以下主题：

大小写转换器转换概览

大小写转换器转换是被动转换，用于标准化输入字符串中的字母字符的大小写。

可以选择一种大小写转换格式，如大写、小写、标题式大写和句中首词大写。还可以反转输入数据中每个字符的当前大小写。

大小写转换器转换可以使用引用表的“有效”列中的值来定义输入字符的大小写。当转换在输入值和有效值之间找到匹配时，转换会将有效值的大小写应用到输入值。当大小写转换类型为**标题式大写**或**句中首词大写**时，可以使用引用表。

可以在大小写转换器转换中创建多个大小写转换策略。每个策略仅使用一种转换类型。

大小写策略属性

可以为大小写转换策略配置属性。

在**策略**视图中，可以配置以下大小写转换属性：

转换类型

定义策略使用的大小写转换方法。 可以应用以下大小写转换类型：

- **大写**。 将所有字母转换为大写。
- **句中首词大写**。 将字段数据字符串的第一个字母大写。
- **切换大小写**。 将小写字母转换为大写，将大写字母转换为小写。
- **标题式大写**。 将每个子字符串中的第一个字母大写。
- **小写**。 将所有字母转换为小写。

默认大小写转换方法为大写。

保留大写单词不变

替代大写字符串的选定大写字符。

分隔符

定义标题式大写转换的大写方式。 例如，选择破折号作为分隔符可将“smith-jones”转换为“Smith-Jones”。 默认分隔符为空格字符。

引用表

应用引用表所指定的大写格式。 仅当大小写转换选项为**标题式大写**或**句中首词大写**时才适用。单击**新建**向策略中添加引用表。

注意: 如果在标志开头引用表匹配，则标志中的下一个字符将更改为大写字符。 例如，如果输入字符串为 mcdonald，且引用表包含一个 Mc 条目，则输出字符串为 McDonald。

配置大小写转换器策略

要更改输入字符串的大小写，请在大小写转换器转换的**策略**视图中配置设置。

1. 选择**策略**视图。
2. 单击**新建**。

   此时将打开**新建策略**向导。
3. 或者，编辑策略名称和说明。
4. 单击**输入**和**输出**字段为策略选择端口。
5. 配置策略属性。 默认转换策略为**大写**。
6. 单击**下一步**。
7. 或者，添加引用表自定义与引用表条目相匹配的输入数据的大小写选项。 引用表大小写自定义仅适用于标题式大写和句中首词大写策略。
8. 单击**完成**。

大小写转换器转换高级属性

配置有助于确定数据集成服务如何为小写转换器转换处理数据的属性。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的大小写转换器转换

非本地环境中的大小写转换器转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中无限支持。在流映射中不受支持。
- Databricks Spark 引擎。无限支持。

第 9 章

分类器转换

本章包括以下主题：

分类器转换概览

分类器转换是一种被动转换，可分析输入字段并标识每个字段中的信息类型。 如果输入字段包含多个文本值，请使用分类器转换。

配置分类器转换时，请选择分类器模型和分类器算法。 分类器模型是一种引用数据对象。 分类器算法是一组规则，用于计算某个字符串中相似单词的数量以及这些单词的相对位置。 该转换会将算法分析结果与分类器模型的内容进行比较。 该转换将返回标识字符串中主要信息类型的模型分类。

分类器转换可以分析很长的字符串。 例如，可以使用转换来对电子邮件、社交媒体消息和文档文本的内容进行分类。 可以将每个文档或消息的内容传递到数据源列的某个字段中，然后将该列连接到分类器转换。 在每种情况下，您都可以对数据源进行合理准备，以使每个字段都包含要分析的完整文档或字符串内容。

分类器模型

分类器转换会使用名为“分类器模型”的引用数据对象来分析输入数据。 在配置分类器转换时，可以选择分类器模型。 转换会将输入数据列与分类器模型数据相比较，并返回描述了每个输入字段中的信息类型的标签。

分类器模型包含引用数据行和标签值。行代表可能连接到分类器转换的端口上的输入数据。标签值描述数据行包含的信息类型。配置分类器模型时，应为模型中的每个引用数据行分配标签。

要将引用数据行链接到分类器模型中的标签，您可以对该模型进行编译。编译过程会在数据行与标签值之间生成一系列逻辑关联。当您运行读取模型的映射时，数据集成服务会将模型逻辑应用到分类器转换输入数据。数据集成服务将返回对每个输入数据字段中的信息描述最准确的标签。

请在 Developer tool 中创建分类器模型。模型存储库存储分类器模型对象。Developer tool 将数据行、标签和编译数据写入 Informatica 目录结构中的一个文件。

分类器算法

向转换策略添加分类器模型时，请同时选择一个分类器算法。 此算法用于确定该转换将分类器模型数据与输入数据进行比较的方法。

可以选择 **Naive Bayes** 算法或**最大熵**算法。

选择算法时，请考虑以下因素：

- “最大熵”算法比“Naive Bayes”算法执行的分析更全面。
- 对于相同数据，使用“Naive Bayes”算法的映射比使用“最大熵”算法的映射运行速度更快。
- 对于 Informatica 核心加速器中的分类器模型，请选择“最大熵”算法。

分类器转换选项

分类器转换会在 Developer 工具的一系列选项卡或视图上显示可配置的选项。

打开可重用转换时，这些选项会显示在转换编辑器的一系列选项卡上。 打开映射中的不可重用转换时，这些选项会显示在映射编辑器的一系列视图上。 选择映射属性选项卡可在不可重用转换上查看这些视图。

可以选择以下视图：

常规

查看和更新转换名称和说明。

端口

查看转换上的输入和输出端口。

注意: 在可重用分类器转换中，“常规”和“端口”视图合并在**概览**选项卡中。

策略

添加、删除或编辑策略。

相关性

查看每个策略上的输入和输出端口。

高级

设置该转换向日志文件写入的详细级别。

分类器策略

一个策略就是一组数据分析操作，转换可对输入数据执行这些操作。 请在分类器转换中至少创建一个策略。 分类器策略可读取单个输入端口。

可在一个策略中定义一个或多个操作。 分类器操作可标识要应用于输入端口数据的分类器模型和分类器算法。 每个操作都会写入一个不同的输出端口。 如果要以不同的方式分析输入端口，请在策略中创建多个操作。

注意: 如果要识别源数据中使用的语言，请在分类器操作中选择“最大熵”算法。

分类器转换高级属性

配置属性以帮助确定数据集成服务如何处理分类器转换的数据。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

配置分类器策略

可以配置一个策略来标识数据中的信息类型。 每个策略都会分析一个输入端口。

在不可重用转换中，请先将输入端口连接到转换，然后再配置策略。

1. 打开转换，然后选择**策略**视图。
2. 单击**新建策略**。

   此时将打开策略创建向导。
3. 输入策略的名称和可选说明。
4. 从“输入”字段中选择一个输入端口。
5. 验证输入端口精度值是否足够高，以便可以读取输入端口上的所有字段。 达到该精度限制后，端口会截断输入数据。
6. 选择或清除该选项可将得分数据添加到策略输出中。
7. 单击**下一步**。
8. 确认分类器操作的类型，然后单击**下一步**。
9. 选择分类器算法。 可以选择以下算法：
   - Naive Bayes
   - 最大熵

   注意: 要识别源数据中使用的语言，请选择“最大熵”算法。
10. 验证输出端口。

    转换将为策略中的每个操作创建一个输出端口。 您可以编辑端口名称和精度。
11. 选择分类器模型。

该向导将列出模型存储库中的分类器模型对象。

12. 单击**下一步**向策略中添加其他操作。 否则，单击**完成**。

分类器分析示例

您是某软件公司的数据管理者，该公司发布了一款全新的智能手机应用程序。 该公司希望了解公众对该应用程序的反馈以及所收到的媒体报道。 公司让您和您的团队分析有关该应用程序的社交媒体评论。

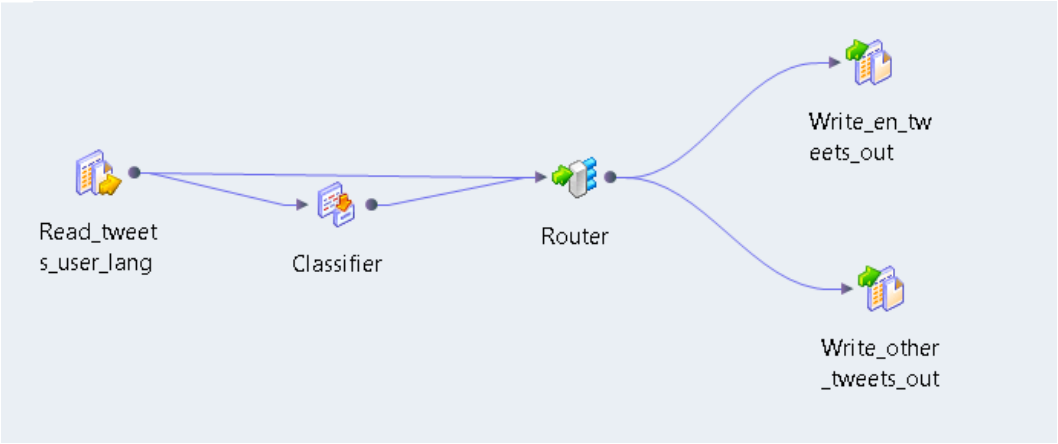
您决定从讨论智能手机的 Twitter 订阅源中捕获数据。 您使用 Twitter 应用程序编程接口来筛选 Twitter 数据流， 并创建包含要分析的 Twitter 数据的数据源。

由于 Twitter 订阅源包含多种语言的消息，因此必须识别每条消息中所使用的语言。 您决定使用分类器转换来分析语言。 您可以创建一个映射来识别源数据中的语言，然后将 Twitter 消息写入英语和非英语数据目标。

创建分类器映射

您可以创建一个映射来读取数据源，并对数据中的语言进行分类，然后根据数据中包含的语言将数据写入目标。

下图显示了 Developer 工具中的映射：



您创建的映射包含以下对象：

对象名称	说明
Read_tweet_user_lang	数据源。 包含 Twitter 消息
分类器	分类器转换。 识别 Twitter 消息中所使用的语言。
路由器	路由器转换。 根据 Twitter 消息中包含的语言将这些消息路由到数据目标对象。

对象名称	说明
Write_en_tweets_out	数据目标。 包含英语 Twitter 消息。
Write_other_tweets_out	数据目标。 包含非英语 Twitter 消息。

输入数据示例

以下数据段显示了映射中所分析的 Twitter 数据示例：

```
Twitter Message

RT @GanaphoneS3: Faltan 10 minutos para la gran rifa de un iPhone 5...
RT @Clarified: How to Downgrade Your iPhone 4 From iOS 6.x to iOS 5.x (Mac)...
RT @jerseyjazz: The razor was the iPhone of the early 2000s
RT @KrissiDevine: Apple Pie that I made for Thanksgiving. http://t.com/s9ImzFxO
RT @sophieHz: Dan yang punya 2 kupon undian. Masuk dalam kotak undian yang berhadiah Samsung
RT @IsabelFreitas: o galaxy tem isso isso isso e a bateria Ã melhor que do iPhone
RT @PremiusIpad: Faltan 15 minutos para la gran rifa de un iPhone 5...
RT @payyton3: I want apple cider
RT @wiesteronder: Retweet als je iets van Apple, Nike, Adidas of microsoft hebt!
```

数据源配置

数据源包含一个端口。 端口上的每一行都包含一条 Twitter 消息。

下表介绍了数据源的配置：

端口名称	端口类型	精度
文本	n/a	200

分类器转换配置

分类器转换使用一个输入端口和输出端口。 转换输入端口会从数据源中读取文本字段。 输出端口包含针对文本字段中的每条 Twitter 消息所识别的语言。 分类器转换使用 ISO 国家/地区代码来识别语言。

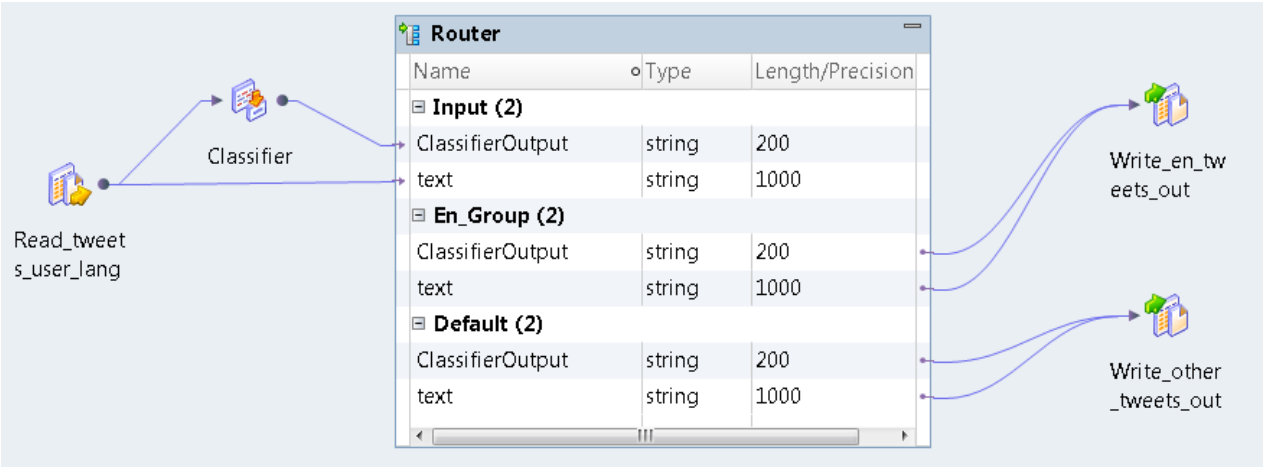
下表介绍了分类器转换的配置：

端口名称	端口类型	精度	策略
text_input	输入	200	Classifier1
Classifier_Output	输出	2	Classifier1

路由器转换配置

路由器转换使用两个输入端口。 它会从数据源中读取 Twitter 消息，从分类器转换中读取 ISO 国家/地区代码。 路由器转换会根据指定的条件将输入端口上的数据路由到不同的输出端口。

下图显示了路由器转换端口组和端口连接：



下表介绍了路由器转换的配置：

端口名称	端口类型	端口组	精度
Classifier_Output	输入	输入	2
文本	输入	输入	200
Classifier_Output	输入	默认值	2
文本	输入	默认值	200
Classifier_Output	输入	En_Group	2
文本	输入	En_Group	200

您可以配置该转换，以便为英语消息和其他语言的消息创建数据流。 要创建数据流，请向转换中添加输出端口组。 在转换上使用**组**选项来添加端口组。

要确定转换如何将数据路由到每个数据流，可在端口组上定义一个条件。 此条件可标识一个端口并在该端口上指定一个可能的值。 如果转换找到与该条件匹配的输入端口值，则它会将该输入数据路由到应用了该条件的端口组。

在 En_Group 上定义以下条件：

```
ClassifierOutput='en'
```

注意: 路由器转换会从映射中的两个对象读取数据。 该转换可以将该数据合并到每个输出组中，因为它不会更改数据对象中定义的行顺序。

数据目标配置

映射包含一个英语 Twitter 消息目标和一个其他语言消息目标。 您可以将路由器转换输出组中的端口连接到一个数据目标。

下表介绍了数据目标的配置：

端口名称	端口类型	精度
文本	n/a	200
Classifier_Output	n/a	2

分类器映射结果

运行映射时，分类器转换将标识每个 Twitter 消息的语言。 路由器转换基于语言分类将消息文本写入数据目标。

以下数据段显示了一个英语目标数据示例：

```
ISO Country Code   Twitter Message

en                 RT @Clarified: How to Downgrade Your iPhone 4 From iOS 6.x to iOS 5.x (Mac)...

en                 RT @jerseyjazz: The razor was the iPhone of the early 2000s

en                 RT @KrissiDevine: Apple Pie that I made for Thanksgiving. http://t.com/s9ImzFxO

en                 RT @payyton3: I want apple cider
```

以下数据段显示了标识为其他语言的目标数据示例：

```
ISO Country CodeTwitter Message

es              RT @GanaphoneS3: Faltan 10 minutos para la gran rifa de un iPhone 5...

id              RT @sophieHz: Dan yang punya 2 kupon undian. Masuk dalam kotak undian yang berhadiah Samsung
                Champ.

pt              RT @IsabelFreitas: o galaxy tem isso isso isso e a bateria Ã melhor que do iPhone

es              RT @PremiusIpad: Faltan 15 minutos para la gran rifa de un iPhone 5...

nl              RT @wiesteronder: Retweet als je iets van Apple, Nike, Adidas of microsoft hebt! http://t.co/
                Je6TsOOH
```

非本地环境中的分类器转换

非本地环境中的分类器转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。

- Spark 引擎。在批处理映射和流映射中无限支持。
- Databricks Spark 引擎。无限支持。

第 10 章

比较转换

本章包括以下主题：

比较转换概览

比较转换属于被动转换，用于评估成对输入字符串之间的相似性，以及计算每一对的相似度并用数字得分表示。

配置转换时，可以选择一对输入列，并为其分配匹配策略。

比较转换将输出介于 0 到 1 之间的匹配得分，其中 1 指示完全匹配。

注意: 比较转换中可用的策略在匹配转换中同样可用。 使用比较转换可以定义要添加到匹配 Mapplet 中的匹配比较操作。 您可以将多个比较转换添加到 Mapplet 中。 使用匹配转换可以在一个转换中定义一个多个匹配比较。 您可以在匹配转换中嵌入匹配 Mapplet。

字段匹配策略

比较转换包括一些预定义的字段匹配策略，用于比较成对输入数据字段。

二元语法

使用“二元语法”算法可比较长文本字符串，例如，在一个字段中输入的邮政地址。

“二元语法”算法将根据两个数据字符串中连续字符的出现情况计算两个字符串的匹配得分。 该算法查找这两个字符串共有的连续字符对。 将在这两个字符串中均匹配的字符对的数量除以字符对的总数。

“二元语法”示例

请考虑以下字符串：

- larder
- lerder

这些字符串将产生以下“二元语法”组：

```
l a, a r, r d, d e, e r
l e, e r, r d, d e, e r
```

请注意，字符串“lerder”中第二次出现的字符串“e r”不匹配，因为字符串“larder”中没有第二次出现的对应“e r”。

要计算“二元语法”匹配得分，转换会将匹配对数 (6) 除以两个字符串中的总对数 (10)。 在本示例中，字符串的相似度为 60%，匹配得分为 0.60。

汉明距离

如果数据字符的位置非常关键，可使用“汉明距离”算法，例如，在电话号码、邮政编码或产品代码等数值字段或代码字段中。

“汉明距离”算法通过计算两个数据字符串中字符不同的位置数来计算这两个数据字符串的匹配得分。 对于长度不同的字符串，最长字符串中每个额外字符都将视为两个字符串之间的一个差异。

“汉明距离”示例

请考虑以下字符串：

- Morlow
- **Ma**rlow**es**

突出显示的字符指示汉明算法确定为差异的位置。

要计算汉明匹配得分，转换会将匹配字符数 (5) 除以最长字符串的长度 (8)。 在本示例中，字符串的相似度为 62.5%，匹配得分为 0.625。

编辑距离

使用“编辑距离”算法可比较单词或短文本字符串，例如名称。

“编辑距离”算法通过插入、删除或替换字符来计算将一个字符串转换为另一个字符串的最低“成本”。

“编辑距离”示例

请考虑以下字符串：

- Levenston
- Levens**hte**i**n**

突出显示的字符指示将一个字符串转换为另一个字符串所需的操作。

“编辑距离”算法会将未更改的字符数 (8) 除以最长字符串的长度 (11)。 在本示例中，字符串的相似度为 72.7%，匹配得分为 0.727。

Jaro 距离

如果字符串中开头字符的相似性需要优先考虑，可使用“Jaro 距离”算法比较两个字符串。

Jaro 距离匹配得分反映这两个字符串中的前四个字符之间的相似度以及已标识字符转换的数量。 转换通过使用在罚值属性中输入的值来权衡前四个字符之间的匹配重要性。

“Jaro 距离”属性

配置“Jaro 距离”算法时，可以配置以下属性：

罚值

在两个比较字符串中的前四个字符不同时确定匹配得分罚值。 如果首字符不匹配，则转换将减去整个罚值。 转换将根据其他不匹配字符的位置减去相应罚值分数。 默认罚值为 0.20。

区分大小写

确定 Jaro 距离算法在比较字符时是否考虑字符的大小写。

“Jaro 距离”示例

请考虑以下字符串：

- 391859
- 813995

如果使用默认罚值 0.20 来分析这些字符串，“Jaro 距离”算法将返回匹配得分 0.513。 该匹配得分指示字符串相似度为 51.3%。

反向汉明距离

使用反向汉明距离算法可计算两个字符串之间字符位置不同的百分比（从右往左读取）。

“汉明距离”算法通过计算两个数据字符串中字符不同的位置数来计算这两个数据字符串的匹配得分。 对于不同长度的字符串，该算法会将最长字符串中的每个附加字符计算为字符串之间的一个差异。

反向汉明距离示例

请考虑使用以下字符串，采用了从右往左的对齐，以模仿反向海明算法：

- 1-999-9999
- **011-0**1-999-999**1**

突出显示的字符指示反向汉明距离算法识别为有差异的位置。

要计算反向海明匹配得分，该转换会将匹配字符数 (9) 除以最长字符串的长度 (15)。 在本示例中，匹配得分为 0.6，指示字符串的相似度为 60%。

标识匹配策略

比较转换包括一些预定义的标识匹配策略，可用于查找个人、地址或法人实体的匹配项。

下表介绍了每个标识匹配策略执行的匹配操作：

标识匹配策略	匹配操作
地址	标识地址匹配项。
联系人	标识某个位置的某个组织中的联系人。
法人实体	按法人名称标识组织。
分支机构	标识某个地址的组织。
家庭	按姓氏及地址或电话号码标识家庭。
字段	标识所选的自定义字段。
住户	标识同一居住地的同一家庭的成员。
个人	按姓名及身份证号或出生日期标识个人。
组织	按名称标识组织。
人名	按姓名标识个人。
居民	标识某个地址的个人。
广义联系人	标识某个组织中的联系人（无论在什么位置）。
广义住户	标识同一家庭的成员（无论在什么位置）。

注意: 标识匹配策略读取名为 **populations** 的引用数据文件。 有关系统中安装的人口数据文件的信息，请联系 Informatica Administrator 用户。

配置比较策略

要配置比较策略，请在比较转换的**策略**视图中编辑设置。

1. 选择**策略**视图。
2. 从**策略**部分选择比较策略。
3. 在**字段**部分，双击**可用字段**列中的单元格以选择输入。

   注意: 必须为在**输入字段**列中显示粗体输入名称的每一行选择一个输入。

比较转换高级属性

配置属性以帮助确定数据集成服务如何处理比较转换的数据。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的比较转换

非本地环境中的比较转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中无限支持。在流映射中不受支持。
- Databricks Spark 引擎。无限支持。

第 11 章

合并转换

本章包括以下主题：

合并转换概览

合并转换是一种主动转换，可用于分析多个相关记录的组并为每个组创建一个合并记录。 使用合并转换来合并由转换（如键生成器转换、匹配转换和关联转换）生成的记录组。

合并转换通过将策略应用到相关记录组来生成合并记录。 此转换包含一个用于指示哪个记录为合并记录的输出端口。 可选择将转换输出限制为仅包含合并记录。

例如，可以合并由匹配转换生成的重复员工记录组。 合并转换可创建合并记录，其中包含来自组中所有记录的合并数据。

根据您的合并要求，可以将合并转换配置为使用不同类型的策略。 使用简单策略可创建来自多个记录的合并记录。 使用简单策略时，请为每个端口指定一个策略。 使用基于行的策略可分析记录组中的行，并使用其中某一行的值来创建合并记录。 通过应用您创建的表达式，使用高级策略来创建合并记录。

合并映射

要合并记录，请创建一个可创建相关记录组的映射。将合并转换添加到映射中，并将该转换配置为将每个记录组合并到单个主记录。

根据业务目标和数据要求，将合并转换连接到其他转换。要合并匹配的记录，可以将合并转换连接到匹配转换。要将记录合并到异常记录管理中，请将该合并转换连接到某个异常转换。如果使用键生成器转换对记录进行分组，可以将某个合并转换直接连接到该键生成器转换。合并转换会为由键生成器转换创建的每个组创建一条合并的记录。

本地和 Hadoop 环境中的映射输出

当您分别在本地环境和 Hadoop 环境中运行合并映射时，合并转换可能会产生不同的结果。因为映射运行于 Hadoop 中的多个节点上，所以输入记录可能会以与本地环境中不同的顺序输入合并转换。其结果是，对于同一个输入数据集，转换可能会在每个环境中生成不同的残存记录集。对于每种情况下的输入行顺序，转换计算和合并后的结果都是准确的。

要在本地和 Hadoop 环境中生成相同的残存记录，请将合并转换配置为按以下顺序排列记录：

- 首先，在“分组依据”端口上对记录进行排序。
- 然后，按照输入端口在转换中的出现顺序对记录进行排序。

合并转换端口

Developer 工具会为添加的每个输入端口创建一个输出端口。 您不能将输出端口手动添加到此转换。 合并转换还包含一个可指示合并记录的 **IsSurvivor** 输出端口。

您添加到合并转换的其中一个输入端口必须包含组键。 合并转换需要组键信息，因为合并策略处理的是记录组，而非整个数据集。

添加输入端口时，Developer 工具通过为输入端口名称添加后缀“1”来创建输出端口名称。 该转换还包含一个可指示某记录是否为合并记录的 **IsSurvivor** 输出端口。 对于合并记录，合并转换会将字符“Y”写入 **IsSurvivor** 端口。 对于输入记录，合并转换会将字符“N”写入 **IsSurvivor** 端口。

合并转换视图

合并转换包含端口、策略和高级属性的视图。

合并转换策略视图

策略视图包含简单策略、基于行的策略和高级策略的属性。

以下列表介绍了合并策略的类型：

简单策略

简单策略可分析记录组中端口的所有值，然后选择一个值。 为每个端口指定一个简单策略。 合并转换将使用按所有简单策略选择的端口值来创建合并记录。 简单策略的示例包括端口中最常使用的值、端口中最长的值，或端口中最常使用的非空值。

基于行的策略

基于行的策略可分析记录组中的行，然后选择一行。 合并转换将使用该行中的端口值来创建合并记录。 基于行的策略的示例包括最多字符数、最少空白字段数或最常使用的字段的最大计数。

高级策略

高级策略可使用您定义的策略来分析记录组。 在表达式中使用合并函数可构建高级策略。 合并转换将基于表达式的输出来创建合并记录。 创建的表达式也可以使用判定转换中的所有可用函数。

合并转换高级属性

合并转换包含确定排序行为、输出模式、高速缓存行为以及跟踪级别的高级属性。

您可以配置以下高级属性：

排序

确定转换是否根据**分组依据**端口数据对输入行排序。默认情况下启用此属性。

如果输入行未预先排序，请选择此属性。

排序(区分大小写)

确定排序操作是否区分大小写。默认情况下启用此属性。

输出模式

确定转换是将全部记录作为输出写入，还是将合并的记录作为输出写入。默认值为“全部”。

缓存文件目录

指定数据集成服务将当前转换的临时数据写入到的目录。当输入数据量超出可用系统内存时，数据集成服务会将临时文件写入该目录。数据集成服务会在运行映射后删除这些临时文件。

您可以在该属性上输入目录路径，也可以使用参数标识目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。默认值为 CacheDir 系统参数。

缓存文件大小

确定数据集成服务对转换的输入数据排序使用的系统内存量。

在对数据排序之前，数据集成服务会分配您指定的内存量。如果排序操作生成的数据较多，数据集成服务会将超出的数据写入缓存文件目录。如果排序操作所需的内存超出系统内存和文件存储可以提供的内存，映射会失败。

转换以字节为单位读取值。默认值为 400,000 字节。最大值为 2,147,483,647 字节。可以使用参数指定缓存文件大小。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

缓存文件大小

缓存文件大小属性确定数据集成服务分配给合并转换进行排序操作的系统内存量。为此属性配置一个小于或等于数据集成服务主机上的 RAM 大小的值。

为获得最佳性能，请至少指定 16 MB 的缓存文件大小。在**高级属性**视图上，设置缓存文件大小值（以字节为单位）。

在执行排序操作之前，数据集成服务会分配缓存文件大小属性指定的内存。数据集成服务会先将所有输入数据传递至合并转换，然后再执行排序操作。

如果输入数据量超出缓存文件大小，数据集成服务会将数据写入缓存文件目录。将数据写入缓存文件目录时，数据集成服务占用的磁盘空间至少为输入数据量的两倍。

使用以下公式确定传入数据的大小：

```
[number_of_input_rows *( Sum (column_size) + 16]
```

下表列出了在缓存文件数据计算中应用的数据类型和列大小值。

数据类型	列大小
二进制	精度 + 8。 舍入到最接近的 8 的倍数。
日期/时间	29
十进制，高精度关闭（所有精度）	16
十进制，高精度打开（精度 <=18）	24
十进制，高精度打开（精度 >18，<=28）	32
十进制，高精度打开（精度 >28）	16
十进制，高精度打开（负小数位数）	16
双精度型	16
实型	16
整型	16
字符串、文本	Unicode 模式：2*(精度 + 5) ASCII 模式：精度 + 9

简单策略

简单策略会分析记录组中的端口，然后返回一个值。 为每个端口指定一个简单策略。 合并转换将使用按所有简单策略选择的端口值来创建合并记录。

在转换的**策略**视图中配置策略时，策略将显示以下文本作为合并方法：

使用默认值。

默认策略为“最高行 ID”。

您可以从以下简单策略中进行选择：

平均值

分析记录组中的端口，然后返回所有值的平均值。

对于 String 和 Date/time 数据类型，策略将返回最常出现的值。

最长值

分析记录组中的端口，然后返回具有最多字符数的值。 如果两个或多个值的最多字符数相同，则策略将返回第一个满足条件的值。

最大值

分析记录组中的端口，然后返回最高值。

对于 String 数据类型，策略将返回最长字符串。 对于 Date/time 数据类型，策略将返回最近的日期。

最小值

分析记录组中的端口，然后返回最低值。

对于 String 数据类型，策略将返回最短字符串。 对于 Date/time 数据类型，策略将返回最早日期。

最常出现的值

分析记录组中的端口，然后返回最常出现的值，包括空值。 如果两个或多个值出现的最高次数相同，则策略将返回第一个满足条件的值。

最常出现的非空值

分析记录组中的端口，然后返回最常出现的值，空值除外。 如果两个或多个值出现非空值的最高次数相同，则策略将返回第一个满足条件的值。

最短值

分析记录组中的端口，然后返回具有最少字符数的值。 如果两个或多个值的最少字符数相同，则策略将返回第一个满足条件的值。

最高行 ID

分析记录组中的端口，然后返回具有最高行 ID 的值。

基于行的策略

基于行的策略可分析记录组中的行，然后选择一行。 合并转换将使用该行中的端口值来创建合并记录。 默认策略为“最多数据”。

请选择以下基于行的策略之一：

最多数据

选择具有最多字符数的行。 如果两行或多个行的最多字符数相同，则策略将返回最后一个满足条件的值。

最多填充

选择具有最多非空列数的行。 如果两行或多个行的最多非空列数相同，则策略将返回最后一个满足条件的值。

精确模型

选择具有最常使用的非空值最大计数的行。 例如，请考虑具有三个端口的行，其中包含记录组中最常使用的值。 该行最常使用的值的计数为“3”。

如果两行或多个行的最常使用的非空值最大计数相同，则策略将返回最后一个满足条件的值。

基于行的策略示例

下表显示了示例记录组。 最后一列介绍了基于行的特定策略在该记录组中选择不同行的原因。

产品 ID	名	姓	邮政编码	策略选择
2106	Bartholomew		28516	“最多数据”策略选择此行的原因是该行包含的字符比其他行多。
2236	Bart	Smith	28579	“最多填充”策略选择此行的原因是该行包含的非空列比其他行多。
2236	<Blank>	Smith	28516	“精确模型”策略选择此行的原因是该行包含最常使用的值的最大计数。

高级策略

可以使用高级策略通过预定义的函数创建合并策略。 可以使用合并函数和其他 Informatica 函数。

可以创建包含简单合并函数或基于行的合并函数的表达式。 使用简单合并函数基于记录组中的端口值构建合并记录。 使用基于行的合并函数从记录组中选择行。

在合并转换中，合并表达式必须填充所有输出端口。 如果合并表达式未使用所有输出端口，则转换将导致映射失败。

可以将简单策略或基于行的策略用作高级策略的模板。 配置简单策略或基于行的策略，然后选择“高级”。 合并转换将使用执行此策略的函数生成一个表达式。 可以添加多个函数来实施其他要求。

简单合并函数

使用简单合并函数可以在记录组的所有端口值中选择一个值。 使用简单合并函数时，需要向其提供端口和分组依据端口。

CONSOL_AVG

分析记录组中的端口，然后返回所有值的平均值。

语法

```
CONSOL_AVG(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

某个端口中所有值的平均值。

对于 String 和 Date/time 数据类型，函数将返回最常出现的值。

示例

以下表达式使用 CONSOL_AVG 函数查找 SalesTotal 输入端口的平均值：

```
SalesTotal1:= CONSOL_AVG(SalesTotal, GroupKey)
```

在此表达式中，CONSOL_AVG 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数分析 SalesTotal 端口并返回平均值。 该表达式会将平均值写入 SalesTotal1 输出端口。

CONSOL_LONGEST

分析记录组中的端口，然后返回具有最多字符数的值。

语法

```
CONSOL_LONGEST(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

具有最多字符数的端口值。

如果两个或多个值的最多字符数相同，则策略将返回第一个满足条件的值。

示例

以下表达式将使用 CONSOL_LONGEST 函数分析 FirstName 输入端口，并查找具有最多字符数的值：

```
FirstName1:= CONSOL_LONGEST(FirstName, GroupKey)
```

在此表达式中，CONSOL_LONGEST 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数会对 FirstName 端口进行分析并返回最长值。 表达式会将此值写入 FirstName1 输出端口。

CONSOL_MAX

分析记录组中的端口，然后返回最高值。

语法

```
CONSOL_MAX(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

最高端口值。

对于 String 数据类型，函数将返回最长字符串。 对于 Date/time 数据类型，函数将返回最近的日期。

示例

以下表达式使用 CONSOL_MAX 函数分析 SalesTotal 输入端口并查找最高值：

```
SalesTotal1:= CONSOL_MAX(SalesTotal, GroupKey)
```

在此表达式中，CONSOL_MAX 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数分析 SalesTotal 端口并返回最高值。 表达式会将此值写入 SalesTotal1 输出端口。

CONSOL_MIN

分析记录组中的端口，然后返回最低值。

语法

```
CONSOL_MIN(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

最低端口值。

对于 String 数据类型，函数将返回最短字符串。 对于 Date/time 数据类型，函数将返回最早日期。

示例

以下表达式将使用 CONSOL_MIN 函数分析 SalesTotal 输入端口并查找最低值：

```
SalesTotal1:= CONSOL_MIN(SalesTotal, GroupKey)
```

在此表达式中，CONSOL_MIN 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数会对 SalesTotal 端口进行分析并返回最低值。 表达式会将此值写入 SalesTotal1 输出端口。

CONSOL_MOSTFREQ

分析记录组中的端口，然后返回最常出现的值，包括空值。

语法

```
CONSOL_MOSTFREQ(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

最常出现的值，包括空值。

如果两个或多个值出现的最高次数相同，则策略将返回第一个满足条件的值。

示例

以下表达式使用 CONSOL_MOSTFREQ 函数分析 Company 输入端口并查找最常出现的值：

```
Company1:= CONSOL_MOSTFREQ(Company, GroupKey)
```

在此表达式中，CONSOL_MOSTFREQ 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数会对 Company 端口进行分析并返回最常出现的值。 表达式会将此值写入 Company1 输出端口。

CONSOL_MOSTFREQ_NB

分析记录组中的端口，然后返回最常出现的值，空值除外。

语法

```
CONSOL_MOSTFREQ_NB(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

最常出现的值，不包括空值。

如果两个或多个值出现的最高次数相同，则策略将返回第一个满足条件的值。

示例

以下表达式使用 CONSOL_MOSTFREQ_NB 函数分析 Company 输入端口并查找最常出现的值：

```
Company1:= CONSOL_MOSTFREQ_NB(Company, GroupKey)
```

在此表达式中，CONSOL_MOSTFREQ_NB 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数会对 Company 端口进行分析并返回最常出现的值。 表达式会将此值写入 Company1 输出端口。

CONSOL_SHORTEST

分析记录组中的端口，然后返回具有最少字符数的值。

语法

```
CONSOL_SHORTEST(string, group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

具有最少字符数的端口值。

如果两个或多个值的最少字符数相同，则策略将返回第一个满足条件的值。

示例

以下表达式将使用 CONSOL_SHORTEST 函数分析 FirstName 输入端口，并查找具有最少字符数的值：

```
FirstName1:= CONSOL_SHORTEST(FirstName, GroupKey)
```

在此表达式中，CONSOL_SHORTEST 函数使用 GroupKey 端口标识记录组。 在该记录组中，该函数会对 FirstName 端口进行分析并返回最短值。 表达式会将此值写入 FirstName1 输出端口。

基于行的合并函数

使用基于行的合并函数可以选择记录组中的记录。 必须在 IF-THEN-ELSE 语句中使用基于行的合并函数。

CONSOL_GETROWFIELD

读取由基于行的合并函数所标识的行，并返回指定端口的值。 使用数值参数指定端口。

必须将 CONSOL_GETROWFIELD 函数与以下某个基于行的合并函数一起使用：

- CONSOL_MODALEXACT
- CONSOL_MOSTDATA
- CONSOL_MOSTFILLED

对于基于行的合并函数中的每个输入端口，必须使用 CONSOL_GETROWFIELD 函数的一个实例。

语法

CONSOL_GETROWFIELD(*value*)

下表介绍了此命令的参数：

参数	Required/ Optional	说明
值	必需	数字，用于指示基于行的合并函数中的一个输入端口。 使用“0”指定函数中最左侧的端口。 使用后续的数字指示其他端口。

返回值

指定的端口值。 函数从基于行的合并函数所标识的行中读取该值。

示例

以下表达式将 CONSOL_GETROWFIELD 函数与 CONSOL_MOSTDATA 函数结合起来使用：

```
IF (CONSOL_MOSTDATA(First_Name,Last_Name,GroupKey,GroupKey))
THEN
First_Name1 := CONSOL_GETROWFIELD(0)
Last_Name1 := CONSOL_GETROWFIELD(1)
GroupKey1 := CONSOL_GETROWFIELD(2)
ELSE
First_Name1 := First_Name
Last_Name1 := Last_Name
GroupKey1 := GroupKey
ENDIF
```

在此表达式中，CONSOL_MOSTDATA 函数分析记录组中的行，并标识单个行。 CONSOL_GETROWFIELD 函数使用连续数字读取该行的端口值并将这些值写入输出端口。

CONSOL_MODALEXACT

标识具有最常使用的值的最大计数的行。

例如，请考虑具有三个端口的行，其中包含记录组中最常使用的值。 该行最常使用的值的计数为“3”。

必须将此函数与 CONSOL_GETROWFIELD 函数一起使用。 CONSOL_GETROWFIELD 会从 CONSOL_MODALEXACT 函数标识的行返回值。

语法

CONSOL_MODALEXACT(*string1, [string2, ..., stringN,] group by*)

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

对于为最常出现字段的最大计数进行评分的行，返回 TRUE，对所有其他行返回 FALSE。

示例

以下表达式使用 CONSOL_MODALEXACT 函数查找包含最常出现字段的最大计数的行：

```
IF (CONSOL_MODALEXACT(First_Name,Last_Name,GroupKey,GroupKey))
THEN
First_Name1 := CONSOL_GETROWFIELD(0)
Last_Name1 := CONSOL_GETROWFIELD(1)
GroupKey1 := CONSOL_GETROWFIELD(2)
ELSE
First_Name1 := First_Name
Last_Name1 := Last_Name
GroupKey1 := GroupKey
ENDIF
```

在此表达式中，CONSOL_MODALEXACT 函数可分析记录组中的行，并标识单个行。 CONSOL_GETROWFIELD 函数使用连续数字读取该行的端口值并将这些值写入输出端口。

CONSOL_MOSTDATA

标识包含所有端口中最多字符的行

必须将此函数与 CONSOL_GETROWFIELD 函数一起使用。 CONSOL_GETROWFIELD 会返回 CONSOL_MOSTDATA 函数所标识的行中的值。

语法

```
CONSOL_MOSTDATA(string1, [string2, ..., stringN,]
 group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

对于包含所有端口中最多字符的行，返回 TRUE；对于其他所有行，则返回 FALSE。

示例

以下表达式使用 CONSOL_MOSTDATA 函数查找包含最多字符的行：

```
IF (CONSOL_MOSTDATA(First_Name,Last_Name,GroupKey,GroupKey))
THEN
First_Name1 := CONSOL_GETROWFIELD(0)
Last_Name1 := CONSOL_GETROWFIELD(1)
GroupKey1 := CONSOL_GETROWFIELD(2)
```

```
ELSE
First_Name1 := First_Name
Last_Name1 := Last_Name
GroupKey1 := GroupKey
ENDIF
```

在此表达式中，CONSOL_MOSTDATA 函数分析记录组中的行，并标识单个行。 CONSOL_GETROWFIELD 函数使用连续数字读取该行的端口值并将这些值写入输出端口。

CONSOL_MOSTFILLED

标识包含最多非空字段数的行。

必须将此函数与 CONSOL_GETROWFIELD 函数一起使用。 CONSOL_GETROWFIELD 会从 CONSOL_MOSTFILLED 函数标识的行返回值。

语法

```
CONSOL_MOSTFILLED(string1, [string2, ..., stringN,]
 group by)
```

下表介绍了此命令的参数：

参数	Required/Optional	说明
string	必需	输入端口名称。
分组依据	必需	包含组标识符的输入端口的名称。

返回值

对于包含最多非空字段数的行，返回 TRUE，对所有其他行返回 FALSE。

示例

以下表达式使用 CONSOL_MOSTFILLED 函数查找包含最多字符的行：

```
IF (CONSOL_MOSTFILLED(First_Name,Last_Name,GroupKey,GroupKey))
THEN
First_Name1 := CONSOL_GETROWFIELD(0)
Last_Name1 := CONSOL_GETROWFIELD(1)
GroupKey1 := CONSOL_GETROWFIELD(2)
ELSE
First_Name1 := First_Name
Last_Name1 := Last_Name
GroupKey1 := GroupKey
ENDIF
```

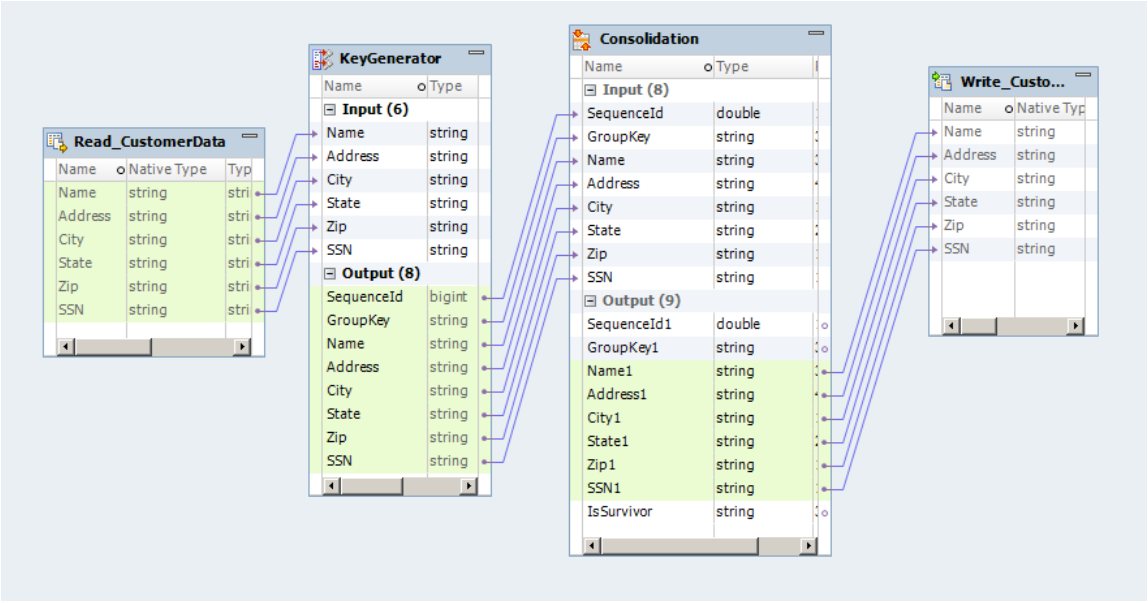
在此表达式中，CONSOL_MOSTFILLED 函数可分析记录组中的行，并标识单个行。 CONSOL_GETROWFIELD 函数使用连续数字读取该行的端口值并将这些值写入输出端口。

合并映射示例

您的组织需要合并重复的客户记录。 要合并客户记录，请使用键生成器转换对数据进行分组，然后使用合并转换来合并记录。

创建具有以下内容的映射：包含客户记录的数据源、键生成器转换、合并转换和数据目标。 映射会对客户记录进行分组、合并组，并写入单个合并记录。

下图显示了该映射：



输入数据

要分析的输入数据包含客户信息。

下表包含此示例的输入数据：

名称	地址	城市	省/自治区/直辖市	邮政编码	SSN
Dennis Jones	100 All Saints Ave	纽约	NY	10547	987-65-4320
Dennis Jones	1000 Alberta Rd	纽约	NY	10547	987-65-4320
D Jones	100 All Saints Ave	纽约	NY	10547-1521	

键生成器转换

使用键生成器转换基于邮政编码端口对输入数据进行分组。

转换将返回以下数据：

SequenceId	GroupKey	名称	地址	城市	省/自治区/直辖市	邮政编码	SSN
1	10547	Dennis Jones	100 All Saints Ave	纽约	NY	10547	987-65-4320
2	10547	Dennis Jones	1000 Alberta Rd	纽约	NY	10547	
3	10547	D Jones	100 All Saints Ave	纽约	NY	10547-1521	987-65-4320

合并转换

使用合并转换生成合并记录。

将合并转换配置为使用基于行的策略类型。 选择“精确模型”策略以选择具有最常使用值的最大计数的行。 “精确模型”策略使用该行中的值来生成合并记录。 合并记录是指在 IsSurvivor 端口中具有“Y”值的记录。

转换将返回以下数据：

GroupKey	名称	地址	城市	省/自治区/直辖市	邮政编码	SSN	IsSurvivor
10547	Dennis Jones	100 All Saints Ave	纽约	NY	10547	987-65-4320	N
10547	Dennis Jones	1000 Alberta Rd	纽约	NY	10547		N
10547	D Jones	100 All Saints Ave	纽约	NY	10547-1521	987-65-4320	N
10547	D Jones	100 All Saints Ave	纽约	NY	10547-1521	987-65-4320	Y

合并映射输出

配置合并转换，以便映射输出仅包含合并记录。

在本示例中，完全可以确定“精确模型”策略选择的最常使用的值是正确的端口值。 要仅将合并记录写入映射目标，请选择**高级**视图，然后将输出模式设置为“仅幸存者”。

运行映射后，映射输出将仅包含合并记录。

配置合并转换

配置合并转换时，请选择策略类型、选择策略或编写表达式、选择分组端口并配置高级选项。

1. 选择**合并**视图。
2. 选择一个策略类型。
3. 配置该策略。
   - 对于简单策略类型，请为每个端口选择一个策略。
   - 对于基于行的策略类型，请选择一个策略。
   - 对于高级策略类型，请创建使用合并函数的表达式。
4. 在“分组依据”字段中，选择包含组标识符的端口。
5. 如果输入数据没有排序，请在**高级**视图中启用排序。
6. 将输出配置为包含合并记录或所有记录。

非本地环境中的整合转换

非本地环境中的整合转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的整合转换

整合转换在本地环境和非本地环境中可能会对数据进行不同的处理。

支持该转换，但存在以下限制：

- 转换可能在每个环境中以不同的顺序处理记录。
- 转换可能在每个环境中将不同的记录标识为剩余记录。

Spark 引擎上的整合转换

整合转换在本地环境和非本地环境中可能会对数据进行不同的处理。

支持该转换，但存在以下限制：

- 转换可能在每个环境中以不同的顺序处理记录。
- 转换可能在每个环境中将不同的记录标识为剩余记录。

Databricks Spark 引擎上的整合转换

整合转换在本地环境和非本地环境中可能会对数据进行不同的处理。

支持该转换，但存在以下限制：

- 转换可能在每个环境中以不同的顺序处理记录。
- 转换可能在每个环境中将不同的记录标识为剩余记录。

第 12 章

数据屏蔽转换

本章包括以下主题：

数据屏蔽转换概览

数据屏蔽转换可将敏感的生产数据更改为非生产环境的现实测试数据。 数据屏蔽转换将基于为每一列配置的屏蔽技术来修改源数据。

可针对软件开发、测试、培训和数据挖掘创建屏蔽的数据。可以维护屏蔽数据中的数据关系，并维护数据库表之间的引用完整性。

数据屏蔽转换根据为列配置的源数据类型和屏蔽技术提供屏蔽规则。 对于字符串，可以限制字符串中要替换的字符。 可以限制屏蔽时要应用的字符。 对于数值和日期，可以为屏蔽的数据提供一个数字范围。可以根据原始数字的固定差额或百分比差额配置一个范围。数据集成服务将根据为转换配置的区域设置来替换字符。

屏蔽技术

屏蔽技术是要应用于选定列的数据屏蔽类型。

可以为输入列选择以下屏蔽技术之一：

随机

为相同的源数据和屏蔽规则生成不可重复的随机结果。可以屏蔽 Date、Numeric 和 String 数据类型。随机屏蔽无需使用种子值。随机屏蔽的结果不是确定性的。

表达式

对源列应用表达式可创建或屏蔽数据。您可以屏蔽所有数据类型。

键

将源数据替换为可重复的值。数据屏蔽转换将为相同的源数据、屏蔽规则和种子值生成确定性的结果。您可以屏蔽日期、数字和字符串数据类型。

置换

将数据列替换为字典中相似却无关的数据。您可以屏蔽字符串数据类型。

相关

根据其他源列的值替换某个源列的值。您可以屏蔽字符串数据类型。

标志化

用基于标志化屏蔽条件生成的数据替换源数据。数据屏蔽转换将应用在自定义算法中指定的规则。您可以屏蔽字符串数据类型。

加密

将元数据替换为根据您在转换中配置的加密标准加密的值。您可以加密字符串数据类型。

特殊屏蔽格式

信用卡号、电子邮件地址、IP 地址、电话号码、SSN、SIN 或 URL。数据屏蔽转换将应用内置规则以智能方式屏蔽这些常见类型的敏感数据。

无屏蔽

数据屏蔽转换不会更改源数据。

默认设置为“无屏蔽”。

随机屏蔽

随机屏蔽将生成不确定性的随机屏蔽数据。 当不同的行中出现相同的源值时，数据屏蔽转换将返回不同的值。 可以定义屏蔽规则以影响数据屏蔽转换所返回数据的格式。 使用随机屏蔽方法屏蔽数值、字符串值和日期值。

屏蔽字符串值

配置随机屏蔽可为字符串列生成随机输出。 要配置对输出字符串中每个字符的限制，请配置屏蔽格式。 配置筛选字符以定义要屏蔽的源字符和屏蔽所用的字符。

可以为字符串端口应用以下屏蔽规则：

范围

配置字符串的最小长度和最大长度。 数据屏蔽转换将返回一个介于最小字符串长度和最大字符串长度之间的随机字符串。

屏蔽格式

定义用于置换输入数据中每个字符的字符类型。 可以将每个字符限制为字母、数字或字母数字字符类型。

源字符串字符

定义源字符串中要屏蔽的字符。 例如，每当输入数据中出现数字符号 (#) 字符时，屏蔽该字符。 当“源字符串字符”为空时，数据屏蔽转换将屏蔽所有输入字符。

结果字符串替换字符

用“结果字符串字符”中定义的字符置换目标字符串中的字符。 例如，输入以下字符可将每个屏蔽配置为包含大写字母字符 A - Z：

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

屏蔽数值

屏蔽数值数据时，可以配置列的输出值范围。 数据屏蔽转换将根据端口精度返回一个介于范围上限和下限之间的值。 要定义范围，请基于原始源值的差额配置最小和最大范围值或配置模糊范围。

可以为数值数据配置以下屏蔽参数：

范围

定义输出值的范围。 数据屏蔽转换将返回介于最小值和最大值之间的数值数据。

模糊范围

定义一个落在源数据的固定差额或百分比差额内的输出值范围。 数据屏蔽转换将返回接近源数据值的数值数据。 可以配置范围和模糊范围。

屏蔽日期值

要使用随机屏蔽来屏蔽日期值，请配置输出日期的范围或选择一个差额。 配置差额时，请选择要模糊的日期部分。 选择年、月、日、小时、分钟或秒。 数据屏蔽转换将返回配置范围内的一个日期。

屏蔽日期时间值时，可以配置以下屏蔽规则：

范围

为选定的日期时间值设置要返回的最小值和最大值。

模糊

基于要应用于日期单位的差额来屏蔽日期。 数据屏蔽转换将返回差额范围内的一个日期。 可以模糊年、月、日、小时、分钟或秒。 选择要应用的低差额和高差额。

表达式屏蔽

表达式屏蔽会对端口应用表达式以更改数据或创建新数据。 在配置表达式屏蔽时，请在“表达式编辑器”中创建一个表达式。 选择输入和输出端口、函数、变量和运算符以构建表达式。

可以连接多个端口中的数据以便为其他端口创建值。 例如，需要创建登录名。 源具有名字列和姓氏列。 屏蔽查找文件中的名字和姓氏。 在数据屏蔽转换中，创建另一个名为“Login”的端口。 对于 Login 端口，配置一个表达式以连接名字的首字母和姓氏：

```
SUBSTR(FIRSTNM,1,1)||LASTNM
```

从点击式界面选择函数、端口、变量和运算符，以便在构建表达式时最大程度地减少错误。

“表达式编辑器”将显示没有配置表达式屏蔽的输出端口。 无法将表达式的输出用作其他表达式的输入。 如果将输出端口名称手动添加到表达式中，可能会获得意外的结果。

创建表达式时，请验证表达式是否返回与端口数据类型匹配的值。 如果表达式端口的数据类型是 Numeric 而表达式为其他数据类型，则数据屏蔽转换将返回零。 如果表达式端口的数据类型是 String 而表达式为其他数据类型，则数据屏蔽转换将返回空值。

可重复表达式屏蔽

源列出现在多个表中，且需要使用相同值屏蔽每个表中的该列时，请配置可重复表达式屏蔽。

配置可重复表达式屏蔽时，数据屏蔽转换会将表达式结果保存到存储表中。 如果该列出现在其他源表中，数据屏蔽转换将返回存储表中的屏蔽值，而不是表达式中的屏蔽值。

字典名称

配置可重复的表达式屏蔽时，必须输入字典名称。 字典名称是允许多个数据屏蔽转换从相同的源值生成相同屏蔽值的键。 请在每个数据屏蔽转换中定义相同的字典名称。 字典名称可以是任意文本。

存储表

存储表包含会话之间可重复表达式屏蔽的结果。 存储表行包含源列和屏蔽的值对。 表达式屏蔽的存储表是来自置换屏蔽存储表的单独表。

当数据屏蔽转换每次使用可重复表达式屏蔽某个值时，将根据字典名称、区域设置、列名称和输入值搜索存储表。 如果在存储表中找到了行，将从存储表中返回屏蔽的值。 如果数据屏蔽转换未找到行，将根据列的表达式生成屏蔽值。

当存储中包含未加密数据并使用相同字典名称作为密钥时，需要加密表达式屏蔽的存储表。

加密存储表以用于表达式屏蔽

可以使用转换语言编码函数加密存储表。 如果已启用存储加密，则需要加密存储表。

1. 将 IDM_EXPRESSION_STORAGE 存储表作为源来创建一个映射。
2. 创建数据屏蔽转换。
3. 在屏蔽值端口上应用表达式屏蔽技术。
4. 对 MASKEDVALUE 端口使用以下表达式：

   ```
   Enc_Base64(AES_Encrypt(MASKEDVALUE, Key))
   ```

5. 将这些端口链接到目标。

示例

例如，Employees 表中包含以下列：

```
FirstName
LastName
LoginID
```

在数据屏蔽转换中，使用组合了 FirstName 和 LastName 的表达式屏蔽 LoginID。 配置可重复的表达式屏蔽。 输入一个字典名称，作为可重复屏蔽的键。

Computer_Users 表包含一个 LoginID 列，但不包含 FirstName 或 LastName 列：

```
Dept
LoginID
Password
```

要屏蔽 Computer_Users 中与 Employees 中相同的 LoginID，请为 LoginID 列配置表达式屏蔽。 启用可重复屏蔽并输入为 Employees 表中 LoginID 所定义的字典名称。 集成服务将从存储表中检索 LoginID 值。

请创建一个当集成服务在存储表中找不到 LoginID 行时要使用的默认表达式。 Computer_Users 表没有 FirstName 或 LastName 列，因此表达式将创建一个意义较小的 LoginID 列。

存储表脚本

Informatica 提供可运行以创建存储表的脚本。 这些脚本位于以下位置：

```
<PowerCenter installation directory>\client\bin\Extensions\DataMasking
```

该目录包含 Sybase、Microsoft SQL Server、IBM DB2 和 Oracle 数据库的脚本。 每个脚本都命名为 Expression_<数据库类型>。

表达式屏蔽的规则和准则

对于表达式屏蔽，请使用以下规则和准则：

- 无法将表达式的输出用作其他表达式的输入。 如果将输出端口名称手动添加到表达式中，可能会获得意外的结果。
- 使用点击方法来构建表达式。 从点击式界面选择函数、端口、变量和运算符，以便在构建表达式时最大程度地减少错误。
- 如果已为可重复的屏蔽配置数据屏蔽转换，并且不存在存储表，则集成服务会将源数据替换为默认值。

键屏蔽

每当源值和种子值相同时，配置了键屏蔽的列将返回确定性的屏蔽数据。 数据屏蔽转换将为该列返回唯一值。

为列配置键屏蔽时，数据屏蔽转换将为该列创建种子值。 可以更改种子值，以便在不同的数据屏蔽转换之间生成可重复的数据。 例如，配置键屏蔽以强制执行引用完整性。 使用相同的种子值屏蔽表中的主键和另一表中的外键值。

可以定义屏蔽规则以影响数据屏蔽转换所返回数据的格式。 使用键屏蔽来屏蔽字符串值和数值。

屏蔽字符串值

可以配置键屏蔽，以便为字符串生成可重复输出。 配置屏蔽格式，以便定义对输出字符串中每个字符的限制。 配置源字符串字符，这些字符定义要屏蔽的源字符。 配置结果字符串替换字符，以便将屏蔽的数据限制为特定字符。

可以为键屏蔽字符串配置以下屏蔽规则：

种子

应用种子值，以便为列生成确定性屏蔽数据。 可以输入一个介于 1 和 1,000 之间的数字。

屏蔽格式

定义用于置换输入数据中每个字符的字符类型。 可以将每个字符限制为字母、数字或字母数字字符类型。

源字符串字符

定义源字符串中要屏蔽的字符。 例如，每当输入数据中出现数字符号 (#) 字符时，屏蔽该字符。 当“源字符串字符”为空时，数据屏蔽转换将屏蔽所有输入字符。 如果源字符串的字符数小于结果字符串的字符数，则数据屏蔽转换不会始终返回唯一数据。

结果字符串字符

用“结果字符串字符”中定义的字符置换目标字符串中的字符。 例如，输入以下字符可将每个屏蔽配置为全部包含大写字母字符：

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

屏蔽数值

为数值源数据配置键屏蔽以生成确定性的输出。 为列配置数值键屏蔽时，需要为该列分配随机种子值。 数据屏蔽转换屏蔽源数据时，将应用需要使用种子的屏蔽算法。

如果不同的列中出现相同的源值，可以更改列的种子值以生成可重复的结果。 例如，要维护两个表之间的主键-外键关系， 请在每个数据屏蔽转换中，为主键列输入与外键列相同的种子值。 数据屏蔽转换将为相同的数值生成确定性的结果。 表之间的引用完整性得到维护。

屏蔽日期时间值

如果可以为日期时间值配置键屏蔽，数据屏蔽转换将需要使用随机数作为种子。 可以更改种子以匹配其他列的种子值，以便返回列之间可重复的日期时间值。

数据屏蔽转换可以使用键屏蔽方法屏蔽介于 1753 和 2400 之间的日期。 如果源年份是闰年，则数据屏蔽转换返回的年份也是闰年。 如果源月份包含 31 天，则数据屏蔽转换返回的月份也是 31 天。 如果源月份是二月，则数据屏蔽转换将返回二月。

数据屏蔽转换将始终生成有效日期。

置换屏蔽

置换屏蔽将一列数据替换为相似但无关的数据。 使用置换屏蔽将生产数据替换为真实的测试数据。 配置置换屏蔽时，请定义包含置换值的字典。

数据屏蔽转换会对您配置的字典执行查找。 数据屏蔽转换将源数据替换为字典中的数据。 字典文件可以包含字符串数据、日期时间值、整数以及浮点数。 按以下格式输入日期时间值：

```
mm/dd/yyyy
```

可将数据置换为可重复或不可重复值。 选择可重复值时，数据屏蔽转换将为相同的源数据和种子值生成确定性的结果。 必须配置种子值才能将数据置换为确定性的结果。集成服务为可重复屏蔽维护一份源值和已屏蔽值的存储表。

可将多列数据置换为同一字典行中的已屏蔽值。 为一个输入列配置置换屏蔽。 为从同一字典行接收已屏蔽数据的其他列配置相关数据屏蔽。

字典

字典是一个引用表，该表包含置换数据以及表中每一行的序列号。 使用导入到模型存储库中的平面文件或关系表创建一个引用表以进行置换屏蔽。

数据屏蔽转换会生成一个数字，以便按序列号检索字典行。 数据屏蔽转换会生成一个哈希键以进行可重复的置换屏蔽，或者生成一个随机数字以进行不可重复的屏蔽。 如果要配置可重复的置换屏蔽，则可以配置一个附加查找条件。

可以配置一个字典以屏蔽数据屏蔽转换中的多个端口。

当数据屏蔽转换从字典中检索置换数据时，该转换不会检查置换数据值是否与原始值相同。 例如，数据屏蔽转换可能会用姓名 John 置换字典文件中的相同姓名 (John)。

以下示例显示了一个字典表，其中包含名字和性别：

SNO	性别	名字
1	M	Adam
2	M	Adeel
3	M	Adil

SNO	性别	名字
4	F	Alice
5	F	Alison

在此字典中，行中的第一个字段是序列号，第二个字段是性别。 集成服务始终会按序列号查找字典记录。 如果要配置可重复的屏蔽，则可以将性别添加为查找条件。 集成服务会使用哈希键从字典中检索行，并查找性别与源数据中的性别匹配的行。

创建引用表时，请遵循以下规则和准则：

- 表中的每个记录都必须有一个序列号。
- 序列号是指从 1 开始的有序整数。 序列中的序列号不能缺号。
- 序列号列可位于表行中的任意位置。 该列可以具有任何标签。

如果使用平面文件表创建引用表，请遵循以下规则和准则：

- 平面文件表的第一行必须具有列标签，才能标识每个记录中的字段。 这些字段以逗号分隔。 如果第一行不包含列标签，则集成服务会使用第一行中的字段值作为列名称。
- 如果在 Windows 上创建了一个平面文件表，并将其复制到 UNIX 计算机中，请验证该文件格式是否适用于 UNIX。 例如，Windows 和 UNIX 使用不同的字符作为行尾标记。

存储表

数据屏蔽转换为会话间的可重复置换维护存储表。 存储表行包含源列和屏蔽的值对。 当数据屏蔽转换每次使用可重复置换值屏蔽某个值时，将根据字典名称、区域设置、列名称、输入值和种子搜索存储表。 如果找到一个行，将从存储表返回该屏蔽值。 如果数据屏蔽转换未找到行，将从字典接收一个行以及一个哈希键。

存储表中的字典名称格式对于平面文件字典和关系字典是不同的。 平面文件字典名称通过文件名进行标识。 关系字典名称使用以下语法：

```
<Connection object>_<dictionary table name>
```

Informatica 提供可运行以创建关系存储表的脚本。 这些脚本位于以下位置：

```
<PowerCenter Client installation directory>\client\bin\Extensions\DataMasking
```

该目录包含 Sybase、Microsoft SQL Server、IBM DB2 和 Oracle 数据库的脚本。 每个脚本都命名为 Substitution_<数据库类型>。 如果您配置 SQL 语句和主键约束，就能在一个不同的数据库中创建表。

当存储中具有未加密的数据并使用了相同的种子值和字典来加密相同的列时，需要为置换屏蔽加密存储表。

加密用于置换屏蔽的存储表

可以使用转换语言编码函数加密存储表。 如果已启用存储加密，则需要加密存储表。

1. 将 IDM_SUBSTITUTION_STORAGE 存储表作为源来创建一个映射。
2. 创建数据屏蔽转换。
3. 对输入值端口和屏蔽值端口应用置换屏蔽技术。
4. 对 INPUTVALUE 端口使用以下表达式：

   ```
   Enc_Base64(AES_Encrypt(INPUTVALUE, Key))
   ```

5. 对 MASKEDVALUE 端口使用以下表达式：

   ```
   Enc_Base64(AES_Encrypt(MASKEDVALUE, Key))
   ```

6. 将这些端口链接到目标。

置换屏蔽属性

可以为置换屏蔽配置以下屏蔽规则：

- **可重复的输出**。返回会话之间的确定性结果。数据屏蔽会转换将屏蔽值存储在存储表中。
- **种子值**。应用种子值，以便为列生成确定性屏蔽数据。输入一个介于 1 到 1,000 之间的数字。
- **唯一输出**。强制数据屏蔽转换为唯一的输入值创建唯一的输出值。两个不同的输入值在屏蔽后不会产生相同的输出值。字典必须具有足够的唯一行来支持唯一输出。
  如果禁用唯一输出，则数据屏蔽转换可能不会为输入值生成唯一的屏蔽输出值。字典包含的行可能会较少。
- **唯一端口**。该端口用于标识要进行置换屏蔽的唯一记录。例如，您要对客户表中的名字进行屏蔽。如果选择包含名字的表列作为唯一端口，则数据屏蔽转换会将重复的名字替换为相同的屏蔽值。如果选择 Customer_ID 列作为唯一端口，则数据屏蔽转换会将每个名字替换为唯一值。
- **优化字典用法**。

  选择**可重复的输出**选项时适用。在字典中增加屏蔽值的使用。
- **字典信息**。配置包含置换数据值的引用表。单击**选择源**以选择引用表。
  - **字典名称**。显示您选择的引用表的名称。
  - **字典连接**。显示包含字典的连接的名称。
  - **序列号列**。选择要返回到数据屏蔽转换的列。
  - **排序列**。您要对条目进行排序的字典列。指定排序列以生成确定性结果，即使字典中条目的顺序发生变化也是如此。例如，如果您移动了关系字典并且条目顺序发生了变化，则对序列号列上进行排序以始终屏蔽数据。

    注意: 您选择的列必须包含唯一值。不能使用可能包含重复值的列对数据进行排序。
  - **输出列**。选择要返回到数据屏蔽转换的列。
- **查找条件。**配置查找条件以进一步限定要用于置换屏蔽的字典行。查找条件类似于 SQL 查询中的 WHERE 子句。配置查找条件时，可以将源中某个列的值与字典中某个列的值进行比较。

  例如，您要屏蔽名字。源数据和字典中都有名字列和性别列。您可以添加一个条件，使每个女性名字都用字典中的一个女性名字来替换。查找条件会将源中的性别与字典中的性别进行比较。
  - **输入端口**。要在查找中使用的源数据列。
  - **字典列**。要与输入端口进行比较的字典列。

置换屏蔽的规则和准则

对于置换屏蔽，请使用以下规则和准则：

- 如果唯一可重复置换屏蔽的存储表不存在，会话将失败。
- 如果字典不包含行，数据屏蔽转换将返回错误消息。
- 数据屏蔽转换找到存储表中包含区域设置、字典和种子的输入值时，它将检索屏蔽值，即使行不再位于字典中。
- 如果删除连接对象或修改字典，请截断存储表。 否则，可能会产生意外结果。
- 如果字典中的值数量小于源数据中的唯一值数量，数据屏蔽转换将无法使用唯一可重复值来屏蔽数据。 数据屏蔽转换将返回错误消息。

相关屏蔽

相关屏蔽会将源数据的多个列置换为相同字典行中的数据。

数据屏蔽转换对多个列执行置换屏蔽时，屏蔽的数据可能包含不切实际的字段组合。 可以配置相关屏蔽，以便置换同一字典行中多个输入列的数据。 屏蔽的数据将接收到有效组合，例如“New York, New York”或“Chicago, Illinois”。

配置相关屏蔽时，请先配置进行置换屏蔽的输入列。 配置要与该置换列相关的其他输入列。 例如，选择邮政编码列进行置换屏蔽，然后选择要与该邮政编码列相关的城市列和省/自治区/直辖市列。 相关屏蔽可确保所置换的城市和省/自治区/直辖市值对于所置换的邮政编码值有效。

注意: 如果没有先配置进行置换屏蔽的列，将无法配置相关屏蔽列。

配置相关屏蔽列时，请配置以下屏蔽规则：

相关列

要配置为置换屏蔽的输入列的名称。 数据屏蔽转换将使用适用于该列的屏蔽规则从字典中检索置换数据。 配置为置换屏蔽的列将成为从字典中检索屏蔽数据的键列。

输出列

字典列的名称，其中包含配置为相关屏蔽的列的值。

相关屏蔽示例

数据屏蔽字典中可能包含具有以下值的地址行：

SNO	STREET	CITY	STATE	邮政编码	COUNTRY
1	32 Apple Lane	Chicago	IL	61523	US
2	776 Ash Street	Dallas	TX	75240	US
3	2229 Big Square	Atleeville	TN	38057	US
4	6698 Cowboy Street	Houston	TX	77001	US

需要从“地址”字典中屏蔽包含城市、省/自治区/直辖市和邮政编码的有效组合的源数据。

为 ZIP 端口配置置换屏蔽。 为 ZIP 端口输入以下屏蔽规则：

规则	值
字典名称	地址
序号列	SNO
输出列	邮政编码

为“城市”端口配置相关屏蔽。 为“城市”端口输入以下屏蔽规则：

规则	值
相关列	邮政编码
输出列	城市

为“省/自治区/直辖市”端口配置相关屏蔽。 为“省/自治区/直辖市”端口输入以下屏蔽规则：

规则	值
相关列	邮政编码
输出列	省/自治区/直辖市

当数据屏蔽转换屏蔽邮政编码时，将为字典行中的邮政编码返回正确的城市和省/自治区/直辖市。

标志化屏蔽

使用标志化屏蔽技术可根据您在算法中指定的条件屏蔽源字符串数据。 例如，您可以创建一个包含虚假电子邮件地址的算法，以替换源数据中的字段条目。

可以使用标志化屏蔽配置屏蔽的数据的格式。 必须先向屏蔽算法分配一个标志器名称，才能使用该算法。 标志器名称引用所使用的屏蔽算法 (JAR)。 应用标志化屏蔽技术时，请指定标志器名称。

配置标志化屏蔽

使用标志化屏蔽技术之前，请执行以下任务：

1. 浏览到以下路径中的 tokenprovider 目录：<Informatica_home>\services\shared。
2. 打开以下 XML 文件：com.informatica.products.ilm.tx-tokenizerprovider.xml。
3. 添加标志器名称以及要使用的每个标志器的类文件的完全限定名称。 在 tokenprovider 目录中实现 com.informatica.products.ilm.tx-tokenprovider-<Build-Number>.jar 类中的标志器类。 对于每个标志器，在 XML 文件中输入的信息如下例所示：

   ```
   <TokenizerProvider>
   <Tokenizer Name="CCTokenizer"
   ClassName="com.informatica.tokenprovider.CCTokenizer"/>
   </TokenizerProvider>
   ```

   其中：

   - 标志器名称是用户定义的名称，用引号引起。
   - ClassName 是 CLASSNAME 属性的用户定义的名称。 从 com.informatica.products.ilm.tx-tokenprovider-<Build-Number>.jar 内部实现此类。

配置完成后，即可使用标志化屏蔽技术。 输入标志器名称，以指定创建映射时要使用的算法。

加密

加密屏蔽应用加密算法来屏蔽源数据。

使用加密屏蔽来屏蔽字符串数据类型。

您可以选择保留源数据的格式和长度或源数据的长度。您还可以选择在加密后更改源数据的格式和长度。

您可以选择不想加密的字符。

加密源数据后，还可以对其进行解密以恢复原始数据。要解密数据，必须创建和运行一个映射，该映射要使用加密源数据时使用的加密技术和密码。将模式设置为“解密”。

注意: 如果源数据包含 UTF-8 四字节字符，则无法使用加密来屏蔽数据。

选择以下加密技术之一：

保留格式和元数据

使用“保留格式和元数据”加密选项可以保留源数据的格式和长度。选择保留格式和元数据时，加密后，所有大写字符都将替换为大写字符，小写字符替换为小写字符，数字替换为数字，特殊字符替换为特殊字符。例如，电子邮件地址 Abc123@xyz.com 可能会变成 Mpz849#dje!kuw。在此示例中，如果您将 "@" 和 "."字符配置为“不加密字符”），则该电子邮件可能会变成 Mpz849@dje.kuw。

保留元数据

使用“保留元数据”加密选项可以保留源数据的长度。选择保留元数据时，加密后，数据的长度保持不变。例如，Alexender 这个名字可能会变成 jl6＃HB91v，其长度与源数据中的长度相同。

更改元数据

使用“更改元数据”加密选项可以更改源数据在加密后的长度。当您选择更改元数据时，加密后的数据不会保留源数据的长度和格式。例如，城市名称 London 可能会变成 Xuep@8f5, fmch529 或 6ky#ke33h*we。

注意: 在使用“更改元数据”加密选项之前，必须更改要应用加密的数据库中的列精度。

可以使用以下公式计算精度，并将该值舍入为下一个更高的整数：

```
Required Precision = (1.33*Original Precision)+24
```

更改数据库中的列精度后，必须更新映射中的列精度。要更新列精度，可以从更新后的数据库中重新导入元数据，也可以在映射的每个转换中手动更改列精度。

屏蔽规则

屏蔽规则是在选择屏蔽技术后配置的选项。

选择随机屏蔽技术或键屏蔽技术后，可以配置屏蔽格式、源字符串字符和结果字符串字符。 可以使用随机屏蔽配置范围或模糊。

下表介绍了对于每种屏蔽技术可以配置的屏蔽规则：

屏蔽规则	说明	屏蔽技术	源数据类型
屏蔽格式	屏蔽时将输出字符串中的每个字符限制为字母、数字或字母数字字符。	随机和键	String
源字符串字符	要屏蔽的源字符集或要从屏蔽中排除的源字符集。	随机和键	String
结果字符串替换字符	要在屏蔽中包含或排除的字符集。	随机和键	String

屏蔽规则	说明	屏蔽技术	源数据类型
范围	输出值的范围。 - 数值。 数据屏蔽转换将返回介于最小值和最大值之间的数值数据。 - 字符串。 返回介于最小字符串长度和最大字符串长度之间的随机字符串。 - 日期/时间。 返回介于最小日期时间 和最大日期时间之间的日期和时间。	随机	数值 String Date/Time
模糊	使用源数据固定差额或百分比差额的输出值范围。 数据屏蔽转换将返回接近源数据值的数据。 日期时间列要求使用固定差额。 列要求使用固定差额。	随机	数值 Date/Time

屏蔽格式

配置屏蔽格式，将输出列中的每个字符限制为字母、数字或字母数字字符。使用以下字符定义屏蔽格式：

```
A, D, N, X, +, R
```

注意: 屏蔽格式包含大写字符。输入小写屏蔽字符时，数据屏蔽转换会将该字符转换为大写。

下表介绍了屏蔽格式字符：

Character	说明
A	字母字符。例如，ASCII 字符 a 到 z 以及 A 到 Z。
D	数字 0 到 9。对于数字 0 到 9 之外的其他字符，数据屏蔽转换将返回“X”。
N	字母数字字符。例如，ASCII 字符 a 到 z、A 到 Z 以及 0-9。
X	任何字符。例如，字母数字或符号。
+	无屏蔽。
R	剩余字符。R 指定字符串中的剩余字符可以是任何字符类型。R 必须显示为屏蔽中的最后一个字符。

例如，某部门名称使用以下格式：

```
nnn-<department_name>
```

可以配置屏蔽，强制前三个字符为数字、部门名称为字母，并且在输出中保留短划线。配置以下屏蔽格式：

```
DDD+AAAAAAAAAAAAAAA
```

数据屏蔽转换会将前三个字符替换为数字字符。不会替换第四个字符。数据屏蔽转换会将剩余字符替换为字母字符。

如果未定义屏蔽格式，则数据屏蔽转换会将每个源字符替换为任意字符。如果屏蔽格式比输入字符串长，则数据屏蔽转换将忽略屏蔽格式中的多余字符。如果屏蔽格式比源字符串短，则数据屏蔽转换将屏蔽格式为 R 的剩余字符。

注意: 无法使用范围选项配置屏蔽格式。

源字符串字符

源字符串字符是选择要对其进行屏蔽或不对其进行屏蔽的源字符。 源字符串中字符的位置不会造成影响。 源字符区分大小写。

可以配置任意数量的字符。 当字符为空时，数据屏蔽转换将替换列中所有的源字符。

为源字符串字符选择以下选项之一：

仅屏蔽

数据屏蔽转换对源中配置为源字符串字符的字符进行屏蔽。 例如，如果输入字符 A、B 和 c，当该字符出现在源数据中时，数据屏蔽转换会将 A、B 或 c 替换为不同的字符。 不是 A、B 或 c 的源字符将不会发生更改。 屏蔽区分大小写。

除了以下内容，全部屏蔽

除了出现在源字符串中的源字符串字符，屏蔽所有字符。 例如，如果输入筛选器源字符“-”并选择“除了以下内容，全部屏蔽”，当该字符出现在源数据中时，数据屏蔽转换不会替换“-”字符。 其余源字符将发生更改。

源字符串示例

源文件包含一个名为“相关项”的列。 “相关项”列包含多个用逗号分隔的名称。 需要屏蔽“相关项”列，并在测试数据中使用逗号以分隔名称。

对于“相关项”列，请选择“源字符串字符”。 选择“不屏蔽”，然后输入“,”作为要跳过的源字符。 请勿输入引号。

数据屏蔽转换会替换掉源字符串中除逗号之外的所有字符。

结果字符串替换字符

结果字符串替换字符是您选择用于作为屏蔽数据中置换字符的字符。 配置结果字符串替换字符时，数据屏蔽转换将使用结果字符串替换字符替换源字符串中的字符。 为了避免为不同的输入值生成相同的输出，请配置多种置换字符，或者只屏蔽一些源字符。 字符串中每个字符的位置不会造成影响。

为结果字符串替换字符选择以下选项之一：

仅使用

仅使用您定义为结果字符串替换字符的字符屏蔽源。 例如，如果输入字符 A、B 和 c，数据屏蔽转换将使用 A、B 和 c 替换源列中的每个字符。单词“horse”将替换为“BAcBA”。

除了以下内容，全部使用

使用除您定义为结果字符串替换字符的字符外的任意字符屏蔽源。 例如，如果您输入 A、B 和 c 结果字符串替换字符，屏蔽数据将永远不会出现字符 A、B 或 c。

结果字符串替换字符示例

要将相关列中的所有逗号替换为分号，请完成以下任务：

1. 将逗号配置为源字符串字符，然后选择“仅屏蔽”。

   如果相关列中出现逗号，则数据屏蔽转换将仅屏蔽逗号。

2. 将分号配置为结果字符串替换字符，然后选择“仅使用”。

   数据屏蔽转换会将相关列中的每个逗号替换为分号。

范围

为数字、日期或字符串数据定义范围。 如果为数值或日期值定义了范围，则数据屏蔽转换将使用一个介于最小值和最大值之间的值来屏蔽源数据。 为字符串配置范围时，请配置字符串长度范围。

字符串范围

配置随机字符串屏蔽时，数据屏蔽转换将生成与源字符串长度不同的字符串。 或者，可以配置字符串宽度的下限和上限。 为宽度上限或下限输入的值必须是正整数。 每个宽度都必须小于或等于端口精度。

数值范围

为数值列设置最小值和最大值。 最大值必须小于或等于端口精度。 默认范围介于一到端口精度长度之间。

日期范围

为日期时间值设置最小值和最大值。 最小值字段和最大值字段包含默认的最小日期和最大日期。 默认的日期时间格式为 MM/DD/YYYY HH24:MI:SS。 最大的日期时间必须晚于最小的日期时间。

模糊

模糊将创建在源数据值的固定差额或百分比差额范围内的输出值。 配置模糊可返回接近原始值的随机值。 可以模糊数值和日期值。

模糊数值

选择固定差额或百分比差额以模糊数值源值。 低模糊值是小于源值的差额。 高模糊值是大于源值的差额。 低值和高值都必须大于或等于零。 数据屏蔽转换返回屏蔽的数据时，数值数据将落在您定义的范围内。

下表介绍了当输入源值为 66 时模糊范围值的屏蔽结果：

模糊类型	低	高	结果
固定	0	10	介于 66 与 76 之间
固定	10	0	介于 56 与 66 之间
固定	10	10	介于 56 与 76 之间
百分比	0	50	介于 66 与 99 之间
百分比	50	0	介于 33 与 66 之间
百分比	50	50	介于 33 与 99 之间

模糊日期值

通过配置模糊，将日期屏蔽为与源日期不同的日期。 选择要应用差额的日期单位。 可以选择年、月、日或小时。 输入上限和下限以定义高于或低于源日期中的单位的差额。 数据屏蔽转换将应用该差额并返回在该差额内的日期。

例如，要将屏蔽的日期限制为源日期两年内的日期，请选择年作为单位。 输入 2 作为上限和下限。 如果源日期是 02/02/2006，则数据屏蔽转换将返回一个介于 02/02/2004 和 02/02/2008 之间的日期。

默认情况下，模糊单位是年。

特殊屏蔽格式

特殊屏蔽格式是指可应用于常用类型数据的屏蔽。 使用特殊屏蔽格式后，数据屏蔽转换将返回采用真实格式的屏蔽值，但该值不是有效值。

例如，屏蔽 SSN 时，数据屏蔽转换将返回格式正确但无效的 SSN。 可为社会保障号配置可重复屏蔽。

为以下类型的数据配置特殊屏蔽：

- 社会保障号
- 信用卡号
- 电话号码
- URL 地址
- 电子邮件地址
- IP 地址
- 社会保险号

当用于屏蔽的源数据格式或数据类型无效时，数据集成服务将对数据应用默认屏蔽。 集成服务将应用默认值文件中的屏蔽值。 可以编辑默认值文件以更改默认值。

信用卡号屏蔽

数据屏蔽转换在屏蔽有效的信用卡号时，逻辑上会生成一个有效的信用卡号。 源信用卡号的长度必须介于 13 到 19 位数字之间。 根据信用卡行业规则，输入信用卡号的校验和必须有效。

源信用卡号可以包含数字、空格和连字符。 如果信用卡的字符数不正确，或者长度错误，则集成服务会将错误写入会话日志。 如果源数据无效，集成服务则会应用默认的信用卡号屏蔽。

数据屏蔽转换不会屏蔽六位银行标识号码 (BIN)。 例如，数据屏蔽转换可能会将信用卡号 4539 1596 8210 2773 屏蔽为 4539 1516 0556 7067。 数据屏蔽转换会创建一个具有有效校验和的屏蔽号。

电子邮件地址屏蔽

使用数据屏蔽转换屏蔽包含字符串值的电子邮件地址。 数据屏蔽转换可以使用随机 ASCII 字符屏蔽电子邮件地址，也可以将电子邮件地址替换为真实电子邮件地址。

您可以对电子邮件地址应用以下类型的屏蔽：

标准电子邮件屏蔽

数据屏蔽转换屏蔽电子邮件地址时会返回随机 ASCII 字符。 例如，数据屏蔽转换可将 Georgesmith@yahoo.com 屏蔽为 KtrIupQAPyk@vdSKh.BIC。 默认为标准。

高级电子邮件屏蔽

数据屏蔽转换会使用从转换输出端口或字典列中派生的其他真实电子邮件地址屏蔽电子邮件地址。

高级电子邮件屏蔽

通过高级电子邮件屏蔽类型，可以使用其他真实电子邮件地址来屏蔽电子邮件地址。 数据屏蔽转换会从字典列或转换输出端口创建电子邮件地址。

可以从映射输出端口创建电子邮件地址的本地部分。 或者，也可以从关系表或平面文件列创建电子邮件地址的本地部分。

数据屏蔽转换可从域字典中的一个常量值或一个随机值创建电子邮件地址的域名。

可以根据以下选项创建高级电子邮件屏蔽：

基于相关端口的电子邮件地址

可以根据数据屏蔽转换输出端口创建电子邮件地址。 为名字和姓氏列选择转换输出端口。 根据为名字和姓氏长度指定的值，数据屏蔽转换会屏蔽名字、姓氏或这两者。

基于字典的电子邮件地址

可以根据字典中的列创建电子邮件地址。 选择引用表作为字典的源。

为名字和姓氏选择字典列。 根据为名字和姓氏长度指定的值，数据屏蔽转换会屏蔽名字、姓氏或这两者。

高级电子邮件地址屏蔽类型的配置参数

配置高级电子邮件地址屏蔽时指定配置参数。

您可以指定以下配置参数：

分隔符

可以选择分隔符来分隔电子邮件地址中的名和姓，例如点、连字符或下划线。 如果不想分隔电子邮件地址中的名和姓，请将分隔符保留为空白。

FirstName 列

选择数据屏蔽转换输出端口或字典列以屏蔽电子邮件地址中的名。

LastName 列

选择数据屏蔽转换输出端口或字典列以屏蔽电子邮件地址中的姓。

FirstName 或 LastName 列的长度

限制名和姓列的屏蔽字符长度。 例如，名的输入数据为 Timothy，姓的输入数据为 Smith。 选择 5 作为名列的长度。 选择 1 作为姓列的长度，并使用点作为分隔符。 数据屏蔽转换将生成以下电子邮件地址：

```
timot.s@<domain_name>
```

DomainName

您可以使用常量值作为域名，例如 gmail.com。 或者，可以指定包含一组域名的另一个字典文件。 域字典可以是平面文件或关系表。

IP 地址屏蔽

数据屏蔽转换通过将 IP 地址拆分成 4 个数字（用句点分隔）来将其屏蔽成其他 IP 地址。 第一个数字表示网络。 数据屏蔽转换可屏蔽网络范围内的网络号。

数据屏蔽转换会将 A 类 IP 地址屏蔽为 A 类 IP 地址，将 10.x.x.x 地址屏蔽为 10.x.x.x 地址。 数据屏蔽转换不会屏蔽类和专用网络地址。 例如，数据屏蔽转换可以将 11.12.23.34 屏蔽为 75.32.42.52.，将 10.23.24.32 屏蔽为 10.61.74.84。

注意: 屏蔽多个 IP 地址时，数据屏蔽转换可以返回非唯一值，因为它不会屏蔽类或专用网络的 IP 地址。

电话号码屏蔽

数据屏蔽转换可屏蔽电话号码，但不更改原始电话号码的格式。 例如，数据屏蔽转换可以将电话号码 (408)382 0658 屏蔽为 (607)256 3106。

源数据可包含数字、空格、连字符和括号。 集成服务不会屏蔽字母或特殊字符。

数据屏蔽转换可以屏蔽字符串、整型和长整型数据。

社会保障号屏蔽

数据屏蔽转换将基于社会保障局的最新高层组织列表生成无效的社会保障号。 高层组织列表包含社会保障局已签发的有效编号。

默认高层组织列表为以下位置中的文本文件：

```
<Installation Directory>\infa_shared\SrcFiles\highgroup.txt
```

要在工作流中使用高层组织列表文件，请将文本文件复制到为数据集成服务配置的源目录。

数据屏蔽转换生成高层组织列表中不存在的 SSN 编号。 社会保障局每个月更新一次高层组织列表。 从以下位置下载最新版本的列表：

```
http://www.socialsecurity.gov/employer/ssns/highgroup.txt
```

社会保障号格式

数据屏蔽转换接受任何包含九位数字的 SSN 格式。 可使用任意字符集分隔这些数字。 例如，数据屏蔽转换接受以下格式： +=54-*9944$#789-,*()”.

区号要求

数据屏蔽转换会返回一个社会保障号，该号码因与源格式相同而无效。 SSN 的前三位数字用于定义区号。 数据屏蔽转换不会屏蔽区号。 它会屏蔽组号和序列号。 源 SSN 必须包含一个有效的区号。 数据屏蔽转换将在“高层组织列表”上查找区号，并确定一系列未使用的编号，该转换可将这些编号作为屏蔽数据来应用。 如果 SSN 无效，则数据屏蔽转换不会屏蔽源数据。

可重复社会保障号屏蔽

数据屏蔽转换将返回具有可重复屏蔽的确定性社会保障号。 数据屏蔽转换无法返回所有唯一社会保障号，因为它无法返回社会保障局已签发的有效社会保障号。

URL 地址屏蔽

数据屏蔽转换通过搜索“://”字符串并解析其右侧的子字符串来解析 URL。 源 URL 必须包含“://”字符串。 源 URL 可以包含数字和字母字符。

数据屏蔽转换不会屏蔽 URL 协议。 例如，如果 URL 为 http://www.yahoo.com，数据屏蔽转换将返回 http://MgL.aHjCa.VsD/。 数据屏蔽转换会生成无效 URL。

注意: 数据屏蔽转换总是针对 URL 返回 ASCII 字符。

社会保险号屏蔽

数据屏蔽转换可屏蔽九位数的社会保险号。 可使用任意字符集分隔这些数字。

如果该号码不包含分隔符，则屏蔽后的号码也不包含分隔符。 否则屏蔽的号码将采用以下格式：

```
xxx-xxx-xxx
```

可重复的 SIN 编号

可以将数据屏蔽转换配置为返回可重复的 SIN 值。 为可重复的 SIN 屏蔽配置端口时，数据屏蔽转换将在每次源 SIN 值和种子值相同时返回确定性的屏蔽数据。

要返回可重复的 SIN 编号，请启用**可重复的值**并输入种子号。 数据屏蔽转换将返回每个 SIN 的唯一值。

SIN 起始数字

您可以定义屏蔽 SIN 的第一个数字。

启用**起始数字**，然后输入数字。 数据屏蔽转换会从您输入的数字开始创建屏蔽 SIN 数字。

默认值文件

当源数据格式或数据类型对于某个屏蔽无效时，数据集成服务将对数据应用默认屏蔽。 集成服务将应用默认值文件中的屏蔽值。 可以编辑默认值文件以更改默认值。

默认值文件是位于以下位置的 XML 文件：

<安装目录>\infa_shared\SrcFiles\defaultValue.xml

要在工作流中使用默认值文件，请将默认值文件复制到为数据集成服务配置的源目录中。

defaultValue.xml 文件包含以下名称-值对：

```
<?xml version="1.0" standalone="yes" ?>
<defaultValue
default_char = "X"
default_digit = "9"
default_date = "11/11/1111 00:00:00"
default_email = "abc@xyz.com"
default_ip = "99.99.9.999"
default_url = "http://www.xyz.com"
default_phone = "999 999 999 9999"
default_ssn = "999-99-9999"
default_cc = "9999 9999 9999 9999"
default_sin = "999-999-999"
default_seed = "500"/>
```

相关主题：

- “配置数据集成服务” 页面上 195

数据屏蔽转换配置

使用以下步骤配置数据屏蔽转换。

1. 配置数据集成服务的执行选项。
2. 创建转换。
3. 定义输入端口。
4. 为要更改的每个端口配置屏蔽规则。
5. 预览数据以验证结果。

配置数据集成服务

您可以在 Informatica Administrator（Administrator 工具）中配置数据集成服务的执行选项。

配置执行选项以设置以下默认目录：

- 主目录。 包含源目录和缓存目录。
- 源目录。 包含工作流的源文件。 例如，源目录中可能包含 highgrp.txt 和 defaultvalue.xml 文件。
- 缓存目录。 包含用于置换屏蔽的缓存文件。

要设置执行选项的值，请打开 Administrator 工具，然后在**域导航器**中选择数据集成服务。 单击**属性**视图，然后单击**执行选项**部分中的**编辑**。

相关主题：

- “默认值文件” 页面上 194

创建数据屏蔽转换

在 Developer 工具中创建数据屏蔽转换。

创建数据屏蔽转换之前，请创建源。 导入平面文件或关系数据库表作为物理数据对象。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择数据屏蔽转换。
4. 单击**下一步**。
5. 输入转换的名称。
6. 单击**完成**。

   此时，转换将显示在编辑器中。

定义端口

在**概览**视图中添加数据屏蔽输入端口。 默认情况下，创建输入端口时，Developer 工具将创建相应的输出端口。输出端口与输入端口同名。

1. 在**概览**视图上，单击**新建**以添加端口。
2. 为列配置数据类型、精度和小数位数。

   必须先配置列数据类型，然后再定义列的屏蔽规则。
3. 要为端口配置数据屏蔽，请单击**概览**视图中屏蔽类型列中的箭头。

为每个端口配置数据屏蔽

在“数据屏蔽”对话框上为端口选择屏蔽技术和相应的屏蔽规则。 在**端口**选项卡上单击数据屏蔽列时，将显示“数据屏蔽”对话框。

1. 启用**应用屏蔽**以便为选定端口配置屏蔽。

   Developer 工具将根据要屏蔽的端口的数据类型显示可以使用的屏蔽技术列表。
2. 从列表中选择一种屏蔽技术。

   Developer 工具将根据您选择的屏蔽技术显示不同的屏蔽规则。 部分特殊屏蔽格式没有要配置的屏蔽规则。

3. 配置屏蔽规则。
4. 单击**确定**为端口应用数据屏蔽配置。

   为端口定义数据屏蔽时，Developer 工具将创建名为 **out-<port name>** 的输出端口。 **<port name>** 与输入端口同名。 数据屏蔽转换将返回 **out-<port name>** 端口中的屏蔽数据。

预览屏蔽的数据

在**数据查看器**中查看数据屏蔽转换结果时，可以将屏蔽数据与原始数据进行比较。

1. 配置数据屏蔽转换端口和屏蔽规则后，创建一个包含物理数据对象源和数据屏蔽转换的映射。
2. 将源连接到数据屏蔽转换。
3. 验证源是否包含数据集成服务可以访问的共享位置中的数据。
4. 单击数据屏蔽转换以在映射中选择该转换。
5. 单击**数据查看器**，然后单击**运行**。

   Developer 工具将显示所有数据屏蔽转换输出端口的数据。 具有 **output** 前缀的端口中包含屏蔽的数据。 可以在**数据查看器**视图中将屏蔽数据与原始数据进行比较。

数据屏蔽转换运行时属性

可以配置数据屏蔽转换运行时属性以提高性能。

配置以下运行时属性：

存储连接名称

使用存储连接的屏蔽类型需要此名称。用作存储连接的连接的名称。

缓存大小

主内存中字典缓存的大小。增加内存大小以提高性能。对于 100,000 条记录，建议的最小大小为 32 MB。默认值为 8 MB。

缓存目录

字典缓存的位置。必须具有该目录的写入权限。默认值为 CacheDir。

密码

加密屏蔽需要此密码。密码生成用于加密或解密数据的密钥。在您在 Test Data Management 中创建的映射中，密码经过加密。对于您在 Developer tool 中创建的映射，可以使用站点密钥来加密密码，并在文本框中输入加密后的值。

模式

加密屏蔽需要此密码。确定数据屏蔽转换对数据执行加密还是解密。将值设置为“加密”可加密源数据。要解密被屏蔽的数据并返回原始源数据，请使用相同的加密技术配置和密码来运行映射，并将模式设置为“解密”。

使用 SoftHSM

加密需要此属性。选择在加密过程中是否使用 SoftHSM。SoftHSM 更安全，但您可能会注意到性能有差异。

共享存储表

在数据屏蔽转换实例之间启用存储表共享。当两个数据屏蔽转换实例为数据库连接、种子值和区域设置使用相同的字典列时，可启用共享存储表。当同一数据屏蔽转换中的两个端口为连接、种子和区域设置使用相同的字典列时，也可以启用共享存储表。当数据屏蔽转换或端口未共享字典列时，将禁用共享存储表。默认为“已禁用”。

存储表提交时间间隔

一次提交到存储表中的行数。增加该值可提高性能。未配置共享存储表时，请配置提交时间间隔。默认值为 100,000。

对存储进行加密

对存储表（例如 IDM_SUBSTITUTION_STORAGE 和 IDM_EXPRESSION_STORAGE）进行加密。在启用加密存储属性之前，请验证是否已对存储表中的数据进行加密。如果不希望对存储表进行加密，请清除此选项。默认为“已禁用”。

存储加密键

数据屏蔽转换将基于存储加密键对存储进行加密。为同一数据屏蔽转换实例的所有会话运行使用相同的加密键。

置换字典所有者名称

选择置换屏蔽类型时的置换字典表所有者的名称。如果在数据库连接中指定的数据库用户不是会话中置换字典表的所有者，则需要指定表所有者。

存储所有者名称

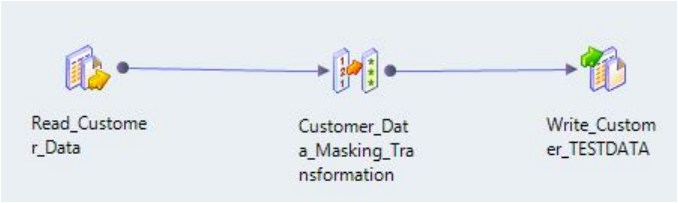
选择可重复表达式或唯一可重复置换屏蔽类型时，IDM_SUBSTITUTION_STORAGE 或 IDM_EXPRESSION_STORAGE 的表所有者名称。

数据屏蔽示例

开发人员需要为客户应用程序创建测试数据。 该数据必须包含其他开发人员可以在公司开发环境中访问的现实客户数据。

开发人员将创建一个数据服务以返回屏蔽的客户数据，例如客户 ID、信用卡号和收入。 映射包括用于转换客户数据的数据屏蔽转换。

下图显示了该映射：



该映射包含以下转换：

- Read_Customer_Data。包含客户信用卡和收入信息。
- Customer_Data_Masking 转换。 屏蔽除 FirstName 和 LastName 之外的所有列。 数据屏蔽转换会将屏蔽的列传递给目标。
- Customer_TestData。用于接收屏蔽的客户数据的输出转换。

Read_Customer 数据

客户数据中包含以下列：

列	数据类型
CustomerID	Integer
LastName	String
FirstName	String
CreditCard	String
收入	Integer
Join_Date	日期时间 (MM/DD/YYYY)

下表显示了客户数据示例：

CustomerID	LastName	FirstName	CreditCard	收入	JoinDate
0095	Bergeron	Barbara	4539-1686-3069-3957	12000	12/31/1999
0102	Brosseau	Derrick	5545-4091-5232-8948	4000	03/03/2011
0105	Anderson	Lauren	1234-5678-9012-3456	5000	04/03/2009
0106	Boonstra	Pauline	4217-9981-5613-6588	2000	07/07/2007
0107	Chan	Brian	4533-3156-8865-3156	4500	06/18/1995

客户数据屏蔽转换

数据屏蔽转换将屏蔽客户行中除名字和姓氏之外的所有列。

数据屏蔽转换将执行以下类型的屏蔽：

- 键屏蔽
- 随机屏蔽
- 信用卡屏蔽

下表显示了数据屏蔽转换中每个端口的屏蔽规则：

输入端口	屏蔽类型	屏蔽规则	说明
CustomerID	键	种子为 934。 客户 ID 不具有任何屏蔽格式。 结果字符串替换字符为 1234567890。	CustomerID 屏蔽是确定性的。 屏蔽的客户 ID 中包含数字。
LastName	无屏蔽	-	-
FirstName	无屏蔽	-	-
CreditCard	CreditCard	-	数据屏蔽转换会将信用卡号屏蔽为具有有效校验和的其他数字。

输入端口	屏蔽类型	屏蔽规则	说明
收入	随机	模糊 百分比 下限 = 1 上限 = 10	屏蔽的收入在源收入的百分之十之内。
JoinDate	随机	模糊 单位 = 年 下限 = 5 HighBound=5	屏蔽的日期在原始日期的 5 年内。

客户测试数据结果

Customer_TestData 转换将从数据屏蔽转换接收现实客户数据。

Customer_TestData 目标将接收以下数据：

out-CustomerID	out-LastName	outFirstName	out-CreditCard	out-Income	out-JoinDate
3954	Bergeron	Barbara	4539-1625-5074-4106	11500	03/22/2001
3962	Brosseau	Derrick	5545-4042-8767-5974	4300	04/17/2007
3964	Anderson	Lauren	1234-5687-2487-9053	5433	09/13/2006
3965	Boonstra	Pauline	4217-9935-7437-4879	1820	02/03/2010
3966	Chan	Brian	4533-3143-4061-8001	4811	10/30/2000

收入在原始收入的百分之十之内。 联接日期在原始日期的五年内。

数据屏蔽转换高级属性

配置有助于确定数据集成服务如何为数据屏蔽转换处理数据的属性。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的数据屏蔽转换

非本地环境中的数据屏蔽转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。不受支持。

Blaze 引擎上的数据屏蔽转换

支持数据屏蔽转换，但存在以下限制：

在下列情况下，映射验证会失败：

- 为可重复表达式屏蔽配置该转换。
- 为唯一可重复置换屏蔽配置该转换。

可以在此引擎上使用以下屏蔽技术：

信用卡

电子邮件

表达式

IP 地址

键

电话

随机

SIN

SSN

标志化

URL

随机置换

可重复置换

取决于随机置换

取决于可重复置换

Spark 引擎上的数据屏蔽转换

支持数据屏蔽转换，但存在以下限制：

在下列情况下，映射验证会失败：

- 为可重复表达式屏蔽配置该转换。
- 为唯一可重复置换屏蔽配置该转换。

可以在此引擎上使用以下屏蔽技术：

信用卡

电子邮件

表达式

IP 地址

键

电话

随机

SIN

SSN

标志化

URL

随机置换

可重复置换

取决于随机置换

取决于可重复置换

要优化数据屏蔽转换的性能，请在 Hadoop 连接中配置以下 Spark 引擎配置属性：

spark.executor.cores

指示每个执行程序进程用于在 Spark 引擎上运行小任务的内核数量。

设置为：`spark.executor.cores=1`

spark.executor.instances

指示每个执行程序进程用于在 Spark 引擎上运行小任务的实例数量。

设置为：`spark.executor.instances=1`

spark.executor.memory

指示每个执行程序进程用于在 Spark 引擎上运行小任务的内存量。

设置为：`spark.executor.memory=3G`

流映射中的数据屏蔽转换

在 Spark 引擎上流映射与批处理映射具有相同的处理规则。

第 13 章

数据处理器转换

本章包括以下主题：

数据处理器转换概览

数据处理器转换用于处理映射中非结构化和半结构化的文件格式。将转换配置为处理消息传递格式、HTML 页面、XML、JSON 和 PDF 文档。您也可以转换结构化格式，如 ACORD、HIPAA、HL7、EDI-X12、EDIFACT 和 SWIFT。

映射使用数据处理器转换将文档从一种格式更改为另一种格式。数据处理器转换可处理映射中任何格式的文件。创建数据处理器转换时，可定义转换数据的组件。

数据处理器转换可以包含多个用于处理数据的组件。每个组件都可能会包含其他组件。

例如，您可能会收到 Microsoft Word 文件形式的客户发票。可以配置数据处理器转换以解析每个 Word 文件中的数据。将客户数据提取到“客户”表中。将订单信息提取到“订单”表中。

创建数据处理器转换时，可定义 XMap、脚本或库。XMap 可将输入层次结构文件转换为其他结构的输出层次结构文件。库可将行业消息传递类型转换为具有层次结构的 XML 文档，或从 XML 转换为行业标准格式。脚本可以将源文档解析到层次结构格式、将层次结构格式转换为其他文件格式或将层次结构文档映射到其他层次结构格式。

在数据处理器转换 IntelliScript 编辑器中定义脚本。您可以定义以下类型的脚本：

- 解析器。将源文档转换为 XML。解析器的输出始终为 XML。输入可为任意格式，如文本、HTML、Word、PDF 或 HL7。

- 序列化程序。将 XML 文件转换为任意格式的输出文档。序列化程序的输出可以是任何格式，如文本文档、HTML 文档或 PDF。
- 映射程序。将 XML 源文档转换为其他 XML 结构或架构。您可以转换与 XMap 中相同的 XML 文档。
- 转换器。修改任意格式的数据。添加、删除、转换或更改文本。可将转换器与解析器、映射程序或序列化程序结合使用。也可以将转换器作为独立组件来运行。
- 流转化器。将较大的输入文档（如几千兆字节的数据流）拆分为多个段。流转化器可处理具有多个消息或记录的文档，如 HIPAA 或 EDI 文件。

数据处理器转换视图

数据处理器转换具有多个视图，您可在配置转换及在 Developer 工具中运行转换时访问这些视图。

默认情况下，某些数据处理器转换视图不会显示在 Developer 工具中。要更改转换的视图，请单击**窗口** > **显示视图** > **其他** > **Informatica**。选择要查看的视图。

数据处理器转换具有以下固定视图：

概览视图

配置端口并定义启动组件。

引用视图

添加架构或从转换中删除架构。

“设置”视图

配置转换的编码、输出控制和 XML 生成设置。

“对象”视图

添加、修改或从转换中删除脚本、XMap 和库对象。

还可以访问数据处理器转换的以下视图：

“数据处理器十六进制源”视图

以十六进制格式显示输入文档。

“数据处理器事件”视图

显示在 Developer 工具中运行转换时发生的事件的相关信息。显示初始化、执行和摘要事件。

“脚本帮助”视图

显示脚本编辑器的上下文相关帮助。

“数据查看器”视图

查看示例输入数据、运行转换和查看输出结果。

数据处理器转换端口

请在转换的**概览**视图中定义数据处理器转换端口。

数据处理器转换可以通过复杂的文件读取器读取文件、缓冲区或流式缓冲区中的输入。可以将平面文件读取器用作缓冲区，以便一次性读取整个文件。还可以读取数据库中的输入文件。

您创建的输出端口取决于是要从转换返回字符串、复杂文件还是关系数据行。

数据处理器转换输入端口

创建数据处理器转换时，Developer 工具将创建一个默认输入端口。 在脚本启动组件中定义附加输入端口时，开发程序工具会在转换中创建一个附加输入端口。

输入类型确定了数据集成服务传递给数据处理器转换的数据类型。 输入类型确定输入是数据还是源文件路径。

请配置以下输入类型之一：

缓冲区

数据处理器转换会在输入端口中接收到多行源数据。 配置转换以接收平面文件或来自 Informatica 转换的数据时，请使用缓冲区输入类型。

文件

数据处理器转换会在输入端口中接收到源文件路径。 数据处理器启动组件会打开源文件。 请使用文件输入类型解析二进制文件，如 Microsoft Excel 或 Microsoft Word 文件。 对于可能需要大量系统内存来处理缓冲区输入端口的大文件，也可以使用文件输入类型。

服务参数

数据处理器转换会接收到多个要应用于服务参数端口中变量的值。 选择变量以接收输入数据时，Developer 工具将为每个变量创建一个服务参数端口。

Output_Filename

配置默认输出端口以返回文件名而非行数据时，Developer 工具将创建 Output_Filename 端口。 可以通过映射将文件名传递给 Output_Filename 端口。

定义输入端口时，可为该端口定义示例输入文件的位置。 示例输入文件是一个小的输入文件示例。 创建脚本时将引用示例输入文件。 在**数据查看器**视图中测试转换时也可以使用示例输入文件。 请在**输入位置**字段中定义示例输入文件。

服务参数端口

可以创建多个用于接收变量值的输入端口。变量可以包含任何数据类型，例如字符串、日期或数值。变量还可以包含源文档的位置。您可以在数据处理器组件中引用变量。

为变量创建输入端口时，Developer 工具将显示您可以从中进行选择的变量列表。

创建服务参数端口

可以创建多个用于接收变量值的输入端口。 还可以删除通过变量创建的端口。

1. 打开数据处理器转换的**概览**视图。
2. 单击**选择**。

   Developer 工具将显示变量列表并指示哪个变量已具有端口。
3. 选择一个或多个变量。

   Developer 工具将为您选择的每个变量创建一个缓冲区输入端口。 您无法修改端口。
4. 要删除通过变量创建的端口，请禁用变量列表中的相应选择。 禁用相应选择时，Developer 工具将删除该输入端口。

数据处理器转换输出端口

默认情况下，数据处理器转换具有一个输出端口。 如果在脚本中定义了其他输出端口，则 Developer 工具会将这些端口添加到数据处理器转换。 如果配置转换以返回关系数据，则可以创建多组端口。 还可以创建服务参数端口和传递端口。

默认输出端口

默认情况下，数据处理器转换具有一个输出端口。 创建关系输出时，可以定义多组相关的输出端口以替代默认输出端口。 在脚本组件中定义附加输出端口时，开发程序工具会将附加输出端口添加到转换。

为默认输出端口配置以下输出类型之一：

缓冲区

数据处理器转换将通过输出端口返回 XML。 解析文档或将 XML 映射至数据处理器转换中的其他 XML 文档时，请选择缓冲区文件类型。

文件

对于每个源实例或行，数据集成服务将返回输出端口中的输出文件名。 数据处理器转换组件将写入输出文件，而不会通过数据处理器转换输出端口返回数据。

选择文件输出端口时，Developer 工具将创建 Output_Filename 输入端口。 可以将文件名传递给 Output 文件名端口。 数据处理器转换将使用在此端口中接收的名称创建输出文件。

如果输出文件名为空，则数据集成服务将返回行错误。 出现错误时，数据集成服务会向输出端口写入一个空值并返回行错误。

将 XML 转换为二进制数据文件（如 PDF 文件或 Microsoft Excel 文件）时，请选择“文件”输出类型。

传递端口

您可以为任何数据处理器转换配置传递端口。传递端口是接收输入数据并将相同数据返回映射而不进行任何更改的输入和输出端口。

您可以在映射中的数据处理器转换实例中配置传递端口。

要添加传递端口，请从映射的其他转换中拖动端口。 也可以在**属性**视图的**端口**选项卡中添加端口。单击**新建**添加一个传递端口。

注意: 向具有关系输入和层次结构输出的数据处理器转换添加传递端口时，请将端口添加到关系结构的根组。

数据处理器转换可以包含具有自定义数据类型的传递端口。

启动组件

启动组件定义用于在数据处理器转换中启动处理过程的组件。请在**概览**视图中配置启动组件。

数据处理器转换可以包含多个用于处理数据的组件。每个组件都可能会包含其他组件。您必须标识哪个组件作为转换的入口点。

在数据处理器转换中配置启动组件时，可以选择 XMap、库或脚本组件作为启动组件。关于脚本，您可以选择以下组件类型之一：

- 解析器。将源文档转换为 XML。输入可为任意格式，如文本、HTML、Word、PDF 或 HL7。
- 映射程序。将 XML 源文档转换为其他 XML 结构或架构。

- 序列化程序。将 XML 文件转换为任意格式的输出文档。
- 流转化器。将较大的输入文档（如几千兆字节的数据流）拆分为多个段。
- 转换器。修改任意格式的数据。添加、删除、转换或更改文本。可将转换器与解析器、映射程序或序列化程序结合使用。也可以将转换器作为独立组件来运行。

注意: 如果启动组件不是 XMap 或库，也可以在脚本中（而非**概览**视图中）配置启动组件。

引用

通过选择架构或 Mapplet 用作引用，可以定义转换引用（例如架构或 Mapplet 引用）。某些数据处理器转换需要层次结构架构以针对转换中的相关组件定义输入或输出层次结构。要在转换中使用架构，可以为转换定义架构引用。还可以使用称为 RunMapplet 操作的专用操作以从数据处理器转换调用 Mapplet。要调用 Mapplet，必须首先为转换定义 Mapplet 引用。

可以在转换**引用**视图中定义转换引用（例如架构或 Mapplet 引用）。

架构引用

数据处理器转换将引用模型存储库中的架构对象。在创建转换之前，存储库中可以存在架构对象。您还可以在转换的**引用**视图中导入架构。

架构编码必须与序列化程序或映射程序对象的输入编码匹配。架构编码必须与解析器或映射程序对象的输出编码匹配。请在转换的**设置**视图中配置工作编码。

一个架构可引用其他多个架构。Developer 工具将显示数据处理器转换引用的每个架构的命名空间和前缀。引用命名空间为空的多个架构时，转换无效。

Mapplet 引用

可以使用 RunMapplet 操作从数据处理器转换中调用 Mapplet。将 RunMapplet 操作添加到数据处理器转换组件之前，必须首先定义对要调用的 Mapplet 的引用。

数据处理器转换设置

在数据处理器转换**设置**视图中配置代码页、XML 处理选项和日志记录设置。

字符编码

字符编码是将一种语言或语言组中的字符映射至十六进制代码。

设计脚本时，需要定义输入文档的编码和输出文档的编码。 定义工作编码以定义 IntelliScript 编辑器显示字符的方式以及数据处理器转换处理字符的方式。

工作编码

工作编码是用于内存中数据的代码页和用于用户界面和工作文件中显示的数据的代码页。 必须选择与数据处理器转换中引用的架构的编码兼容的工作编码。

下表显示了工作编码设置：

设置	说明
使用数据处理器默认代码页	使用数据处理器转换中的默认编码。
其他	从列表中选择编码。
XML 特殊字符编码	确定 XML 特殊字符的表示形式。您可以选择**无**或 XML。 - **无。** 保留为 & < > " ' 将 XML 特殊字符的实体引用解释为文本。 例如，字符 > 将显示为 **>** 默认值为“无”。 - XML. 转换为 & < > " ' 将 XML 特殊字符的实体引用解释为常规字符。 例如，> 将显示为以下字符： >

输入编码

输入编码确定输入文档中字符数据的编码方式。 您可以为脚本中的其他输入端口配置编码。

下表介绍了输入区域中的编码设置：

设置	说明
使用输入文档中指定的编码	使用源文档定义的代码页，如 XML 文档的编码属性。 如果源文档没有编码规范，则数据处理器转换将使用**设置**视图中的编码设置。
使用工作编码	使用与工作编码相同的编码。
其他	从下拉列表中选择输入编码。
XML 特殊字符编码	确定 XML 特殊字符的表示形式。 您可以选择**无**或 XML。 - **无。** 保留为 & < > " ' 将 XML 特殊字符的实体引用解释为文本，例如，字符 > 将显示为 **>** 默认值为“无”。 - XML。 转换为 & < > " ' 将 XML 特殊字符的实体引用解释为常规字符。 例如，> 将显示为以下字符： >
字节顺序	描述多字节字符在输入文档中的显示方式。您可以选择以下选项： - Little-endian。 首先显示最低有效字节。 默认值。 - Big-endian。 首先显示最高有效字节。 - 无二进制转换。

输出编码

输出编码确定字符数据在主要输出文档中的编码方式。

下表介绍了**输出**区域中的编码设置：

设置	说明
使用工作编码	输出编码与工作编码相同。
其他	用户可从列表中选择输出编码。
XML 特殊字符编码	确定 XML 特殊字符的表示形式。您可以选择**无**或 **XML**。 - **无。** 保留为 & < > " ' 将 XML 特殊字符的实体引用解释为文本，例如，字符 **>** 将显示为 **>** 默认值。 - **XML。** 转换为 & < > " ' 将 XML 特殊字符的实体引用解释为常规字符。 例如，> 将显示为以下字符：**>**
与输入编码相同	输出编码与输入编码相同。
字节顺序	描述多字节字符在输入文档中的显示方式。 您可以选择以下选项： - Little-endian。 首先显示最低有效字节。 默认值。 - Big-endian。 首先显示最高有效字节。 - 无二进制转换。

字符编码的规则和准则

配置编码时请使用以下规则和准则：

- 要提高性能，请将工作编码设置为与输出文档相同的编码。
- 将输入编码设置为与输入文档相同的编码。
- 将输出编码设置为与输出文档相同的编码。
- 对于包含多字节字符的语言，请将工作编码设置为 UTF-8。 对于输入和输出编码，可以使用 Unicode 编码（如 UTF-8）或双字节代码页（如 Big5 或 Shift_JIS）。

输出设置

可配置输出控制设置以控制数据处理器转换是否创建事件日志并保存输出文档。

您可以控制数据处理器转换向设计时事件日志写入的消息类型。如果将已解析的输入文档与事件日志一起保存，则可以在**事件**视图中查看出现错误的上下文。

下表介绍了**设计时事件**区域中的设置：

设置	说明
记录设计时事件	确定是否创建设计时事件日志。默认情况下，数据处理器转换会在设计时事件日志中记录通知、警告和故障。可以排除以下事件类型： - 通知 - 警告 - 故障
保存已解析的文档	确定数据处理器转换何时保存已解析输入文档。您可以选择以下选项： - 始终。 - 从不 - 失败时 默认选项为“始终”。

下表介绍了**运行时事件**区域中的设置：

设置	说明
记录运行时事件	确定从映射运行转换时是否创建事件日志。 - 从不。 - 失败时 默认选项为“从不”。

下表介绍了**输出**区域中的设置：

设置	说明
禁用自动输出	确定数据处理器转换是否向标准输出文件写入输出。在下列情况下，请禁用标准输出： - 在转换创建输出文件之前，将解析器的输出传递给其他组件的输入。 - 使用 WriteValue 操作将数据从脚本直接写入输出，而不通过输出端口传递数据。
禁用值压缩	确定数据处理器转换是否使用值压缩优化内存使用。 重要说明：除非 Informatica 全球客户支持部门建议您禁用值压缩，否则请勿执行此操作。

下表介绍了**二进制输出端口收集模式**区域中的设置：对于具有 XML、Avro 或 Parquet 输出的关系到层次结构转换或者具有 Avro 或 Parquet 输出的数据处理器转换解析器，可以为二进制输出选择以下选项之一。

设置	说明
收集多个输入行至单个输出	确定数据处理器转换是否将关系输入累积到单个二进制输出端口。
当输出大小超过以下值时请进行拆分	确定数据处理器转换是否基于指定的输出大小最大值将输出拆分为块。
每行的输出行（请勿收集）	确定数据处理器转换是否以单独的行传递输出。

处理设置

处理设置定义数据处理器转换处理不具有已定义数据类型的元素的方式。这些设置会影响脚本。这些设置不会影响 XMap 所处理的元素。

下表介绍了影响脚本中 XML 处理的处理设置：

设置	说明
视为 xs:string	数据处理器转换将不具有任何类型的元素视为字符串。在**选择 XPath** 对话框中，元素或属性显示为单个节点。
视为 xs:anyType	数据处理器转换将不具有任何类型的元素视为 anyType。在**选择 XPath** 对话框中，元素或属性显示为节点树。如果一个节点为 xs:string 类型，则所有已命名的复杂数据类型均将显示为树节点。

下表描述了影响流转化器处理的处理设置：

设置	说明
流转化器块大小	此设置定义了流转化器每次从输入文件流读取的数据量。数据处理器转换将此设置应用于包含输入文件的流转化器。

下表介绍了影响层次结构到关系转换处理的处理设置：

设置	说明
强制执行严格验证	此设置决定数据处理器转换是否针对层次输入执行严格验证。应用严格验证时，层次结构输入文件必须严格遵守其架构。当数据处理器模式设置为**输出映射**时，可以应用此选项，该模式将为关系输出创建输出端口。 此选项不适用于具有来自低于 10.2.1 版本的 JSON 输入的映射。
规范化 XML 输入	该设置确定数据处理器转换是否规范化 XML 输入。默认情况下，转换针对 XML 输入执行规范化。在某些情况下，您可能选择跳过自动规范化以提高性能。

XMap 设置

XMap 设置定义数据处理器转换如何处理未转换为输出元素的 XMap 输入元素。未读取的元素传递到名为 **XMap_Unread_Input_Values** 的专用端口。只有当 XMap 被选为启动组件时，该设置才会生效。该设置不会影响 XMap 处理的元素。

要将未读取的 XMap 元素传递到专用端口，请启用设置**将未读取的元素写入其他输出端口**。

XML 输出配置

XML 生成设置用于定义 XML 输出文档的特性。

下表介绍了**架构标题**区域中的 XML 生成设置：

设置	说明
架构位置	为主输出文档的根元素定义 schemaLocation 属性。
无命名空间架构位置	为主输出文档的根元素定义 xsi:noNamespaceSchemaLocation 属性。

配置“XML 输出模式”设置以确定数据处理器转换如何处理输入 XML 文档中缺失的元素或属性。 下表介绍了**XML 输出模式**区域中的 XML 生成设置：

设置	说明
保留原样	不添加或删除空元素。默认为已启用。
完整	输出架构中定义的所有必需和可选元素都将写入输出中。 没有内容的元素将作为空元素写入。
压缩	从输出中删除空元素。 如果**为元素添加**处于启用状态，数据处理器转换将仅删除可选元素。 如果**为元素添加**处于禁用状态，数据处理器转换将删除所有空元素。 XML 输出可能无效。

下表介绍了**所需节点的默认值**区域中的 XML 生成设置：

设置	说明
为元素添加	输出架构为某所需元素定义了默认值时，输出中将包括该元素及默认值。默认为已启用。
为属性添加	输出架构为某所需属性定义了默认值时，输出中将包括该属性及其默认值。默认为已启用。
验证添加的值	确定数据处理器转换是否验证完整模式输出添加的空元素。默认为已禁用。 如果**验证添加的值**处于启用状态，且架构不允许包含空元素，则 XML 输出可能无效。

下表介绍了**处理指令**区域中的 XML 生成设置：

设置	说明
添加 XML 处理指令	定义输出文档的字符编码和 XML 版本。 默认为选中状态。
XML 版本	定义 XML 版本。 “XML 版本”设置具有以下选项： - 1.0 - 1.1 默认值为 1.0。

设置	说明
编码	定义在处理指令中指定的字符编码。“编码”设置具有以下选项： - 与输出编码相同。 处理指令中的输出编码与在数据处理器转换设置中定义的输出编码相同。 - 自定义。 定义处理指令中的输出编码。 用户在此字段中键入值。
添加自定义处理指令	将其他处理指令添加到输出文档。 请完全按照输出文档中显示的格式输入处理指令。 默认为“已禁用”。

下表介绍了 **XML 根元素**区域中的 XML 生成设置：

设置	说明
添加 XML 根元素	将根元素添加到输出文档。 在输出文档中多次出现输出架构中定义的根元素时，请使用此选项。 默认为“已禁用”。
根元素名称	为要添加到输出文档的根元素定义名称。

事件

事件是对数据处理器转换中组件处理步骤的记录。 在脚本或库中，每个定位点、操作或转换器均会生成一个事件。 在 XMap 中，每个映射语句均会生成一个事件。

您可以在**数据处理器事件**视图中查看事件。

事件类型

数据处理器转换会在日志文件中写入事件。 每个事件均具有一个事件类型，指示事件是成功还是失败或者事件运行中是否出错。

一个组件可以生成一个或多个事件。 组件是通过还是失败取决于事件是成功还是失败。 如果一个事件失败，则组件也将失败。

下表介绍了数据处理器转换生成的事件类型：

事件类型	说明
通知	正常操作。
警告	数据处理器转换已运行，但出现了意外情况。例如，数据处理器转换多次将数据写入同一个元素。每次该元素被覆盖时，数据处理器转换都将生成一个警告。
失败	数据处理器转换已运行，但组件失败。例如，某必需输入元素为空。
可选故障	数据处理器转换已运行，但可选组件失败。例如，源文档中缺少可选定位点。
致命错误	数据处理器转换因一个严重错误而失败。例如，输入文档不存在。

“数据处理器事件”视图

通过 Developer 工具运行数据处理器转换时，**数据处理器事件**视图将会显示事件。

数据处理器事件视图包含一个**导航**面板和一个**详细信息**面板。 “导航”面板包含导航树。 导航树将列出转换按时间顺序运行的组件。 树中的每个节点下均具有一个表示最严重事件的图标。 在**导航**面板中选择一个节点时，**详细信息**面板中将会显示事件。

导航树包含以下顶级节点：

- 服务初始化。 描述数据处理器转换初始化的文件和变量。
- 执行。 列出脚本、库或 XMap 运行的组件。
- 摘要。 显示与处理相关的统计信息。

运行 XMap 时，导航面板中每个节点名称均包含一个括在方括号中的数字，如 [5]。 要标识为节点生成事件的语句，请在语句网格中右键单击并选择“转到行编号”。 输入节点编号。

运行脚本并双击**导航**面板或**详细信息**面板中的事件时，脚本编辑器会突出显示生成事件的脚本组件。 **数据查看器**视图的**输入**面板将突出显示生成事件的示例源文档部分。

日志

日志包含数据处理器转换记录。 数据处理器转换会将事件写入日志。

数据处理器转换将创建以下类型的日志：

设计时事件日志

设计时事件日志包含在**数据查看器**视图中运行数据处理器转换时发生的事件。 请在**事件**视图中查看设计时日志。

运行时事件日志

运行时事件日志包含在映射中运行数据处理器转换时发生的事件。 您可以在文本编辑器中查看运行时事件日志或将运行时事件日志拖至数据处理器转换的**事件**视图中。

用户日志

用户日志包含在脚本中为组件配置的事件。 在**数据查看器**视图或映射中运行数据处理器转换时，数据处理器转换将写入用户日志。 您可以在文本编辑器中查看用户日志。

设计时事件日志

设计时事件日志包含在 Developer 工具中的**数据查看器**中运行数据处理器转换时发生的事件。

在**数据查看器**视图中运行数据处理器转换时，设计时事件日志将显示在**数据处理器事件**视图中。 默认情况下，设计时事件日志包含通知、警告和故障。 在转换设置中，可以配置数据处理器转换，以从日志中排除一个或多个事件类型。

将输入文档与日志一起保存时，可以通过单击**数据处理器事件**视图中的事件在输入文档中找到生成事件的位置。 配置数据处理器转换设置时，可以选择是每次运行时保存输入文件，还是仅在出现故障时保存输入文件。

设计时事件日志名为 events.cme。 可以在以下目录中找到最后一次运行数据处理器转换的设计时事件日志：

```
C:\<Installation_directory>\clients\DT\CMReports\Init\events.cme
```

每次在**数据查看器**中运行转换时，数据处理器转换均会覆盖设计时事件日志。 如果要在运行转换后查看设计时事件日志或比较多次运行时的日志，请重命名设计时事件日志。 关闭 Developer 工具时，Developer 不会将任何文件保存在

运行时事件日志

运行时事件日志记录在映射中运行数据处理器转换时发生的事件。

如果数据处理器转换运行完时未出现任何故障，则不会写入事件日志。 如果运行中出现故障，则数据处理器转换将再一次运行并在第二次运行过程中写入事件日志。 运行时事件日志名为 events.cme。

在 Windows 计算机上，运行时事件日志位于以下目录中：

C:<Installation_Directory>\clients\DT\CMReports\Tmp\

在 Linux 或 UNIX 计算机上，root 用户的运行时事件日志位于以下目录中：

/root/<Installation_Dirctory>/clients/DT/CMReports/Tmp

在 Linux 或 UNIX 计算机上，可以在以下目录中找到非 root 用户的运行时事件日志：

/home/[UserName]/<Installation_Directory>/DT/CMReports/Tmp

使用配置编辑器可更改运行时事件日志的位置。

在“数据处理器事件”视图中查看事件日志

使用**数据处理器事件**视图可查看设计时事件日志或运行时事件日志。

打开 Windows 资源管理器，然后浏览到您要查看的事件日志文件。 将日志从 Windows 资源管理器窗口拖动到**数据处理器事件**视图。 在**数据处理器事件**视图中右键单击，然后选择**查找**以搜索日志。

注意: 要重新加载最新的设计时事件日志，请右键单击**数据处理器事件**视图，然后选择**重新加载项目事件**。

用户日志

用户日志包含您在脚本中配置的与组件故障有关的自定义消息。

从**数据查看器**视图运行脚本时以及在映射中运行该脚本时，数据处理器转换会将消息写入用户日志。

脚本组件具有 **on_fail** 属性时，可以将其配置为在出现故障时将消息写入用户日志。 在脚本中，将 **on_fail** 属性设置为以下值之一：

- LogInfo
- LogWarning
- LogError

每次运行脚本时都会生成一个新的用户日志。 用户日志文件名包含带有唯一 GUID 的转换名称：

<Transformation_Name>_<GUID>.log

例如 CalculateValue_Aa93a9d14-a01f-442a-b9cb-c9ba5541b538.log

在 Windows 计算机中，您可以在以下目录中查找用户日志：

c:\Users\[UserName]\AppData\Roaming\Informatica\DataTransformation\UserLogs

在 Linux 或 UNIX 计算机中，您可以在以下目录中查找 root 用户的用户日志：

/<Installation_Directory>/DataTransformation/UserLogs

在 Linux 或 UNIX 计算机中，您可以在以下目录中查找非 root 用户的用户日志：

home/<Installation_Dirctory>/DataTransformation/UserLogs

数据处理器转换开发

使用“新建转换”向导自动生成数据处理器转换，或创建空白的数据处理器转换并稍后对其进行配置。如果创建空的数据处理器转换，您必须选择在转换中创建脚本、XMap、库或验证规则对象。脚本可以将源文档解析到层次结构格式、将层次结构格式转换为其他文件格式或将层次结构文档映射到其他层次结构格式。XMap 可将输入层次结构文件转换为其他结构的输出层次结构文件。库可将行业消息传递类型转换为具有层次结构的 XML 文档，或从 XML 转换为行业标准格式。选择定义输入或输出层次结构的架构。

1. 在 Developer tool 中创建转换。
2. 对于空的数据处理器转换，请执行以下额外步骤：
   a. 添加定义了输入或输出 XML 层次结构的架构引用。
   b. 创建脚本、XMap、库或验证规则对象。
3. 配置输入端口和输出端口。
4. 测试该转换。

创建数据处理器转换

在 Developer tool 中创建数据处理器转换。如果创建空的数据处理器转换，则您必须在转换中创建脚本、XMap、库或验证规则对象。或者，您也可以使用“新建转换”向导自动生成数据处理器转换。

1. 在 Developer tool 中，依次单击**文件** > **新建** > **转换**。
2. 选择数据处理器转换，然后单击**下一步**。
3. 输入转换的名称并浏览要放置该转换的模型存储库位置。
4. 选择要使用向导创建数据处理器转换，还是创建空的数据处理器转换。
5. 如果选择创建一个空的数据处理器转换，请单击**完成**。

   Developer tool 将在存储库中创建空的转换。**概览**视图显示在 Developer tool 中。
6. 如果选择使用向导创建数据处理器转换，请执行以下步骤：
   a. 单击**下一步**。
   b. 选择一个输入格式。
   c. 如果特定的输入格式（如 COBOL、或 JSON）需要，请浏览并选择一个架构、复写簿、示例文件或规范文件。
   d. 选择一个输出格式。
   e. 如果输出格式需要，请浏览并选择一个架构、复写簿、示例文件或规范文件。
   f. 单击**完成**。向导将在存储库中创建该转换。

   转换可能包含解析器、序列化程序、映射程序或具有通用组件的对象。如果选择架构、复写簿、示例文件或规范文件，向导还会在存储库中创建相当于文件中的层次结构的架构。

选择架构对象

选择用于为您计划创建的每个 XMap 或脚本组件定义输入或输出层次结构的架构对象。

您可以在“引用”视图中添加架构引用，也可以在创建脚本或 XMap 对象时添加架构引用。模型存储库中必须存在架构对象，才能在脚本或 XMap 中进行引用。

1. 在数据处理器转换的**引用**视图中，单击**添加**。
2. 如果模型存储库中存在架构对象，则可以浏览并选择该架构。
3. 如果模型存储库中不存在架构，请单击**创建新架构对象**，并从层次结构架构文件中导入一个架构对象。

4. 单击“完成”以将架构引用添加到数据处理器转换中。

在空的数据处理器转换中创建对象

在数据处理器转换的**对象**视图中创建一个脚本、库、XMap 或验证规则对象。创建对象后，可以从**对象**视图中打开该对象以对其进行配置。

创建脚本

创建脚本对象并定义要创建的脚本组件的类型。 您还可以定义架构引用和示例源文件。

1. 在数据处理器转换的**对象**视图中，单击**新建**。
2. 输入脚本的名称，然后单击**下一步**。
3. 选择相应选项以创建解析器或序列化程序。 选择“其他”可创建映射程序、转换器或流转化器组件。
4. 输入组件的名称。
5. 如果该组件是在转换中处理数据的第一个组件，请启用**设置为启动组件**。
6. 如果要为该脚本输入架构引用，请单击**下一步**。 如果不想输入架构引用，请单击**完成**。
7. 如果选择了创建架构引用，请选择**添加对架构对象的引用**，然后在模型存储库中浏览架构对象。 单击**创建新架构对象**可在模型存储库中创建架构对象。
8. 单击**下一步**以输入示例源引用或示例文本。 如果不想定义示例源，请单击**完成**。

   使用示例源定义示例数据并测试脚本。
9. 如果选择了示例源，请选择**文件**并浏览示例文件。

   您也可以在**文本**区域中输入示例文本。 开发程序工具将使用该文本以测试脚本。
10. 单击**完成**。

    此时，**脚本**视图将显示在 Developer 工具编辑器中。

创建 XMap

在 Data Transformation 的**对象**视图上创建 XMap。创建 XMap 时，必须具有描述输入和输出层次结构文档的架构。选择架构中作为输入层次结构的根元素的元素。

1. 在数据处理器转换的**对象**视图中，单击**新建**。
2. 选择 XMap，然后单击**下一步**。
3. 输入 XMap 的名称。
4. 如果 XMap 组件是第一个在转换中处理数据的组件，请启用**设置为启动组件**。

   单击**下一步**。
5. 如果选择创建架构引用，请选择**添加对架构对象的引用**，并浏览模型存储库中的架构对象。

   要导入新的架构对象，请单击**创建新的架构对象**。
6. 如果您具有可用于测试 XMap 的示例层次结构文件，请从文件系统中浏览并选择该文件。

   您可以更改示例层次结构文件。
7. 选择输入层次结构的根。

   在**根元素选择**对话框中，选择架构中作为输入层次结构的根元素的元素。可以搜索架构中的元素。可以使用模式搜索。输入 *<string> 可匹配字符串中任意数量的字符。输入 ?<character> 可匹配单个字符。

8. 单击**完成**。

   Developer 工具会为您创建的每个 XMap 创建一个视图。单击该视图可配置映射。

创建库

在 Data Transformation 的**对象**视图上创建一个库对象。 选择消息类型、组件和名称。或者，您可以定义一个示例消息类型源文件用于测试库对象。

在数据处理器转换中创建库之前，请在计算机上安装库软件包。

1. 在数据处理器转换的**对象**视图中，单击**新建**。
2. 选择库，然后单击**下一步**。
3. 浏览并选择消息类型。
4. 选择创建解析器或序列化程序。

   如果库对象输入为消息类型而输出为 XML，则创建解析器。如果库对象输入为 XML 而输出为消息类型，则创建序列化程序。
5. 如果该库是在数据处理器转换中处理数据的第一个组件，请启用**设置为启动组件**。

   单击**下一步**。
6. 如果有可用于与库配合测试的示例消息类型源文件，请从文件系统中浏览并选择该文件。

   可以更改示例文件。
7. 单击**完成**。

   开发程序工具会为您创建的每个消息类型创建一个视图。 单击该视图可访问映射。

创建验证规则

在数据处理器转换**对象**视图中创建验证规则对象。

1. 在数据处理器转换的**对象**视图中，单击**新建**。
2. 选择验证规则，然后单击**下一步**。
3. 输入验证规则的名称。
4. 如果您具有可用于测试验证规则的示例 XML 文件，请从文件系统中浏览并选择该文件。

   可以更改示例 XML 文件。
5. 单击**完成**。

   Developer 工具创建验证规则对象并在验证规则编辑器中打开它。

添加示例源

选择示例源以测试计划创建的脚本、XMap、库或验证规则。

可以在创建脚本、XMap、库或验证规则时添加示例源。一旦选择，示例源会添加到模型存储库中。由于模型存储库限制，示例源文件大小被限制为 5 MB。

您可以更改示例源。

创建端口

在**概览**视图中配置输入和输出端口。

在脚本中配置附加的输入或输出端口时，开发程序工具在默认情况下会将这些输入端口和输出端口添加到转换中。 在**概览**视图中不能添加输入端口。

1. 如果希望返回输出数据的行数而非 XML，请启用**关系输出**。

   启用“关系输出”时，Developer 工具将删除默认输出端口。
2. 选择输入端口数据类型、端口类型、精度和小数位数。
3. 如果未定义关系输出端口，请定义输出端口数据类型、端口类型、精度和小数位数。
4. 如果脚本具有关系输入端口，则可以为端口定义示例输入文件的位置。 单击**输入位置**字段中的**打开**按钮以浏览文件。
5. 如果启用了“关系输出”，请单击**输出映射**以创建输出端口。
6. 在“端口”视图中，将节点从**层次结构输出**区域映射到**关系端口**区域中的字段。

测试转换

在**数据查看器**视图中测试数据处理器转换。

测试转换之前，请确认您已定义启动组件。 您可以在脚本中定义启动组件，或者可以在**概览**选项卡中选择启动组件。 您还需要选择要进行测试的示例输入文件。

1. 打开**数据查看器**视图。
2. 单击**运行**。

   Developer tool 将验证转换。 如果没有错误，Developer tool 将在**输入**区域中显示示例文件。 输出结果在“输出”面板中显示。
3. 单击**显示事件**以显示**数据处理器事件**视图。
4. 双击**数据处理器事件**视图中的某个事件以在脚本编辑器中调试该事件。
5. 测试多个组件时，如果每个组件都具有不同的示例输入文件，请单击**与编辑器同步**以更改输入文件。

   如果您在文件系统中修改了示例文件内容，则所做的更改会显示在**输入**区域中。

数据处理器转换导出和导入

您可以将数据处理器转换作为服务导出并从 Data Transformation 存储库运行该服务。还可将 Data Transformation 服务导入 Developer tool 中。导入 Data Transformation 服务后，Developer tool 将通过该服务创建数据处理器转换。

注意: 将 Data Transformation 服务导入模型存储库后，Developer tool 会将关联的架构导入该存储库。如果修改存储库中的架构，则更改有时不会显示在转换架构引用中。您可以关闭并打开模型存储库连接，或者关闭并打开 Developer tool，以使架构更改显示在转换中。

将数据处理器转换作为服务导出

您可以将数据处理器转换作为 Data Transformation 服务导出。可将服务导出到要在其中运行该服务的计算机的文件系统存储库中。可以使用 PowerCenter、用户定义的应用程序或 Data Transformation CM_console 命令运行服务。

1. 在 **对象浏览器** 视图中，右键单击要导出的数据处理器转换，然后选择**导出**。

   此时将显示**导出**对话框。
2. 选择 **Informatica** > **导出数据处理器转换**，然后单击**下一步**。

   此时将显示**选择**页。
3. 单击**下一步**。

   此时将显示**选择服务名称和目标文件夹**页。
4. 选择目标文件夹：
   - 要在承载 Developer 工具的计算机上导出服务，请单击**服务文件夹**。
   - 要在其他计算机上部署服务，请单击**文件夹**。浏览到要在其中部署服务的计算机上的 \ServiceDB 目录。
5. 单击**完成**。

导入多个 Data Transformation 服务

可以从保存目录的计算机导入 Data Transformation 服务的目录。将 Data Transformation 服务导入 Developer 模型存储库时，Developer tool 会将转换、架构和示例数据与.cmw 文件一起导入。如果需要导入许多服务，请导入服务目录，而不是一次导入一个服务。

1. 单击**文件** > **导入**。

   此时将显示**导入**对话框。
2. 选择 **Informatica 导入 Data Transformation 服务 (文件夹)**，然后单击**下一步**。

   此时将显示**导入 Data Transformation 服务**页。
3. 浏览到要导入的目录。
4. 浏览到存储库中要用来保存转换的位置，然后单击**完成**。

   Developer tool 会将转换、架构和示例数据与 .cmw 文件一起导入。

导入 Data Transformation 服务

您可以将 Data Transformation 服务的 .cmw 文件导入模型存储库，以创建数据处理器转换。Developer tool 会将转换、架构和示例数据与 .cmw 文件一起导入。

1. 单击**文件** > **导入**。

   此时将显示**导入**对话框。
2. 选择 **Informatica 导入 Data Transformation 服务 (单一)**，然后单击**下一步**。

   此时将显示**导入 Data Transformation 服务**页。
3. 浏览到要导入的服务 .cmw 文件。

   Developer tool 将根据服务文件名称命名转换。可以更改此名称。
4. 浏览到存储库中希望保存转换的位置，然后单击**完成**。

   Developer tool 会将转换、架构和示例数据与 .cmw 文件一起导入。
5. 要编辑转换，请在**对象浏览器**视图中双击转换。

使用数据处理器转换将映射导出至 PowerCenter

使用数据处理器转换将映射导出到 PowerCenter 时，可以将对象导出到本地文件，然后将映射导入到 PowerCenter。 或者，您也可以直接将映射导出到 PowerCenter 存储库。

1. 在**对象浏览器**视图中，选择要导出的映射。 右键单击，然后选择**导出**。

   此时将显示**导出**对话框。
2. 选择 **Informatica** > **PowerCenter**。
3. 单击**下一步**。

   此时将显示**导出到 PowerCenter** 对话框。
4. 选择项目。
5. 选择 PowerCenter 版本。
6. 选择导出位置（PowerCenter 导入 XML 文件或 PowerCenter 存储库）。
7. 指定导出选项。
8. 单击**下一步**。

   开发程序工具将提示您选择要导出的对象。
9. 选择要导出的对象，然后单击**完成**。

   开发程序工具会将对象导出到选定的位置。如果将映射导出到某个位置，开发程序工具也会将映射中的数据处理器转换作为服务导出到指定位置的某个文件夹中。
10. 如果将映射导出到 PowerCenter 存储库，服务将导出到以下目录路径中：`%temp%\DTServiceExport2PC\`

    导出功能将为每个服务创建一个下列名称的单独文件夹：<日期><服务完整名称>

    如果转换包含关系映射，则将为关系到层次结构映射创建一个文件夹，并且为层次结构到关系映射创建单独文件夹。
11. 使用数据处理器转换服务从导出文件的本地位置将一个或多个文件夹复制到 PowerCenter ServiceDB 文件夹。
12. 如果将映射导出到 PowerCenter 导入 XML 文件，请将映射导入到 PowerCenter。 有关如何将对象导入到 PowerCenter 的详细信息，请参阅*《PowerCenter 9.6.0 存储库指南》*。

数据处理器转换验证

将数据处理器转换导出为服务后，可以从数据转换存储库对服务运行 VRL 验证。

可以使用速度增强的数据转换引擎进行 VRL 验证。速度增强的数据转换引擎支持以下 VRL 功能：

- `dt:exist`
- `dt:empty`
- `dt:date-valid`
- `dt:next-sequence`
- `dt:all-equal`
- `dt:lookup`
- `dt:regex-match`

速度增强的数据转换引擎生成输出 **ValidateValue**。**ValidateValue** 包含 `max_error_count` 属性，默认为 200 个错误。当错误数超过 `max_error_count` 时，验证将停止。

注意: 速度增强数据转换引擎 VRL 语法不支持 <list> 标记。

使用速度增强的数据转换引擎进行 VRL 验证

导出数据转换服务后，可以使用速度增强的数据转换引擎对该服务进行 VRL 验证。

- 在服务 .cmw 文件中设置以下标志 optimize_vrl。

  将 optimize_vrl 标志添加到 ServiceConfigProf 实例，如以下示例所示：

  ```
  instance ServiceConfig = ServiceConfigProf<add_required_xml_elements, add_required_xml_attributes,
  optimize_vrl>
  ```

数据处理器转换在非本地环境中

非本地环境中的数据处理器转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。*
- Databricks Spark 引擎。不受支持。

** 有关 Spark 引擎上数据处理器转换支持的信息，请参阅* KB article。

第 14 章

判定转换

本章包括以下主题：

判定转换概览

判定转换属于被动转换，它会对输入数据中的条件进行计算，然后基于这些条件的结果创建输出。

配置判定转换，以便根据输入字段中的值来生成不同的值。 例如，如果客户收入大于特定金额，则可以向该客户名称中添加字符串“优先级”。

可以向判定转换中添加多个判定策略。 每个策略会计算一个 IF-THEN-ELSE 条件语句。 在该语句中，您可以使用 ELSEIF 条件或嵌套其他 IF-THEN-ELSE 语句。

判定转换类似于表达式转换，因为它允许您使用条件语句和函数来测试源数据。 但是，在以下方面判定转换与表达式转换有所不同：

- 判定转换使用 IF-THEN-ELSE 语句计算条件。 表达式转换使用 IIF 语句。
- 判定转换包含一些无法在表达式转换中使用的函数。
- 每个判定策略均可以生成多个输出。

判定转换函数

通过判定转换可以访问用于定义判定策略的预定义函数。

判定转换表达式编辑器包含一个 Decision 文件夹。 此文件夹包含判定转换专用的函数。 该编辑器还包含其他文件夹，通过这些文件夹，而可以访问表达式转换函数。

在表达式编辑器中单击某个函数时，转换不仅会显示该函数的功能说明，还会显示该函数的用法和数据类型。

注意: 并非所有表达式转换函数都与判定转换兼容。 通过判定转换只能访问兼容的表达式转换函数。

判定转换函数列表

- ABS
- ADD_TO_DATE
- ASCII
- CEIL
- CHOOSE
- CHR
- CHRCODE
- CONCAT
- CONTAINS
- CONVERT_BASE
- COS
- COSH
- CRC32
- CUME
- CURDATE
- CURTIME
- DATE_COMPARE
- DATE_DIFF
- DATECONVERT
- EXP
- FLOOR
- FV
- GET_DATE_PART
- GREATEST
- IN
- INDEXOF
- INITCAP
- INSTR
- IS_DATE
- IS_NUMBER
- ISNULL
- LAST_DAY
- LEAST
- LEFTSTR
- LENGTH

- LN
- LOG
- LOWER
- LPAD
- LTRIM
- MAKE_DATE_TIME
- MAX
- MD5
- METAPHONE
- MIN
- MOD
- MONTHCOMPARE
- MOVINGAVG
- MOVINGSUM
- NPER
- PMT
- POWER
- PV
- RAND
- RATE
- REG_EXTRACT
- REG_MATCH
- REG_REPLACE
- REPLACECHR
- REPLACESTR
- REVERSE
- RIGHTSTR
- ROUND
- RPAD
- RTRIM
- SET_DATE_PART
- SIGN
- SIN
- SINH
- SOUNDEX
- SQRT
- SUBSTR
- TAN

- TANH
- TIMECOMPARE
- TO_CHAR
- TO_DATE
- TO_FLOAT
- TO_INTEGER
- TRUNC
- UPPER
- XOR

注意: 使用常量值在判定转换的 CURDATE、DATE_COMPARE、DATECONVERT 和 MONTHCOMPARE 函数中定义日期格式。

判定转换条件语句

判定转换使用 IF-THEN-ELSE 条件语句计算输入数据。

在这些条件语句中，可以使用 ELSEIF 条件或嵌套其他 IF-THEN-ELSE 语句。 判定转换条件语句使用以下格式：

```
// Primary condition
IF  <Boolean expression>
THEN <Rule Block>
// Optional - Multiple ELSEIF conditions
ELSEIF  <Boolean expression>
THEN  <Rule Block>
// Optional ELSE condition
ELSE  <Rule Block>
ENDIF
```

您可以在规则块内嵌套其他条件语句。

判定转换运算符

使用判定转换运算符定义判定策略。

下表介绍了判定转换运算符：

运算符类型	运算符	说明
分配	:=	向端口分配值。
Boolean	AND	添加所需的逻辑条件。 要使父布尔表达式为 true，使用此运算符链接的所有逻辑条件都必须为 true。
Boolean	OR	添加逻辑条件。 要使父布尔表达式为 true，使用此运算符链接的逻辑条件中至少有一个必须为 true。

运算符类型	运算符	说明
Boolean	NOT	指定负逻辑条件。 要使父布尔表达式为 true，使用此运算符指定的负条件必须为 true。
判定	=	测试比较项是否相等。 与 String 或 Numeric 数据类型一起使用。
判定	<>	测试比较项是否不相等。 与 String 或 Numeric 数据类型一起使用。
判定	<	测试一个值是否小于另一个值。 与 Numeric 数据类型一起使用。
判定	<=	测试一个值是否小于或等于另一个值。 与 Numeric 数据类型一起使用。
判定	>	测试一个值是否大于另一个值。 与 Numeric 数据类型一起使用。
判定	>=	测试一个值是否大于或等于另一个值。 与 Numeric 数据类型一起使用。
数值	-	减法
数值	NEG	求反
数值	+	加法
数值	*	乘法
数值	/	除法
数值	%	取模。 返回由一个数除以另一个数之后所得的余数。
String	||	连接字符串。

判定转换空值处理

空值处理可确定数据集成服务在判定转换中如何处理空值数据。

启用空值处理后，判定转换会保留空值输入数据的原始格式。 转换会使用空输入值计算函数。

禁用空值处理后，判定转换会向空输入数据分配一个默认值。 转换会使用默认值计算函数。 例如，如果某个 Integer 类型输入字段具有空值，则判定转换会为该输入分配值 0，然后使用输入值 0 来计算函数。

默认情况下，在判定转换中不启用空值处理。 可以在**策略**选项卡上启用空值处理。 可以在为转换配置策略后启用空值处理。

配置判定策略

要配置判定策略，请将源数据连接到判定转换，然后在转换视图中编辑属性。

1. 打开判定转换。
2. 验证转换是否包含输入端口和输出端口。

3. 选择**判定**视图。
4. 单击**添加**。
5. 输入策略名称。
6. 在**表达式**区域中，输入 IF-THEN-ELSE 条件语句。
7. 要向表达式添加函数，请在**函数**选项卡中浏览函数，然后双击函数名称。

   提示: 要快速输入函数，请先键入函数名称的首字母，然后选择 CTRL-Space。
8. 要向表达式添加端口，请在**端口**选项卡中浏览端口。 双击端口名称可将其添加到表达式中。 或者，单击**编辑输出端口**编辑输出端口设置或添加输出端口。
9. 或者，通过键入“//”并键入注释添加注释行。
10. 单击**验证**可确定判定表达式是否有效。
11. 单击**确定**保存策略。
12. 或者，添加其他策略。 每个策略都必须使用唯一的输出端口。 策略无法共享输出端口。

判定转换高级属性

配置有助于确定数据集成服务为判定转换处理数据的方式的属性。

为判定转换配置以下高级属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

可分区

可以使用多个线程处理转换。如果希望数据集成服务使用一个线程来处理转换，请清除此选项。 数据集成服务可以使用多个线程处理余下的映射管道阶段。

如果判定转换使用 CUME、MOVINGSUM 或 MOVINGAVG 等数值函数之一，则建议您为该转换禁用分区。这些函数逐行计算累加值和平均值。使用 CUME、MOVINGSUM 或 MOVINGAVG 函数的已分区转换在每次映射运行时返回的计算结果可能有所不同。

如果转换未使用 CUME、MOVINGSUM 或 MOVINGAVG 函数，则您可为该转换启用分区以优化性能。

非本地环境中的判定转换

非本地环境中的判定转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。
- Databricks Spark 引擎。有限支持。

Spark 引擎上的判定转换

支持判定转换，但存在以下限制：

- 转换属性必须指定该转换是可分区的。

Databricks Spark 引擎上的判定转换

支持判定转换，但存在以下限制：

- 转换属性必须指定该转换是可分区的。

第 15 章

重复记录异常转换

本章包括以下主题：

重复记录异常转换概览

重复记录异常转换是一种主动转换，可用于读取数据质量处理的输出并标识需要手动查看的重复记录。重复记录异常转换属于多组转换。

重复记录异常转换从其他转换或其他映射中的数据对象接收输入。异常转换的输入必须包含数值匹配得分，转换可使用该得分确定记录是否重复。在重复记录异常转换中设置匹配得分阈值的上界和下界。

重复记录异常转换执行以下操作之一：

- 如果匹配得分大于或等于阈值上界，则转换会将该记录视为重复记录并将其写入数据库目标。
- 如果匹配得分小于阈值上界且大于阈值下界，则转换会将该记录视为可能重复的记录并将其写入其他目标供手动查看。如果记录属于某一群集，则转换会将该群集中的所有记录写入目标。
- 如果群集中的任意匹配得分都小于阈值下界，则群集中的所有记录将进入唯一记录输出组。大小为 1 的群集将路由至唯一组，不管其匹配得分为何。默认情况下，异常转换不会将唯一记录写入目标。您可以配置转换以返回唯一记录。
- 如果群集中的任意匹配得分都不在 0 - 100 范围内，异常转换将忽略群集中的所有行。数据集成服务会记录一条包含 clusterID 的消息。

重复记录异常进程流

重复记录异常转换分析其他数据质量转换的输出，并使用不同级别的数据质量创建包含记录的表。

可以在单个映射中配置数据质量转换，也可以在进程的不同阶段创建映射。

可以使用 Analyst 工具查看和更新需要手动查看的重复记录。

使用 Developer 工具执行以下任务：

1. 创建一个映射为数据质量问题生成得分值。
2. 在群集模式下使用匹配转换以生成重复记录异常的得分值。
3. 配置重复记录异常转换以读取匹配转换输出。配置转换以基于记录中的匹配得分值将记录写入数据库表。
4. 为自动合并记录配置目标数据对象。
5. 单击**生成重复记录表**选项创建重复记录表并将其添加到映射画布。
6. 将映射添加到工作流中。
7. 配置人工任务以将可能重复记录的手动查看分配给用户。用户可在 Analyst 工具中查看和更新记录。

重复记录异常

可以使用重复记录异常转换标识需要手动查看的重复数据的群集。 群集中的记录匹配得分确定潜在重复项。 可以配置转换中匹配得分的阈值上界和下界。 阈值上界和下界定义相似度。

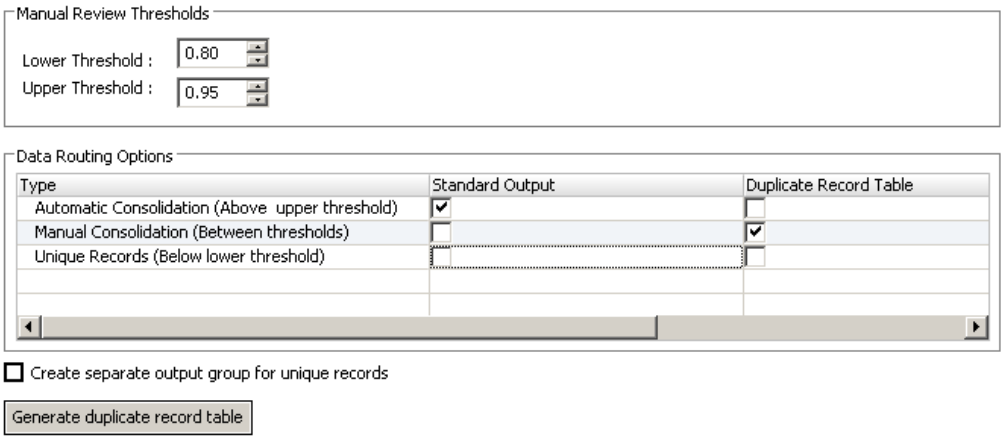
群集包含匹配操作分组在一起的相关记录。 匹配转换使用重复项分析操作和身份识别操作创建群集。 群集中的每个记录具有相同的群集 ID。 当群集中最低匹配得分介于阈值上界和下界之间时，重复记录异常转换会将群集标识为重复记录异常群集。 匹配转换将群集 ID 值列添加到所有记录。 重复记录收到相同的群集 ID。

群集中得分最低的记录确定了群集类型。 群集中可能有 11 个记录匹配得分为 0.95 ，1 个记录匹配得分为 0.79。如果阈值上界为 0.9，阈值下界为 0.8，则异常转换会将记录写入唯一记录表。

重复记录异常配置视图

定义匹配得分阈值并配置重复记录异常转换写入不同类型输出数据的位置。

下图显示了可以配置的属性：



可以配置以下属性：

阈值下界

重复记录得分范围的下限。转换会将匹配得分小于该值的记录处理为唯一记录。阈值下界值介于 0 到 1 之间。

阈值上界

重复记录得分范围的上限。转换会将匹配得分记录大于或等于阈值上界的记录处理为重复记录。阈值上界值大于阈值下界值。

自动合并

群集中所有记录的匹配得分都大于阈值上界。自动合并群集不需要查看。记录重复。可以使用合并转换合并记录。默认情况下，重复记录异常转换会将自动合并群集写入标准输出端口。

手动合并

群集中所有记录的匹配得分都大于或等于阈值下界且至少有一个记录的匹配得分小于阈值上界。必须手动查看群集以确定群集中是否包含重复记录。默认情况下，重复记录异常转换会将手动合并记录写入重复记录表。

唯一合并

群集大小等于 1 或群集中的任何记录匹配得分都小于阈值下界。唯一记录群集不是重复群集。默认情况下，重复记录异常转换不会将唯一记录写入输出表。

标准输出

转换写入标准输出端口的记录类型。

默认为自动合并记录。

重复记录表

转换写入重复记录输出端口的记录类型。默认为手动合并记录。

为唯一记录创建单独的输出组

为唯一记录创建单独的输出组。如果未为唯一记录创建单独的表，则可以将转换配置为将唯一记录写入其他组的其中一个。或者，可以跳过将唯一记录写入输出表。默认为禁用。

生成重复记录表

创建数据库对象以包含重复记录群集数据。选择此选项时，Developer tool 会创建数据库对象。Developer tool 将对象添加到模型存储库，将对象的实例添加到映射画布，并将端口链接至对象。

生成重复记录表

可以从映射中的重复记录异常转换实例生成重复记录表。

1. 在**配置**视图中，单击**生成重复记录表**以生成表。
   此时将显示**创建关系数据对象**对话框。
2. 浏览并选择与数据库的连接，以包含该表。
3. 在数据库中输入重复记录表的名称。
4. 在模型存储库中输入重复记录表对象的名称。
5. 单击**完成**。
   Developer 工具将新表添加到映射画布。

端口

重复记录异常转换具有多个输入和输出端口组。

下图显示了输入和输出端口示例：

Ports

	Name	Type	Precision	Scale	Input	Output	Default	Description
	Inputs							
	Data (3)							
1	Employee	decimal	10	0	☑	☐		
2	Name	string	50	0	☑	☐		
3	Addr1	string	50	0	☑	☐		
	Control (3)							
1	Score	double	15	0	☑	☐		
2	Row_Identifier	string	25	0	☑	☐		
3	Cluster_ID	integer	10	0	☑	☐		
	Outputs							
	Standard Output (6)							
1	Score	double	15	0	☐	☑		
2	Row_Identifier	string	25	0	☐	☑		
3	Cluster_ID	integer	10	0	☐	☑		
4	Employee	decimal	10	0	☐	☑		
5	Name	string	50	0	☐	☑		
6	Addr1	string	50	0	☐	☑		
	Cluster Data (9)							
1	Row_Identifier	bigint	19	0	☐	☑		
2	Sequential_Cl...	bigint	19	0	☐	☑		
3	Cluster_ID	integer	10	0	☐	☑		

重复记录异常转换输入端口

重复记录异常转换具有数据组和控件组输入端口。

数据组包含接收源数据的用户定义端口。

控件端口接收匹配转换添加到源数据的元数据。 下表介绍了**控件**端口：

端口	说明
得分	介于 0 和 1 之间的小数值。 通过链接到群集的记录标识相似度。
Row_Identifier	记录的唯一标识符。
Cluster_ID	记录所属的匹配群集的 ID。

重复记录异常转换输出端口

重复记录异常转换具有多个输出组。 默认情况下，转换将重复记录写入标准输出组。 转换将潜在匹配项写入群集数据组。 可以为唯一记录添加输出组。

可以通过更改配置视图上的默认设置更改转换写入输出端口的记录类型。

下表介绍了标准输出组的输出端口：

端口	说明
得分	介于 0 和 1 之间的小数值。 确定群集中一个记录和另一记录之间的相似度。
Row_Identifier	记录的唯一标识符。
Cluster_ID	匹配转换向记录分配的群集 ID。
用户定义的端口	源数据字段。

下表介绍了群集数据组中的输出端口：

端口	说明
Row_Identifier	记录的唯一标识符。
Sequential_Cluster_ID	确定人工任务中的群集。 工作流使用连续群集 ID 将群集分配给人工任务的实例。
Cluster_ID	确定记录所属的群集。 匹配转换向所有记录分配群集 ID。
得分	介于 0 和 1 之间的小数值。 通过链接到群集的记录标识相似度。
Is_Master	指示记录是否为群集中首选记录的字符串值。 默认情况下，群集中的第一行是首选记录。 值为 Y 或 N。
Workflow_ID	确定任务中记录工作流的 ID。 在工作流外运行映射时，工作流 ID 为 DummyWorkflowID。
用户定义的端口	源数据端口。

创建端口

向数据组添加每个输入端口。 添加输入端口时，Developer 工具向标准输出组、群集数据组和唯一记录组中添加具有同名的输出端口。

1. 选择数据输入组。

   将突出显示该组。

2. 单击**新建(插入)**。

   Developer 工具向数据组、标准输出组、群集数据组和唯一记录组中添加一个字段。

3. 根据需要更改字段的名称。

   Developer 工具更改其他组的字段名称。

4. 输入需要为数据源添加的其余端口。

重复记录异常转换高级属性

重复记录异常转换包含确定排序行为、高速缓存行为以及跟踪级别的高级属性。

您可以配置以下高级属性：

排序

确定转换是否根据**群集 ID** 端口数据对输入行排序。默认情况下启用此属性。

如果输入行未预先排序，请选择此属性。

缓存文件目录

指定数据集成服务将当前转换的临时数据写入到的目录。当输入数据量超出可用系统内存时，数据集成服务会将临时文件写入该目录。数据集成服务会在运行映射后删除这些临时文件。

您可以在该属性上输入目录路径，也可以使用参数标识目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。默认值为 CacheDir 系统参数。

缓存文件大小

确定数据集成服务对转换的输入数据排序使用的系统内存量。默认值为 400,000 字节。

在对数据排序之前，数据集成服务会分配您指定的内存量。如果排序操作生成的数据较多，数据集成服务会将超出的数据写入缓存文件目录。如果排序操作所需的内存超出系统内存和文件存储可以提供的内存，映射会失败。

注意: 如果输入 65536 或更高的值，转换将以字节为单位读取值。如果输入低一些的值，转换将以 MB 为单位读取值。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

重复记录异常映射示例

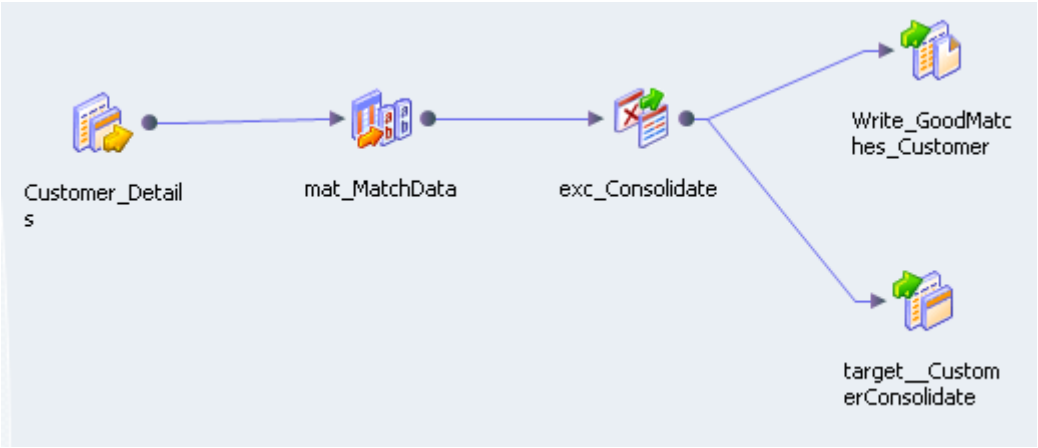
组织运行数据项目以查看客户数据。组织确定客户数据包含多个重复记录。组织需要手动查看可能重复的某些记录。

创建数据质量映射以确定重复的客户记录。映射包含匹配转换。重复记录异常转换从匹配转换接收结果。重复记录异常转换将状态不确定的每个记录群集写入数据库表。数据分析师在 Analyst 工具中查看数据，并确定哪些记录属于重复记录。

重复记录异常映射

配置检查客户记录并查找重复记录的重复记录异常映射。

下图显示了重复记录异常映射：



映射包含以下对象：

Customer_Details

可能包含重复记录的数据源。

mat_MatchData

匹配转换检查客户数据以确定记录是否匹配。匹配转换创建表示两个列值之间相似度的数值得分。通过某种算法计算匹配得分，并以一个介于 0 到 1 之间的小数值表示。如果两列值完全相同，算法将分配得分 1。

exc_Consolidate

重复记录异常转换确定哪些记录是可能重复的客户、哪些是已知重复客户或者哪些是唯一客户记录。

Write_GoodMatches_Customer 表

该表接收不需要手动查看的所有记录。重复记录异常转换将重复记录和唯一记录写入此表。

Target_CustomerConsolidate 表

异常转换将可能重复的记录写入 Target_CustomerConsolidate 表。此表中的记录需要在 Analyst 工具中手动查看。

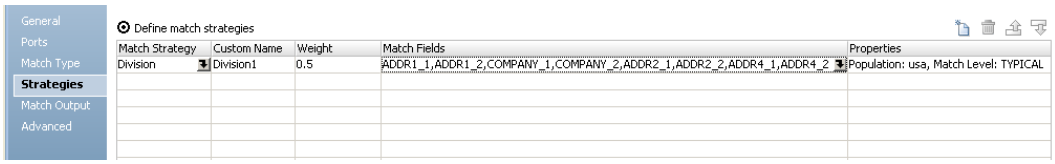
匹配转换

匹配转换接收客户数据并执行身份匹配。

为“群集 - 全部匹配”输出类型配置匹配转换。 匹配转换返回群集中的匹配记录。 群集中的每个记录必须至少与群集中得分大于或等于匹配阈值的一个其他记录匹配。 匹配阈值是 0.75。

选择匹配转换**策略**选项卡上的部门匹配策略。 部门策略是基于地址字段标识组织的预定义匹配策略。 在匹配转换**策略**选项卡上，选择输入端口以检查匹配项。 将策略权重配置为 0.5。

下图显示了匹配转换的部门策略配置：



匹配转换向每个输出记录添加群集信息。 转换还向每个记录添加唯一的 RowID。

重复记录异常输入组

重复记录异常转换具有两个输入组。 转换具有一个接收客户数据的数据组。 转换具有包含行、行标识符以及群集 ID 匹配得分的控件组。

下图显示了异常转换中的输入组：

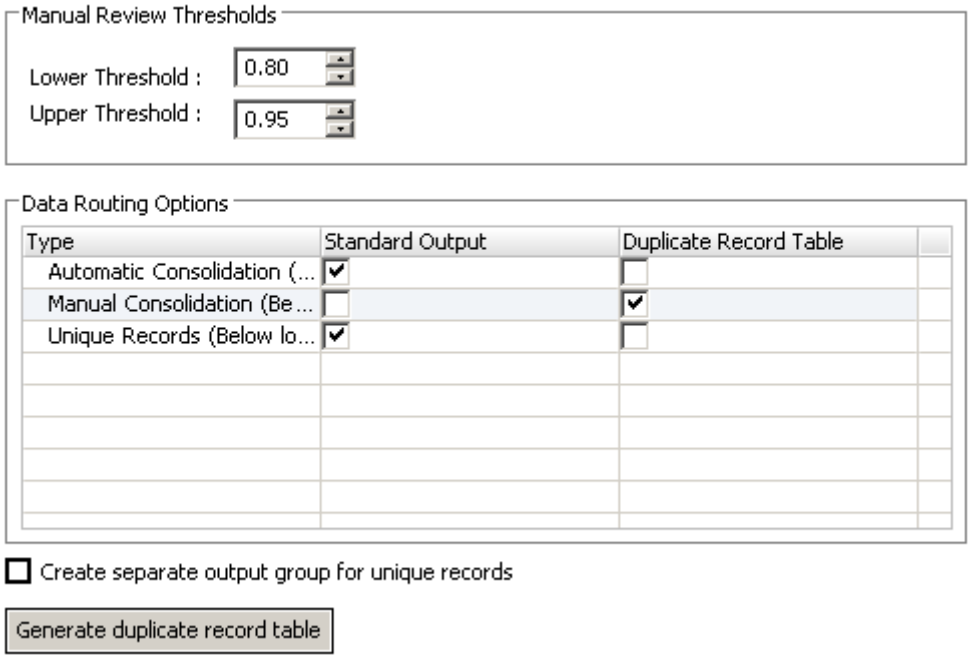
	Name	Type	Precision	Scale	Input	Output	Default	Description
	Inputs							
	Data (11)							
1	CUST_ID	decimal	20	0	☑	☐		
2	COMPANY	string	200	0	☑	☐		
3	CONTACT	string	200	0	☑	☐		
4	TITLE	string	200	0	☑	☐		
5	ADDR1	string	200	0	☑	☐		
6	ADDR2	string	100	0	☑	☐		
7	ADDR3	string	100	0	☑	☐		
8	ADDR4	string	50	0	☑	☐		
9	COUNTRY	string	50	0	☑	☐		
10	PHONE	string	100	0	☑	☐		
11	EMAIL	string	100	0	☑	☐		
	Control (3)							
1	Score	double	15	0	☑	☐		
2	Row_Identifier	string	25	0	☑	☐		
3	Cluster_ID	integer	10	0	☑	☐		

数据组包含客户数据。 客户数据包括客户 ID、联系人、职称以及地址字段。 控件组是匹配转换为每个客户记录添加的其他元数据。 控件组包含匹配得分、rowID 以及群集 ID。

重复记录异常示例配置视图

在**配置**视图上定义阈值上界和下界。 标识转换写入重复客户记录、可能重复的记录以及唯一客户记录的位置。

下图显示了重复的记录异常转换**配置**视图：



下表介绍了配置设置：

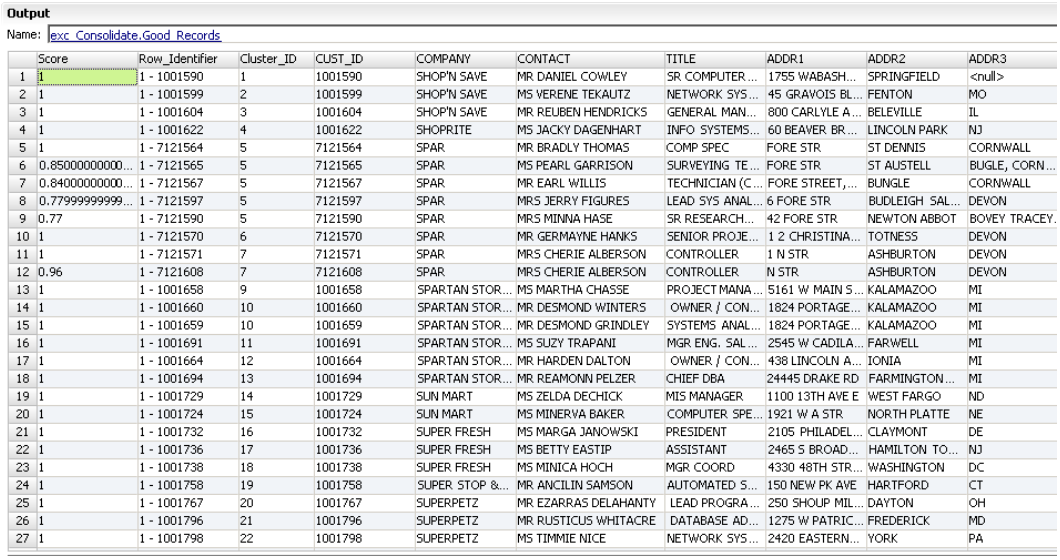
选项	设置
阈值下界	.80
阈值上界	.95
自动合并	标准输出表
手动合并	重复记录表
唯一记录	标准输出表

单击**生成重复记录表**创建重复记录表。 不要为唯一记录创建单独的表。 转换将唯一记录写入标准输出表。

标准输出表记录

Write_GoodMatches_Customer 目标表从标准输出组接收行。 该表接收唯一记录并接收重复的记录。 这些记录不需要手动查看。

下图显示了异常转换返回的标准输出记录：



Output

Name: exc_Consolidate.Good_Records

	Score	Row_Identifier	Cluster_ID	CUST_ID	COMPANY	CONTACT	TITLE	ADDR1	ADDR2	ADDR3
1	1	1 - 1001590	1	1001590	SHOP'N SAVE	MR DANIEL COWLEY	SR COMPUTER...	1755 WABASH...	SPRINGFIELD	<null>
2	1	1 - 1001599	2	1001599	SHOP'N SAVE	MS VERENE TEKAUTZ	NETWORK SYS...	45 GRAVOIS BL...	FENTON	MO
3	1	1 - 1001604	3	1001604	SHOP'N SAVE	MR REUBEN HENDRICKS	GENERAL MAN...	800 CARLYLE A...	BELEVILLE	IL
4	1	1 - 1001622	4	1001622	SHOPRITE	MS JACKY DAGENHART	INFO SYSTEMS...	60 BEAVER BR...	LINCOLN PARK	NJ
5	1	1 - 7121564	5	7121564	SPAR	MR BRADLY THOMAS	COMP SPEC	FORE STR	ST DENNIS	CORNWALL
6	0.850000000000...	1 - 7121565	5	7121565	SPAR	MS PEARL GARRISON	SURVEYING TE...	FORE STR	ST AUSTELL	BUGLE, CORN...
7	0.840000000000...	1 - 7121567	5	7121567	SPAR	MR EARL WILLIS	TECHNICIAN (C...	FORE STREET,...	BUNGLE	CORNWALL
8	0.779999999999...	1 - 7121597	5	7121597	SPAR	MRS JERRY FIGURES	LEAD SYS ANAL...	6 FORE STR	BUDLEIGH SAL...	DEVON
9	0.77	1 - 7121590	5	7121590	SPAR	MRS MINNA HASE	SR RESEARCH...	42 FORE STR	NEWTON ABBOT	BOVEY TRACEY
10	1	1 - 7121570	6	7121570	SPAR	MR GERMAYNE HANKS	SENIOR PROJE...	1 2 CHRISTINA...	TOTNESS	DEVON
11	1	1 - 7121571	7	7121571	SPAR	MRS CHERIE ALBERSON	CONTROLLER	1 N STR	ASHBURTON	DEVON
12	0.96	1 - 7121608	7	7121608	SPAR	MRS CHERIE ALBERSON	CONTROLLER	N STR	ASHBURTON	DEVON
13	1	1 - 1001658	9	1001658	SPARTAN STOR...	MS MARTHA CHASSE	PROJECT MANA...	5161 W MAIN S...	KALAMAZOO	MI
14	1	1 - 1001660	10	1001660	SPARTAN STOR...	MR DESMOND WINTERS	OWNER / CON...	1824 PORTAGE...	KALAMAZOO	MI
15	1	1 - 1001659	10	1001659	SPARTAN STOR...	MR DESMOND GRINDLEY	SYSTEMS ANAL...	1824 PORTAGE...	KALAMAZOO	MI
16	1	1 - 1001691	11	1001691	SPARTAN STOR...	MS SUZY TRAPANI	MGR ENG. SAL...	2545 W CADILA...	FARWELL	MI
17	1	1 - 1001664	12	1001664	SPARTAN STOR...	MR HARDEN DALTON	OWNER / CON...	438 LINCOLN A...	IONIA	MI
18	1	1 - 1001694	13	1001694	SPARTAN STOR...	MR REAMONN PELZER	CHIEF DBA	24445 DRAKE RD	FARMINGTON...	MI
19	1	1 - 1001729	14	1001729	SUN MART	MS ZELDA DECHICK	MIS MANAGER	1100 13TH AVE E	WEST FARGO	ND
20	1	1 - 1001724	15	1001724	SUN MART	MS MINERVA BAKER	COMPUTER SPE...	1921 W A STR	NORTH PLATTE	NE
21	1	1 - 1001732	16	1001732	SUPER FRESH	MS MARGA JANOWSKI	PRESIDENT	2105 PHILADEL...	CLAYMONT	DE
22	1	1 - 1001736	17	1001736	SUPER FRESH	MS BETTY EASTIP	ASSISTANT	2465 S BROAD...	HAMILTON TO...	NJ
23	1	1 - 1001738	18	1001738	SUPER FRESH	MS MINICA HOCH	MGR COORD	4330 48TH STR...	WASHINGTON	DC
24	1	1 - 1001758	19	1001758	SUPER STOP &...	MR ANCILIN SAMSON	AUTOMATED S...	150 NEW PK AVE	HARTFORD	CT
25	1	1 - 1001767	20	1001767	SUPERPETZ	MR EZARRAS DELAHANTY	LEAD PROGRA...	250 SHOUP MIL...	DAYTON	OH
26	1	1 - 1001796	21	1001796	SUPERPETZ	MR RUSTICUS WHITACRE	DATABASE AD...	1275 W PATRIC...	FREDERICK	MD
27	1	1 - 1001798	22	1001798	SUPERPETZ	MS TIMMIE NICE	NETWORK SYS...	2420 EASTERN...	YORK	PA

记录包含以下字段：

得分

指示群集中一个记录与另一记录之间相似度的匹配得分。 匹配得分为 1 的记录是重复记录，不需要查看。 其中所有记录的匹配得分都低于阈值下界的群集不是重复群集。

Row_Identifier

唯一标识表中每行的行编号。 对于该示例，行标识符包含客户 ID。

群集 ID

群集的唯一标识符。 群集中每个记录接收相同的群集 ID。 示例输出数据中的前四个记录是唯一的。 每个记录具有唯一的群集 ID。 第 5 到 9 行属于群集 5。 由于地址字段中的相似性，该群集中每个记录都是重复记录。

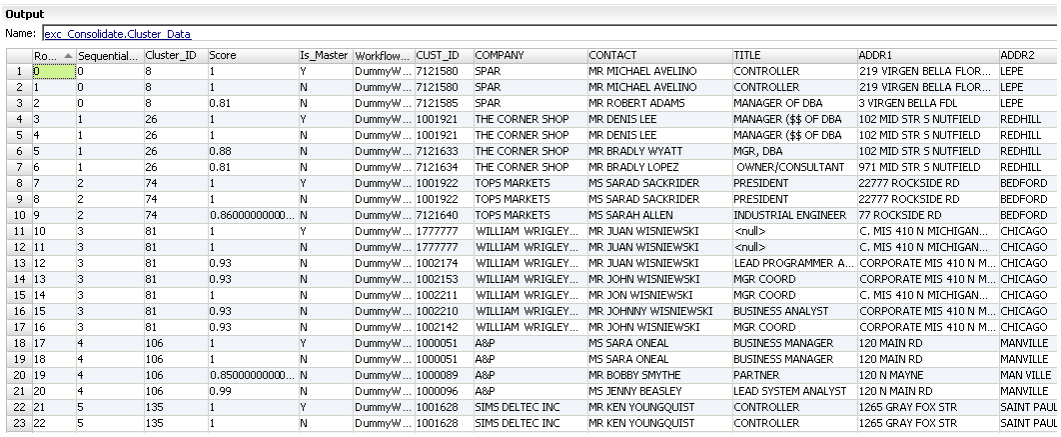
源数据字段

标准输出表组也接收所有源数据字段。

群集输出

Target_CustomerConsolidate 表从群集输出组接收记录。群集输出组返回可能是重复记录的记录。 Target_CustomerConsolidate 表中的记录需要在 Analyst 工具中手动查看。

以下图像显示了 Target_CustomerConsolidate 表中的一些记录和字段：



Output

Name: exc_Consolidate.Cluster_Data

	Ro...	Sequential...	Cluster_ID	Score	Is_Master	Workflow...	CUST_ID	COMPANY	CONTACT	TITLE	ADDR1	ADDR2
1	0	0	8	1	Y	DummyW...	7121580	SPAR	MR MICHAEL AVELINO	CONTROLLER	219 VIRGEN BELLA FLOR...	LEPE
2	1	0	8	1	N	DummyW...	7121580	SPAR	MR MICHAEL AVELINO	CONTROLLER	219 VIRGEN BELLA FLOR...	LEPE
3	2	0	8	0.81	N	DummyW...	7121585	SPAR	MR ROBERT ADAMS	MANAGER OF DBA	3 VIRGEN BELLA FDL	LEPE
4	3	1	26	1	Y	DummyW...	1001921	THE CORNER SHOP	MR DENIS LEE	MANAGER ($$ OF DBA	102 MID STR S NUTFIELD	REDHILL
5	4	1	26	1	N	DummyW...	1001921	THE CORNER SHOP	MR DENIS LEE	MANAGER ($$ OF DBA	102 MID STR S NUTFIELD	REDHILL
6	5	1	26	0.88	N	DummyW...	7121633	THE CORNER SHOP	MR BRADLY WYATT	MGR, DBA	102 MID STR S NUTFIELD	REDHILL
7	6	1	26	0.81	N	DummyW...	7121634	THE CORNER SHOP	MR BRADLY LOPEZ	OWNER/CONSULTANT	971 MID STR S NUTFIELD	REDHILL
8	7	2	74	1	Y	DummyW...	1001922	TOPS MARKETS	MS SARAD SACKRIDER	PRESIDENT	22777 ROCKSIDE RD	BEDFORD
9	8	2	74	1	N	DummyW...	1001922	TOPS MARKETS	MS SARAD SACKRIDER	PRESIDENT	22777 ROCKSIDE RD	BEDFORD
10	9	2	74	0.860000000000...	N	DummyW...	7121640	TOPS MARKETS	MS SARAH ALLEN	INDUSTRIAL ENGINEER	77 ROCKSIDE RD	BEDFORD
11	10	3	81	1	Y	DummyW...	1777777	WILLIAM WRIGLEY...	MR JUAN WISNIEWSKI	<null>	C. MIS 410 N MICHIGAN...	CHICAGO
12	11	3	81	1	N	DummyW...	1777777	WILLIAM WRIGLEY...	MR JUAN WISNIEWSKI	<null>	C. MIS 410 N MICHIGAN...	CHICAGO
13	12	3	81	0.93	N	DummyW...	1002174	WILLIAM WRIGLEY...	MR JUAN WISNIEWSKI	LEAD PROGRAMMER A...	CORPORATE MIS 410 N M...	CHICAGO
14	13	3	81	0.93	N	DummyW...	1002153	WILLIAM WRIGLEY...	MR JOHN WISNIEWSKI	MGR COORD	CORPORATE MIS 410 N M...	CHICAGO
15	14	3	81	1	N	DummyW...	1002211	WILLIAM WRIGLEY...	MR JON WISNIEWSKI	MGR COORD	C. MIS 410 N MICHIGAN...	CHICAGO
16	15	3	81	0.93	N	DummyW...	1002210	WILLIAM WRIGLEY...	MR JOHNNY WISNIEWSKI	BUSINESS ANALYST	CORPORATE MIS 410 N M...	CHICAGO
17	16	3	81	0.93	N	DummyW...	1002142	WILLIAM WRIGLEY...	MR JOHN WISNIEWSKI	MGR COORD	CORPORATE MIS 410 N M...	CHICAGO
18	17	4	106	1	Y	DummyW...	1000051	A&P	MS SARA ONEAL	BUSINESS MANAGER	120 MAIN RD	MANVILLE
19	18	4	106	1	N	DummyW...	1000051	A&P	MS SARA ONEAL	BUSINESS MANAGER	120 MAIN RD	MANVILLE
20	19	4	106	0.850000000000...	N	DummyW...	1000089	A&P	MR BOBBY SMYTHE	PARTNER	120 N MAYNE	MAN VILLE
21	20	4	106	0.99	N	DummyW...	1000096	A&P	MS JENNY BEASLEY	LEAD SYSTEM ANALYST	120 N MAIN RD	MANVILLE
22	21	5	135	1	Y	DummyW...	1001628	SIMS DELTEC INC	MR KEN YOUNGQUIST	CONTROLLER	1265 GRAY FOX STR	SAINT PAUL
23	22	5	135	1	N	DummyW...	1001628	SIMS DELTEC INC	MR KEN YOUNGQUIST	CONTROLLER	1265 GRAY FOX STR	SAINT PAUL

记录包含以下字段：

Row_Identifier

唯一标识表中每行的编号。

连续群集 ID

供每个群集在人工任务中查看的连续标识符。重复记录异常转换将连续群集 ID 添加至群集数据输出组中的记录。

群集 ID

群集的唯一标识符。匹配转换向所有输出记录分配群集 ID。重复记录和可能重复的记录共享一个群集 ID。唯一记录收到群集 ID，但是该记录不与任何其他记录共享 ID 号。

得分

指示群集中一个记录与另一记录之间相似度的匹配得分。需要手动查看的记录的匹配得分小于 0.95 且大于 0.80。

是主记录

指示记录是否是群集中的首选记录。

WorkflowID

因为转换不在工作流中，所以 WorkflowID 是 DummyWorkflowID。

记录字段

记录中的其他字段包含客户源数据。

创建重复记录异常转换

配置重复记录异常转换时，配置输入端口。定义阈值上界和下界用于确定匹配。配置写入重复记录和唯一记录的位置。

1. 创建可重用或不可重用重复记录异常转换。
   - 要创建可重用转换，请选择**文件** > **新建** > **转换**，然后选择重复记录异常转换。
   - 要创建不可重用转换，请打开映射，并将转换添加到映射画布。从向导中选择重复记录异常转换。
2. 单击**下一步**，或单击**完成**。

   如果单击**完成**，则可在创建转换之前更新默认阈值和数据路由选项。

3. 在“配置”视图中，配置上界和下界匹配得分阈值。
4. 在**数据路由选项**部分，配置标准输出和异常表属性，以设置转换写入每个记录类型的位置。

   或者，修改写入重复记录、要查看的重复记录以及唯一记录的位置。
5. 或者，生成唯一记录表。为新表输入数据库连接和表名称信息。生成唯一记录表时，转换在模型存储库中创建数据库对象。
6. 配置输入端口。添加输入端口时，Developer tool 向输出组中添加相同的端口名称。
   - 如果创建可重用转换，请选择**端口**选项卡，并为要连接到转换的数据添加端口。
   - 如果创建不可重用转换，请将其他对象添加到映射画布，并将输入端口拖动到转换中。
7. 将转换输出端口连接到一个或多个数据目标。将输出端口连接到与您在**配置**视图上设置的输出选项相对应的数据对象。
   - 如果要创建可重用转换，请将转换添加到映射并连接输出端口。
   - 如果创建不可重用转换，转换将端口连接至群集数据表。必须将输出端口连接至其他数据目标。

第 16 章

表达式转换

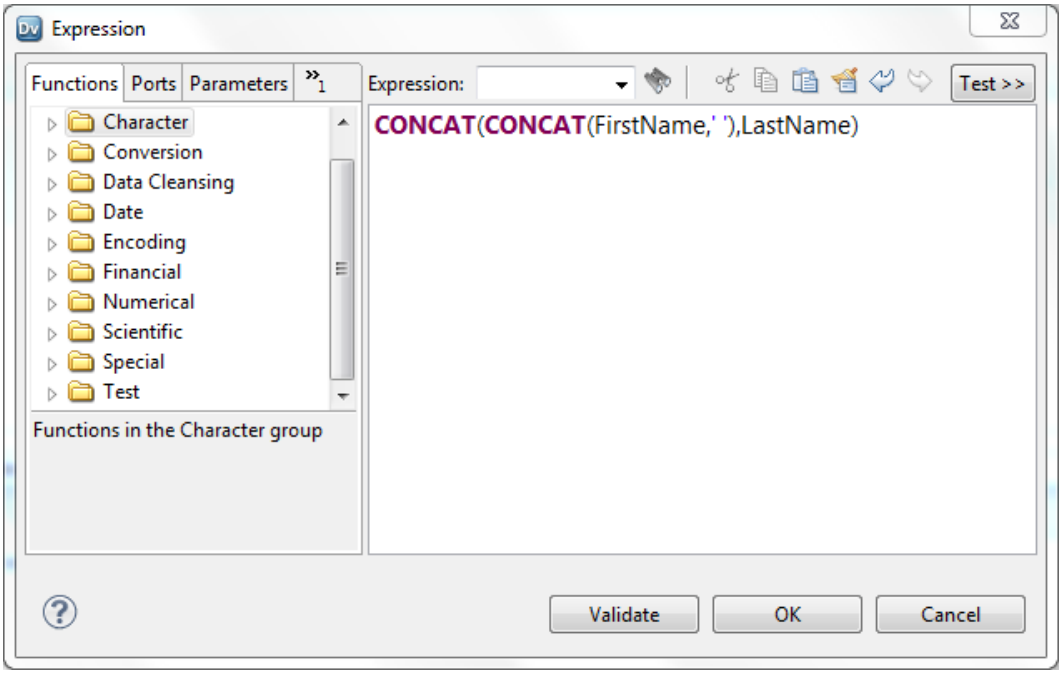
本章包括以下主题：

表达式转换概览

表达式转换是一种被动转换，用于在行中执行计算或测试条件语句。 在不可重用表达式转换中，可在定义映射输出时定义要汇总的映射输出表达式。

在单个行中，您可能需要创建一个表达式来调整员工薪酬、连接名字和姓氏，或将字符串转换为数字。

下图显示了一个表达式转换中用于连接名字、空格和姓氏的表达式：



通过为每个输出端口创建表达式，可以在表达式转换中输入多个表达式。 例如，您可能需要计算每个员工薪酬中不同类型的税款，例如，当地所得税和联邦所得税。 计算这两种税都需要员工薪酬和税率。 为每个计算定义单独的输出端口。 为每个输出端口定义不同的表达式。 可以为薪酬和税率定义传递端口，因为端口值不更改。

表达式转换端口

表达式转换具有多种在定义表达式时可以引用的端口类型。

表达式转换具有以下端口类型：

输入

从上游转换接收数据。如果表达式转换不更改端口值，则可以定义传递端口而非输入端口。

输出

包含表达式的返回值。您输入的表达式将成为输出端口的配置选项。还可以为每个端口配置默认值。

注意: 如果表达式导致数值错误，例如除以零或负数的平方根，则返回无穷值或 NaN 值。

传递

定义传递端口以将数据传递到该转换而不更改值。可以在计算中引用传递端口，但无法在传递端口中更改数据值。

变量

临时存储要在表达式中使用的数据。可以跨多行存储数据。可以定义一个表达式将值返回给变量端口。

动态端口

在动态映射中接收或返回端口。动态端口可以从上游转换接收一个或多个列并为每个列创建一个生成的端口。动态输出端口可以返回一个或多个生成的端口。可以定义输入规则以确定动态端口接收的列。动态输出端口可以包含用于生成多个输出端口的表达式。

生成的端口

表示动态端口内单一列的端口。表达式转换中的生成的端口可能会根据表达式转换从上游转换接收的列而更改。

测试表达式

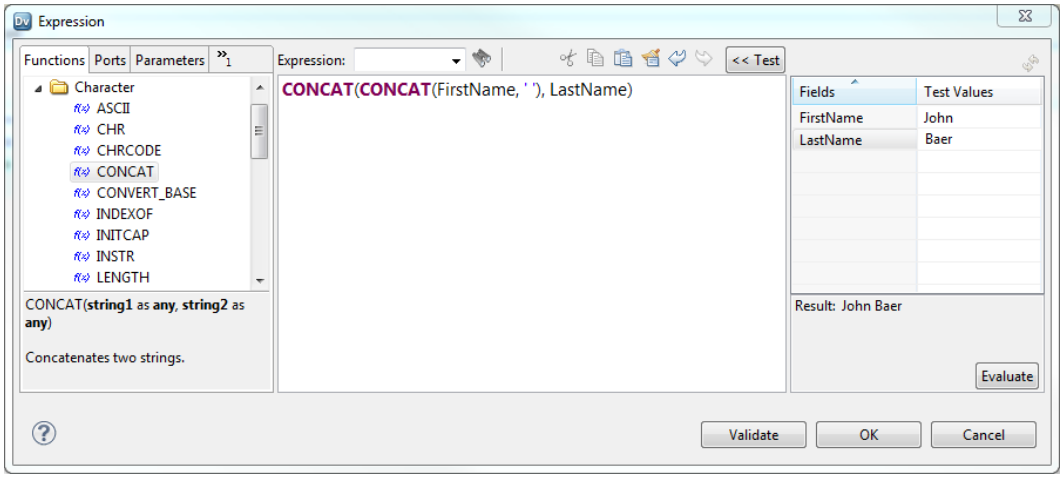
您可以测试在表达式编辑器中配置的表达式。 测试某个表达式时，需要输入示例数据，然后对该表达式求值。

在以下位置配置表达式时，可以测试表达式：

- 在表达式转换中的输出或变量端口中
- 将转换添加到映射后在表达式转换的映射输出视图中

例如，在配置连接名字、空格和姓氏的表达式之后，可以输入端口的示例数据，然后对该表达式求值，以便对结果进行验证。

下图显示了连接示例名字和姓氏的表达式的结果：



示例数据的数据格式字符串

测试一个使用数据类型为日期/时间或具有时区的时间戳的端口的表达式时，必须为使用所需日期格式字符串的端口输入示例数据。

要为数据类型为日期/时间的端口输入示例数据，请使用格式 MM/DD/YYYY HH24:MI:SS。 计算该表达式时，表达式编辑器将使用您在该表达式中指定的格式显示结果。 如果在表达式中省略了格式字符串，表达式编辑器将使用相同的格式 MM/DD/YYYY HH24:MI:SS 显示结果。

要为数据类型为具有时区的时间戳的端口输入示例数据，请使用格式 MM/DD/YYYY HH24:MI:SS TZR。 对该表达式进行求值时，表达式编辑器将使用格式 YYYY-MM-DD HH24:MI:SS.NS TZR 显示结果。

测试表达式

在表达式编辑器中测试表达式以评估表达式并验证结果。

1. 以下列方式之一打开表达式编辑器：
   - 在表达式转换中，单击**打开**按钮 ( )，该按钮位于输出端口或变量端口的**表达式**列中。

- 选择映射中所包括的表达式转换。 在**映射输出**视图中，单击**打开**按钮 ( )，该按钮位于输出的**表达式**列中。

2. 配置表达式。
3. 单击**测试 >>**，打开测试面板。
4. 在**测试值**列中输入每个字段的示例数据。

   可以为表达式中包括的每个端口或参数输入测试值。
5. 单击**计算**。

   表达式结果即会显示在测试面板底部。

端口选择器

转换具有生成的端口时，您需要对该转换进行配置，以便在生成的端口发生变化时转换成功运行。 您可以使用端口选择器来确定要在动态表达式、查找条件或联接器条件中使用的端口。

端口选择器是一个有序的端口列表，您可以在表达式中进行引用。生成的端口在动态映射中发生变化时，端口选择器可以包含不同端口。

例如，以下表达式在动态映射中引用生成的端口：

```
Salary * 12
```

您将映射配置为使用动态源，但各源文件中含薪资信息的列有不同的名称。列名称为 Salary、Monthly_Salary 或 Base_Salary。

您需要执行以下任务以适应不同的列名称：

1. 创建名为“Salary_PortSelector”的端口选择器。
2. 创建选择规则，以接受带“Salary”后缀的任何端口名称。
3. 更改表达式以包含端口选择器名称，而非“Salary”列名称。该表达式使用以下语法：

   ```
   Salary_PortSelector * 12
   ```

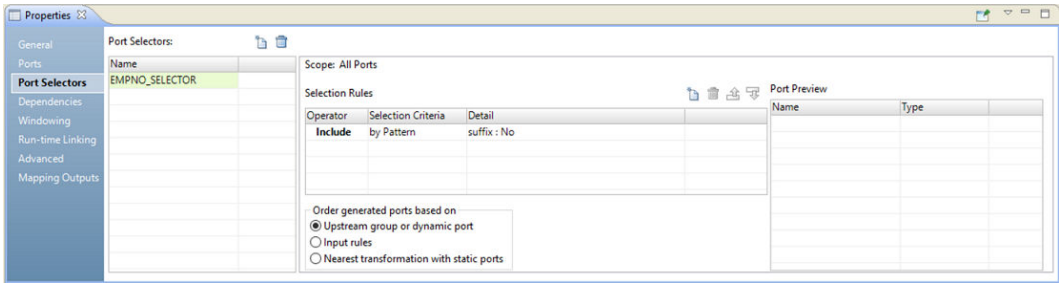
这样，无论使用哪个薪资端口名称，该表达式都可成功运行。

端口选择器配置

配置端口选择器时，可以定义选择规则来确定要包括哪些生成的端口。选择规则与您可以为动态端口配置的输入规则类似。

端口选择器可以包括静态端口或生成的端口。在**端口选择器**选项卡上配置端口选择器。

下图显示了**端口选择器**选项卡：



配置端口选择器的以下属性：

名称

标识端口选择器。可以在一个转换中创建多个端口选择器，然后在表达式中引用它们。

范围

标识要应用端口选择器的一组端口。为联接器或查找转换创建端口选择器时，必须选择范围。这些转换包含多个输入组。联接器转换具有“主”或“详细信息”范围。查找转换具有“导入”或“查找”范围。表达式转换包含一个输入组。范围始终是“所有端口”。

选择规则

确定要包括在端口选择器中的端口。创建选择规则时，**端口预览**面板显示当前输入端口中符合条件的端口。这些端口可能会发生变化。配置选择规则以接受来自不同源的端口。

选择规则

与端口选择器相关联的选择规则决定了端口选择器中要包括的端口。

创建选择规则时，**端口预览**面板显示当前输入端口中符合条件的端口。这些端口可能会发生变化。配置选择规则以接受来自不同源的端口。

请根据以下条件创建选择规则：

运算符

包括或排除选择规则返回的端口。默认为包括。必须先包括端口，然后才能排除端口。

选择条件

要创建的选择规则的类型。您可以根据列名称、端口类型、模式或复杂数据类型定义来创建规则。要根据列名称包括端口，请搜索特定名称或搜索名称中的字符模式。

详细信息

要应用到选择条件的值。如果选择条件是按列名称，则配置要搜索的字符串或名称。如果选择条件是按端口类型，则选择要包括的端口类型。

下表介绍了选择条件以及如何指定条件的详细信息：

选择条件	说明	详细信息
全部	包括所有端口。	无需详细信息。
名称	根据端口名称筛选端口。	从值列表中选择端口名称或使用一个类型为“端口”或“端口列表”的参数。
类型	根据每个端口的数据类型筛选端口。	从列表中选择数据类型。
模式	按照名称中的字符串或按照正则表达式来筛选端口。	选择前缀、后缀或正则表达式作为端口名称的模式类型。然后，输入模式值或使用类型为“字符串”的参数。
复杂数据类型定义	按照复杂数据类型定义来筛选端口。	选择前缀、后缀或正则表达式作为复杂数据类型定义的模式类型。然后，输入模式值或使用类型为“字符串”的参数。

创建端口选择器

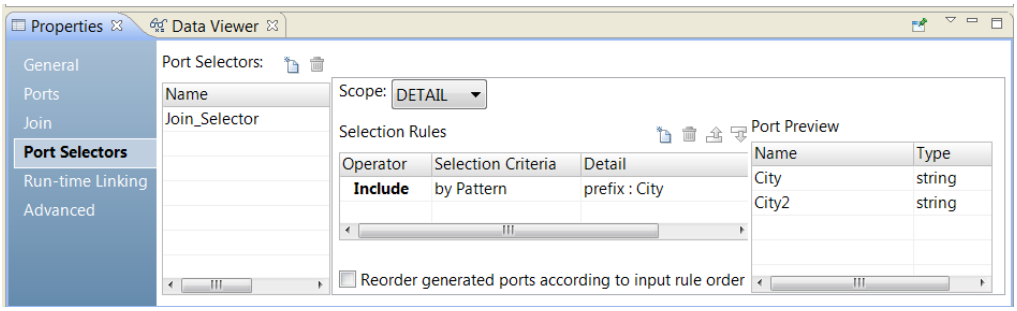
使用端口选择器可确定要在动态表达式、查找条件或联接器条件中使用的端口。

1. 单击**端口选择器**选项卡。
2. 在**端口选择器**区域中，单击**新建**。

   Developer tool 使用包含所有端口的默认选择规则来创建端口选择器。
3. 在**端口选择器**区域内，将端口选择器名称更改为唯一名称。
4. 如果您要使用联接器转换或查找转换，请选择范围。

   可用端口会根据所选的端口组发生相应变化。
5. 在**选择规则**区域内，选择**运算符**。
   - 包含。创建一个包含端口选择器端口的规则。必须先包括端口，然后才能排除端口。
   - 排除。创建一个在端口选择器中排除特定端口的规则。
6. 选择**选择条件**。
   - 按名称。按名称选择特定端口。您可以在范围内从端口列表中选择端口名称。
   - 按类型。按类型选择端口。可以选择一个或多个数据类型。
   - 按模式。按端口名称中的字符模式选择端口。可以使用特定字符进行搜索，也可以创建正则表达式。

   下图显示了“端口选择器”选项卡：



7. 单击**详细信息**列。

   输入规则详细信息对话框随即打开。
8. 选择作为端口筛选依据的值。
   - 按名称。选择按值或参数来创建端口列表。单击**选择**在列表中选择端口。
   - 按类型。从列表中选择一个或多个数据类型。**端口预览**区域显示所选类型的端口。
   - 按模式。选择在端口名称的前缀或后缀中搜索特定字符模式。或者，选择创建在搜索时使用的正则表达式。配置参数或者配置在搜索时使用的模式。

   配置规则时，**端口预览**区域会显示端口选择器中的端口。
9. 要重新排序端口选择器中的端口，请选择**根据输入规则顺序重新排序生成的端口**。

窗口化

当转换包含窗口函数时，您需要配置窗口化属性。窗口化仅可用于 Spark 引擎上的转换。

窗口函数对一组行执行运算，并为每个输入行计算返回值。

在表达式转换中定义窗口函数之前，您需要通过配置窗口化属性来描述该窗口。窗口化属性包括框架规范、分区键和排序键。框架规范指明哪些行将包括到当前行的总体计算中。分区键决定哪些行将位于同一个分区中。排序键决定如何对分区中的行进行排序。

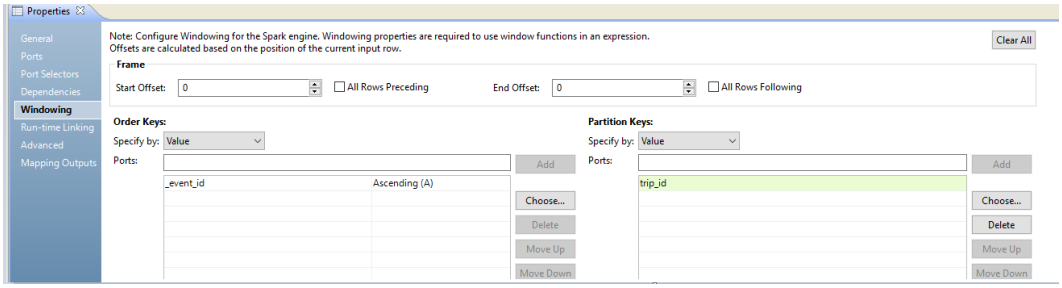
配置窗口化属性后，可以在表达式转换中定义窗口函数。Informatica 支持窗口函数 LEAD 和 LAG。还可以在表达式转换中使用汇总函数作为窗口函数。

窗口化配置

在表达式转换中包括窗口函数时，应配置与该函数关联的窗口化属性。窗口化属性定义与特定输入行关联的分区、排序和框架边界。

可以在“窗口化”选项卡上为转换配置窗口化。

下图显示了“窗口化”选项卡：



可以在“窗口化”选项卡上配置以下属性组：

框架

根据相对于当前输入行的位置的物理偏移来定义当前输入行的框架中包括的行。

如果使用汇总函数作为窗口函数，请配置框架。窗口函数 LEAD 和 LAG 引用单个行并忽略框架规范。

分区键

将输入行分隔到不同的分区。如果不定义分区键，则所有行属于单个分区。

排序键

定义分区中的行的排序方式。所选端口决定行在分区中的位置。排序键可以是升序或降序。如果不定义排序键，则行没有任何特定顺序。

框架

框架根据相对于当前行的位置来确定哪些行将包括在当前输入行的计算中。

如果使用汇总函数而非 LEAD 或 LAG，您必须指定窗口框架。LEAD 和 LAG 引用单个行并忽略框架规范。

起始偏移量和结束偏移量分别描述在当前输入行之前和之后出现的行数。偏移量“0”表示当前输入行。例如，起始偏移量 -3 和结束偏移量 0 描述包括当前输入行和当前行之前三行的框架。

下图显示了起始偏移量为 -1 且结束偏移量为 1 的框架：

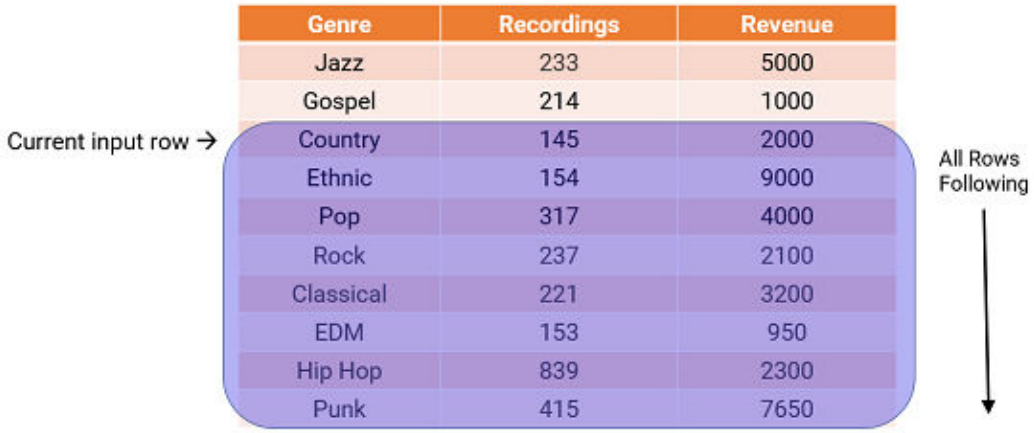
对于每个输入行，函数将对框架内的行执行汇总运算。如果使用前述框架配置诸如 SUM 等汇总表达式，该表达式将计算框架内值的总和并为输入行返回值 6000。

也可以指定不包括当前输入行的框架。例如，起始偏移量 10 和结束偏移量 15 描述总共包含六行（从当前行之后的第十行到第十五行）的框架。

注意: 起始偏移量必须小于或等于结束偏移量。

偏移量**所有在前的行**和**所有在后的行**分别表示分区的第一行和最后一行。例如，如果起始偏移量为“所有在前的行”且结束偏移量为 -1，则框架包括当前行之前一行以及该行之前的所有行。

下图显示了起始偏移量 0 且结束偏移量为“所有在后的行”的框架：



分区键和排序键

配置分区键和排序键，以构成行组并定义每个分区中的行的顺序或序列。

在窗口中使用以下键指定如何对行进行排序。

分区键

配置分区键可定义分区边界，而不对所有输入执行计算。窗口函数对与当前行位于同一个分区的行执行运算。

可以使用值或参数来指定分区键。选择**值**可使用端口名称。选择**参数**可使用排序键列表参数。排序键列表参数包含要用作排序依据的端口的列表。如果不指定分区键，所有数据将包含在同一个分区中。

排序键

使用排序键可确定如何对分区中的行进行排序。排序键定义了特定行在分区中的位置。

可以使用值或参数来指定排序键。选择**值**可使用端口名称。选择**参数**可使用排序键列表参数。排序键列表参数包含要用作排序依据的端口的列表。还必须选择要按升序还是降序来排列数据。如果不指定排序键，分区中的行不会按任何特定顺序进行排列。

示例

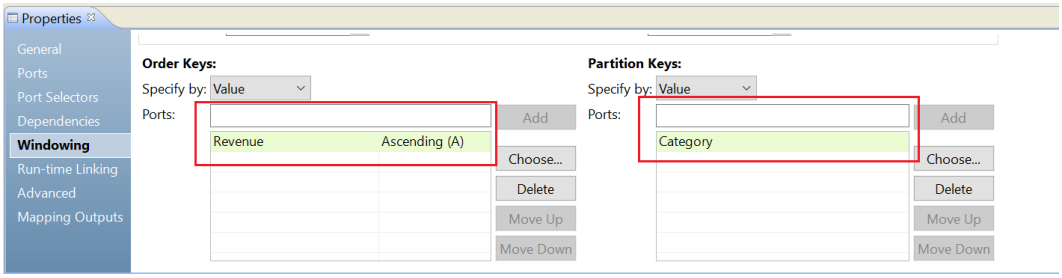
您经营一间咖啡店。您要计算最畅销以及第二畅销的咖啡和茶叶产品。

下表列出了产品、相应的产品类别和每种产品的收入：

Product	Category	Revenue
Espresso	Coffee	600
Black	Tea	550
Cappuccino	Coffee	500
Americano	Coffee	600
Oolong	Tea	250
Macchiato	Coffee	300
Green	Tea	450
White	Tea	650

可以按类别对数据进行分区，并按收入以降序对数据进行排序。

下图显示了可在“窗口化”选项卡上配置的属性：



下表显示了数据根据类别分组为两个分区。在每个分区中，收入按降序进行组织：

Product	Category	Revenue
Espresso	Coffee	600
Americano	Coffee	600
Cappuccino	Coffee	500
Macchiato	Coffee	300
White	Tea	650
Black	Tea	550

Product	Category	Revenue
Green	Tea	450
Oolong	Tea	250

根据分区规范和排序规范，确定了最畅销的两款咖啡是浓咖啡和美式咖啡，最畅销的两款茶叶是白茶和红茶。

窗口化配置的规则和准则

为转换配置窗口化时，应遵循某些准则。

为窗口函数定义窗口化属性时，请考虑以下规则和准则：

- 配置框架时，起始偏移量必须小于或等于结束偏移量。否则框架无效。
- 如果使用汇总函数作为窗口函数，请配置框架规范。LEAD 和 LAG 根据偏移量值执行运算并忽略框架规范。
- 不能将复杂端口用作分区键或排序键。
- 为分区键和排序键分配唯一的端口名称以避免运行时错误。
- 分区键和排序键不能同时使用动态端口以及从同一个动态端口生成的一个或多个端口。必须选择动态端口或生成的端口。

动态表达式

在动态输出端口中配置表达式时，该表达式将成为动态表达式。一个动态表达式可以生成多个输出端口。

可以在动态表达式中引用端口选择器或动态端口。端口选择器或动态端口包含多个端口时，动态表达式将针对每个端口运行。

配置动态表达式时，Developer tool 将不会验证生成的端口对于该表达式而言是否为有效类型。例如，如果在一个需要字符串类型的表达式中引用一个包含小数类型端口的端口选择器，该表达式在设计时将显示为有效。

示例

表达式转换具有以下生成的输入端口：

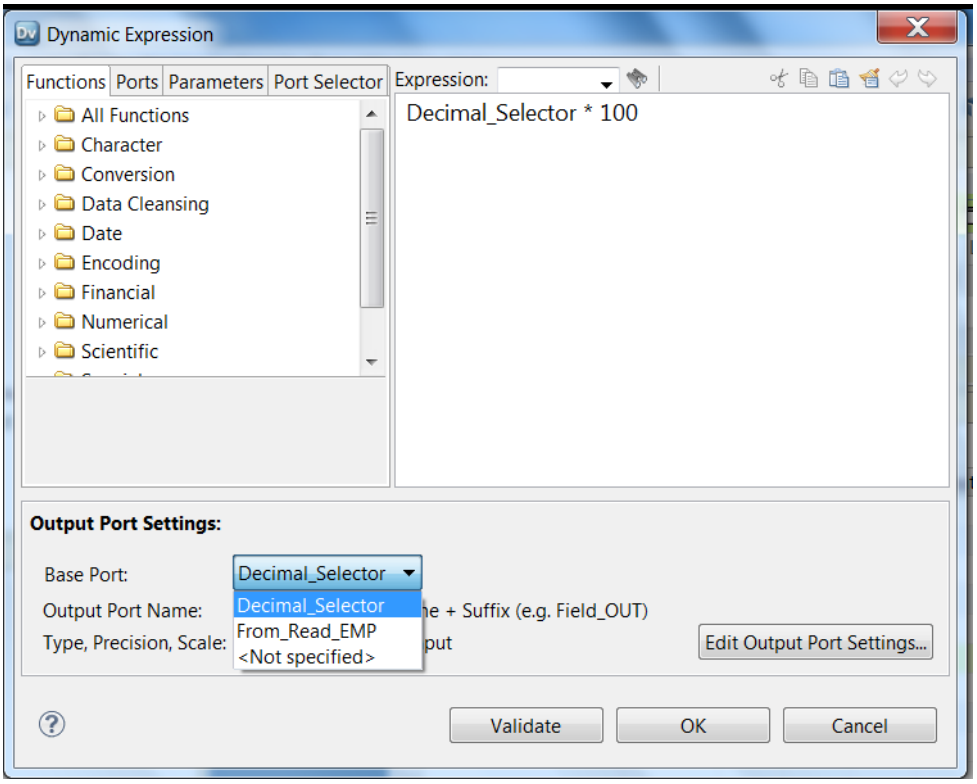
```
EMPNO   Decimal
NAME    String
SALARY  Decimal
DEPTNO  Decimal
```

该转换包含名为 MyDynamicPort 的动态输出端口。输出端口返回动态表达式的结果。动态表达式将端口选择器中每个端口的值乘以 100。该表达式为端口选择器中的每个端口分别运行一次。每个实例可以返回一个不同的结果。表达式转换为每个结果生成一个单独的输出端口。

Decimal_Selector 端口选择器具有一个包括小数数据类型端口的选择规则：

```
EMPNO   Decimal
SALARY  Decimal
DEPTNO  Decimal
```

下图显示了一个引用 Decimal_Selector 端口选择器的动态表达式：



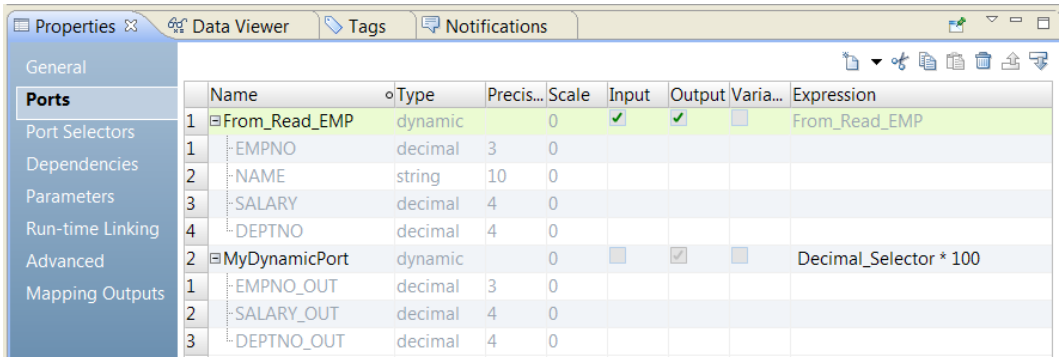
编辑输出端口设置以更改输出端口名称和输出端口属性。您还可以选择基本端口。

输出端口设置

可以指示要将哪些端口用作动态表达式的输入。 在**基本端口**区域中选择端口。

如果选择 Decimal_Selector 端口选择器作为基本端口，动态表达式将返回小数类型端口。 动态表达式不会为 NAME 端口生成端口，因为它是一个字符串。

下图显示了转换中的生成的端口：

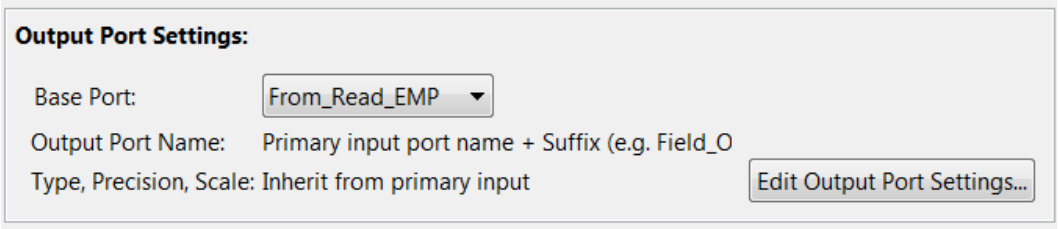


尽管 From_Read_Emp 动态端口是输入-输出端口，但转换只会返回 MyDynamicPort 动态输出端口中的端口。

可以配置如何对输出端口命名。 默认输出端口名称是输入端口名称和后缀 _OUT。

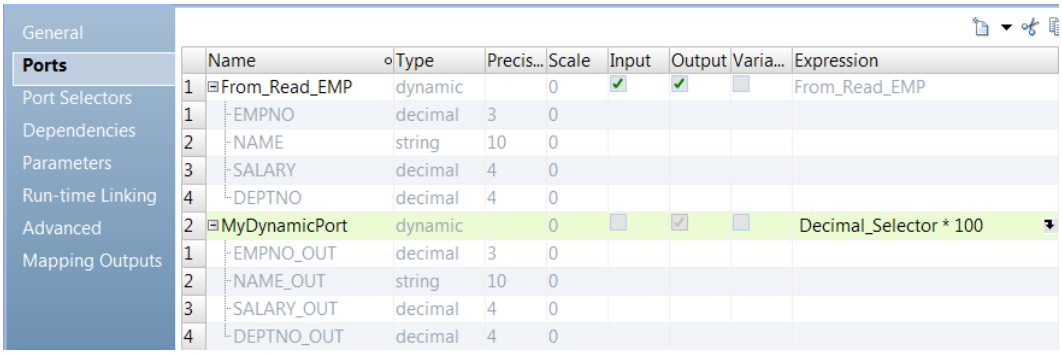
可以将基本端口更改为端口选择器。

下图显示了表达式编辑器中的输出端口设置：



如果将基本端口配置为 From_Read_EMP，您需要选择包含所有生成的输入端口的动态端口。 数据集成服务会针对 From_Read_EMP 中的所有端口运行动态表达式。

下图显示了基于 From_Read_Emp 输入的生成的输出端口：



生成的输出端口包括一个名为 NAME_OUT 的字符串类型的输出端口。

数据集成服务会为每个动态表达式生成输出端口。 如果创建一个生成 15 个端口的动态表达式并定义另一个生成 5 个端口的动态表达式，数据集成服务将生成 20 个输出端口。 每个动态输出端口生成一个不同的端口组。

创建动态表达式

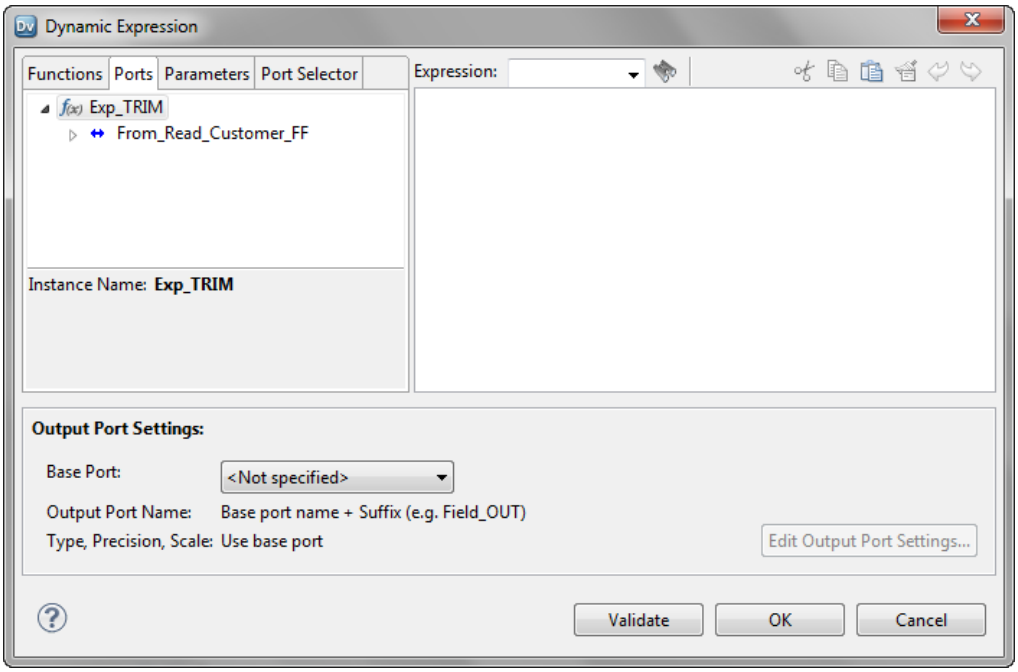
在表达式转换中创建动态表达式，以便针对动态端口或端口选择器中的每个端口运行一次该表达式。 动态表达式将结果返回到每个实例的单独生成的端口。

1. 在表达式转换中，转到**属性**视图，然后单击**端口**选项卡。
2. 单击**新建动态端口**。

   Developer tool 使用默认属性创建动态端口。
3. 重命名该动态端口并禁用输入选项。

   动态端口必须为输出端口。
4. 在动态输出端口的**表达式**列中，单击**打开**按钮 ( ).

此时将显示**动态表达式**对话框：

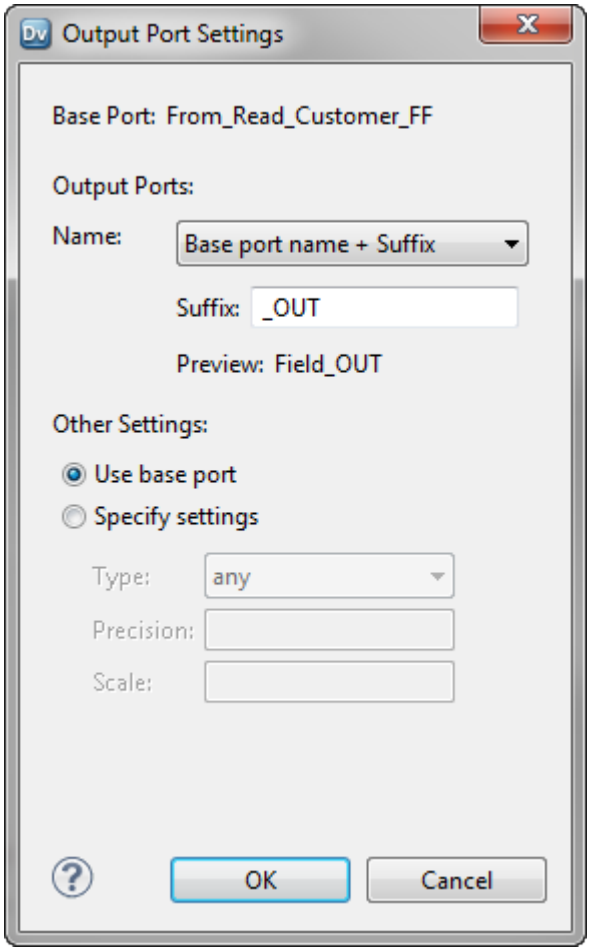


5. 在表达式编辑器中，输入表达式。 该表达式可以包含端口选择器或动态端口。
   例如 `LTRIM(RTRIM(Dynamic_Customer))`，其中 Dynamic_Customer 是动态端口。
6. 单击**验证**以验证表达式。
7. 单击**确定**退出**验证表达式**对话框。
8. 在**输出端口设置**区域中，从**基本端口**列表中选择动态输出端口，或者选择在表达式中引用的端口选择器。
   Developer tool 根据您的选择内容生成输出端口。

9. 按照以下步骤重命名输出端口：

   a. 单击**编辑输出端口设置**。

      此时将显示**输出端口设置**对话框。



   b. 在**名称**列表中，选择一个选项并输入前缀或后缀的值。 如果已选择**固定字符串 + 自动编号**，请输入输出端口名称的文本。 例如，如果您为输出端口名称输入 TRIM，输出端口名称将显示为 TRIM1、TRIM2、TRIM3。

   c. 或者，在**其他设置**区域中选择**指定设置**以更改输出端口的类型、精度和小数位数。 默认情况下，输出端口使用基本端口的设置。

   d. 单击**确定**。

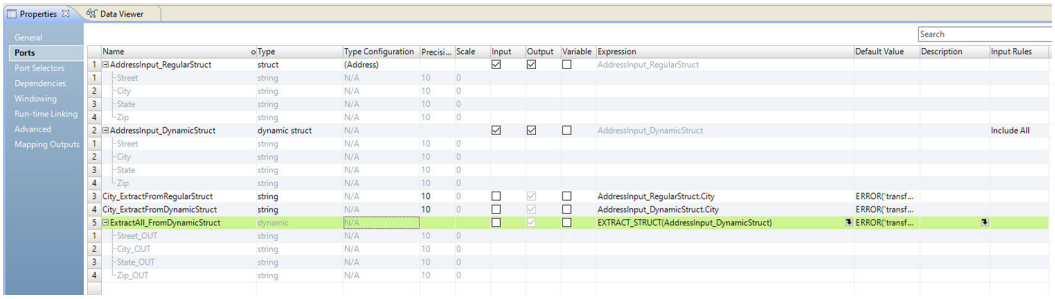
10. 单击**确定**退出**动态表达式**编辑器。

平展动态结构

在表达式转换中使用 EXTRACT_STRUCT 函数可以平展动态结构。

在表达式中使用点运算符可以提取结构的元素。

例如，您可以使用 EXTRACT_STRUCT 函数从一个地址中提取所有数据，而无需单独指定每个地址字段。



表达式转换高级属性

配置属性以帮助确定数据集成服务如何处理表达式转换的数据。

为表达式转换配置以下高级属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

维持行顺序

保持转换输入数据的行顺序。如果数据集成服务不应执行任何可能改变行顺序的优化，请选择此选项。

数据集成服务执行优化时，可能会失去之前在映射中建立的顺序。您可以在具有已排序的平面文件源、已排序的关系源或排序器转换的映射中设置顺序。当您将转换配置为维持行顺序时，数据集成服务将在执行映射优化时考虑此配置。如果数据集成服务可以保持顺序，将为转换执行优化。如果优化会改变行顺序，数据集成服务将不会为转换执行优化。

非本地环境中的表达式转换

非本地环境中的表达式转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的表达式转换

在下列情况下，映射验证会失败：

- 转换包含有状态变量端口。
- 转换在表达式中包含不受支持的函数。

如果表达式转换具有用户定义的函数，则对于函数中存在异常错误的行，转换将返回空值。

Spark 引擎上的表达式转换

在下列情况下，映射验证会失败：

- 转换包含有状态变量端口。
- 转换在表达式中包含不受支持的函数。

注意: 如果表达式导致数值错误，例如除以零或负数的平方根，它将返回一个空值，并且行不会出现在输出中。在本地环境中，表达式返回无穷值或 NaN 值。

流映射中的表达式转换

流映射具有不适用于批处理映射的其他处理规则。

您无法使用 EXTRACT_STRUCT 函数进行流映射。

Databricks Spark 引擎上的表达式转换

在下列情况下，映射验证会失败：

- 转换包含有状态变量端口。
- 转换在表达式中包含不受支持的函数。

注意: 如果表达式导致数值错误，例如除以零或负数的平方根，它将返回一个空值，并且行不会出现在输出中。在本地环境中，表达式返回无穷值或 NaN 值。

第 17 章

筛选器转换

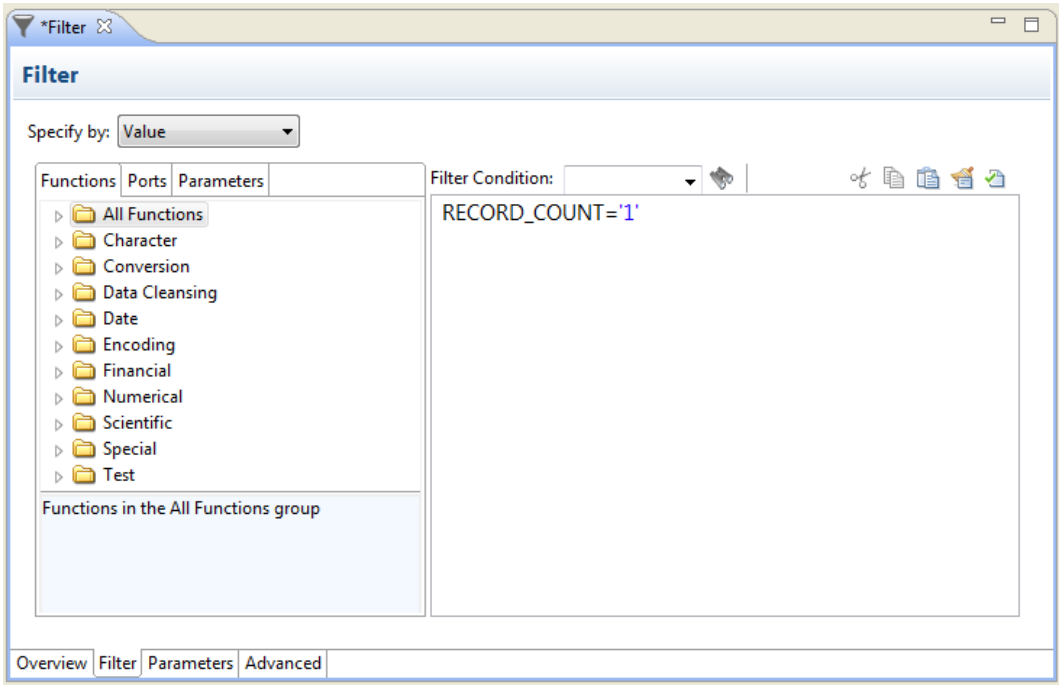
本章包括以下主题：

筛选器转换概览

使用筛选器转换筛选出映射中的行。 作为主动转换，筛选器转换可更改所传递的行数。

筛选器转换允许传递满足指定筛选条件的行。 该转换将删除不满足条件的行。 可以根据一个或多个条件筛选数据。

下图显示了筛选器转换中的筛选条件：



对于数据集成服务计算的每一行，根据该行是否满足指定条件，筛选条件将返回 TRUE 或 FALSE。 对于返回 TRUE 的每一行，数据集成服务将传递转换。 对于返回 FALSE 的每一行，数据集成服务将删除该行并在日志中写入一条消息。

无法将来自多个转换的端口连接到筛选器转换中。 筛选器的输入端口必须来自单个转换。

动态映射中的筛选器转换

可以在动态映射中使用筛选器转换。 可以在转换中配置动态端口，并在筛选条件中引用生成的端口。

您可以参数化整个筛选条件。 使用包含整个表达式的默认值配置表达式参数。 Developer tool 不验证采用参数默认值的筛选条件。

可以在筛选条件中引用动态端口。 动态端口可以包含多个生成的端口。 数据集成服务将筛选条件扩展为包含每个生成的端口。 每个生成的端口必须为有效类型，这样才能包括在表达式中。

可以在筛选条件中引用生成的端口。 但是，如果生成的端口在运行时不存在，则映射将会失败。

筛选条件

筛选条件是一个表达式，该表达式可返回 TRUE 或 FALSE。

在表达式编辑器中输入条件。筛选条件区分大小写。

可以使用返回一个值的任意表达式作为筛选器。例如，如果要筛选出代表薪资低于或等于 $30,000 的员工的行，请输入以下条件：

```
SALARY > 30000
```

可以使用 AND 和 OR 逻辑运算符指定条件的多个组成部分。如果要筛选出薪资低于 $30,000 和高于 $100,000 的员工，请输入以下条件：

```
SALARY > 30000 AND SALARY < 100000
```

可以在筛选条件中使用端口、参数、动态端口以及生成的端口。在表达式编辑器中选择端口和参数。

如果在筛选条件中使用动态端口，则筛选条件扩展为包括在动态端口中生成的所有端口。例如，动态端口 MyDynamicPort 包含三个小数端口：

```
Salary
Bonus
Stock
```

如果您配置以下筛选条件：

```
MyDynamicPort > 100
```

筛选条件会扩展为以下表达式：

```
Salary > 100 AND Bonus > 100 AND Stock > 100
```

可以为筛选条件输入一个常量。与 FALSE 等效的数值为零 (0)。任何非零值均等效于 TRUE。例如，转换包含一个名为 NUMBER_OF_UNITS 的端口，且该端口的数据类型为数值。可以配置筛选条件，使其在 NUMBER_OF_UNITS 的值等于零时返回 FALSE。否则，该条件将返回 TRUE。

注意: 您不能将单个端口选择器或动态端口用作布尔值。

无需将 TRUE 或 FALSE 指定为表达式中的值。TRUE 和 FALSE 是您设置的任何条件的隐式返回值。如果筛选条件的计算结果为 NULL，则该行为 FALSE。

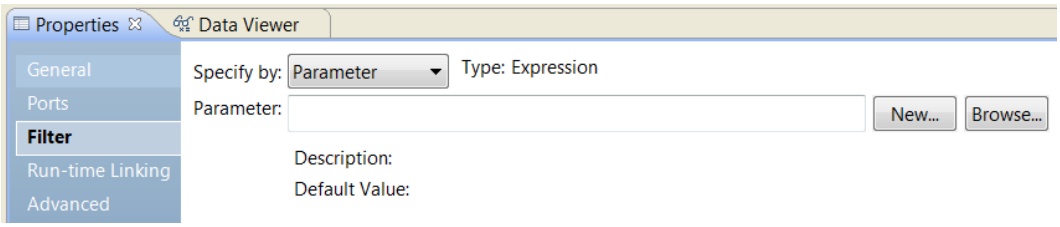
参数化筛选条件

可以配置一个表达式参数来定义筛选条件。 表达式参数包含整个表达式。

筛选器转换位于动态映射中时，您可能需要参数化筛选条件。 筛选条件可能根据运行时转换中的生成的端口而更改。

要将表达式参数用于筛选条件，请在筛选器转换属性的**筛选器**选项卡上选择**按参数指定**。

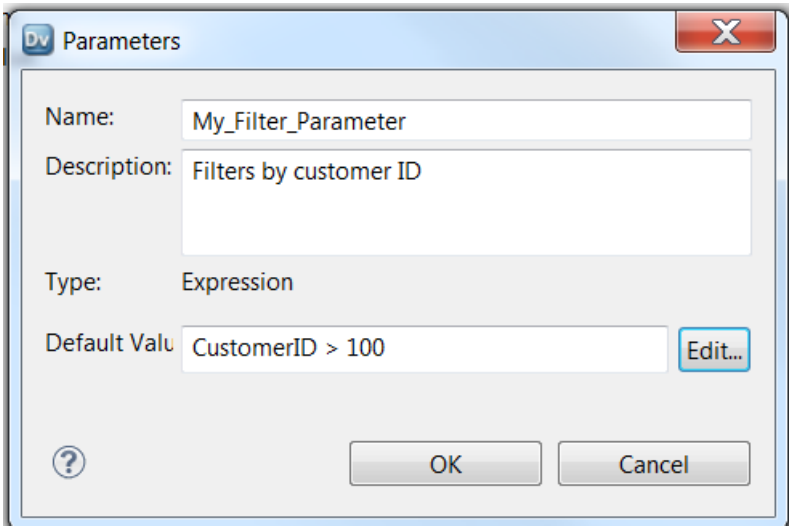
下图显示了使用参数指定筛选条件时的**筛选器**选项卡：



可以浏览并选择已经创建的表达式参数。 也可以创建表达式参数。

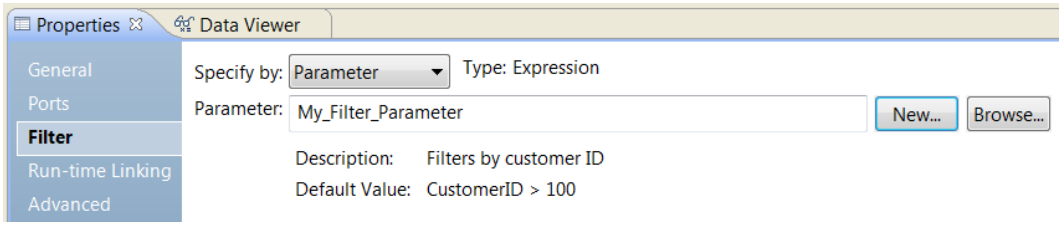
要创建表达式参数，请单击**新建**。 输入参数名称、说明和默认表达式值。

下图显示了参数输入位置：



可以在“参数”对话框上输入默认表达式。 如果希望使用表达式编辑器，请单击“编辑”。 使用表达式编辑器时，可以选择要在表达式中使用的函数和端口。 可以对表达式进行验证。

下图显示了“筛选器”选项卡和筛选条件参数：



表达式参数为映射参数。 可以在运行时替代参数集或参数文件中的参数。

筛选具有空值的行

要筛选包含空值或空格的行，请使用 ISNULL 和 IS_SPACES 函数测试端口的值。

例如，如果要在 FIRST_NAME 端口中筛选出包含 NULL 值的行，请使用以下条件：

```
IIF(ISNULL(FIRST_NAME),FALSE,TRUE)
```

此条件指出，如果 FIRST_NAME 端口为 NULL，则返回值为 FALSE，并且应放弃该行。 否则，该行将传递到下一个转换。

筛选器转换高级属性

配置有助于确定数据集成服务如何为筛选器转换处理数据的属性。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

筛选器转换性能提示

请参考提示来提高筛选器转换的性能。

在映射中尽早使用筛选器转换。

使筛选器转换尽可能接近映射中的源。 您可以在从源到目标的数据流中尽早筛选出不需要的数据，而不是将打算放弃的行传递到映射中。

非本地环境中的筛选器转换

非本地环境中的筛选器转换处理取决于运行该转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中无限支持。
- Databricks Spark 引擎。无限支持。

Blaze 引擎上的筛选器转换

当映射在 Hive 源的已分区列上包含筛选器转换时，Blaze 引擎可以只读取包含满足筛选条件的数据的分区。要将筛选器推送到 Hive 源，请将筛选器转换配置为映射中源之后的下一个转换。

第 18 章

层次结构到关系转换

本章包括以下主题：

层次结构到关系转换概览

层次结构到关系转换会对 XML 或 JSON 层次结构输入进行处理，并将其转换为关系输出。层次结构到关系转换将从输入端口读取层次结构输入，并在转换输出端口将数据转换为关系输出。要将层次结构输入转换为关系输出，请使用一个架构文件来定义层次结构数据。

可以使用层次结构到关系转换向导来自动映射数据。可以在转换的**概览**视图上配置到关系输出端口的映射。

向导生成转换后，可以将关系输出端口中的数据传递给映射中的其他转换。

注意: 层次结构到关系转换最多可以在一个 .xsd 文件中处理 10,000 个架构元素。要处理超过 10,000 个元素，请将数据拆分为多个文件。

示例 - 层次结构到关系转换

Harrinder Shipping 公司的物流部门必须处理出货数据。他们需要将库存和客户数据从层次结构格式转换为关系数据，以便能够存储在数据库表中。

他们需要创建一个映射，将层次结构数据转换为关系数据。公司库存系统生成的出货库存数据采用的是层次结构格式。该映射需要使用层次结构到关系转换，该转换输入出货数据，并采用某种可用的关系格式输出详细信息。

“出货”输入采用层次结构格式。Shipment 元素包括有关每次出货的客户和库存数据的子元素：

```
 Shipments
Shipment
    Items
     Item_Name
```

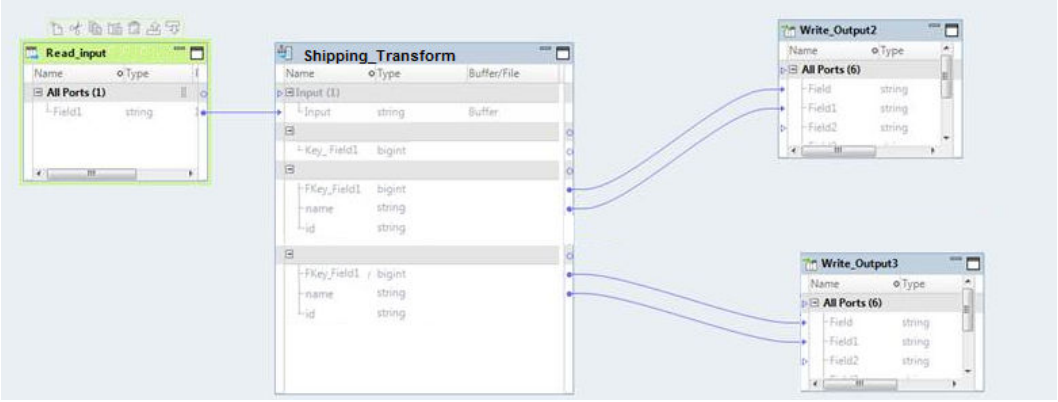
```
 Inventory_ID
Customer
 Customer_Name
 Customer_ID
 Customer_Address
```

在关系输出中，Customer_ID 元素是 Customer 表的主键，Shipment 表的外键。

Customer_ID	Customer_Name	Customer_Address
3543766	Tony Birch	6 Moby Drive
6342562	Sujita Man	22 Dan Street
6471862	Dwayne Horace	7 Jafendar Boulevard
7265204	Carmela Perez	23 Dan Street
4559672	DelilahSoraya	28 Jafendar Boulevard

Shipment_ID	Inventory_Item	Customer_ID
9173327437	908274	7265204
9174562342	553439	7265204
8484526471	546584	3543766
7023847265	908274	3543766
9174596725	553439	3543766

下图显示了本示例中的映射：



该映射包含以下对象：

Read_input

包括含有层次结构数据的文件路径的源。从 XML 文件中读取结算数据。

Shipping_Transform

层次结构到关系转换，该转换将 XML 输入转换为关系输出。

Write_Output2

以关系格式存储一部分转换后的数据（即 Customer 表）的目标。

Write_Output3

以关系格式存储另一部分转换后的数据（即 Shipment 表）的第二个目标。

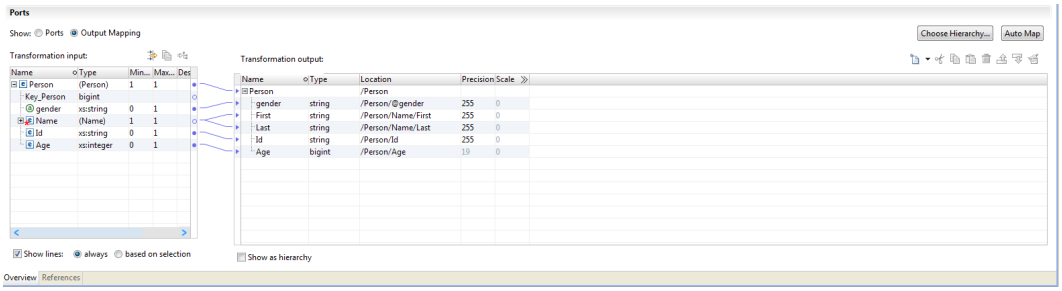
该映射使用 **Read_input** 平面文件来输入层次结构输入的目标路径。该映射通过 **Shipping_Transform** 转换来处理和转换数据，然后将输出存储在两个输出目标中。

输出关系端口与“概览”视图

在层次结构到关系转换中，为了将层次结构数据转换为关系输出，向导会在层次结构节点与关系端口之间生成链接。使用**概览**视图可查看关系端口与层次结构端口之间的链接。您也可以通过将层次结构输出中的节点链接至端口组来创建输出端口组。

要查看关系组的映射，请使用**概览**视图。选择**输出映射**。**端口**面板将显示在**概览**视图中。

下图显示了**端口**面板：



左侧是**转换输入**区域，显示了层次结构架构。右侧是**转换输出**区域，显示了关系输出端口。

您可以在**转换输出**区域中定义关系输出端口，并将架构中的节点链接至这些端口。还可以将指针从架构中的节点拖至**转换输出**区域中的空字段以创建端口。将输出架构中的节点拖至端口时，Developer tool 将显示两者间的链接。

层次结构到关系转换端口

层次结构到关系转换端口在转换的**概览**视图上定义。

层次结构到关系转换可以从文件或缓冲区读取输入。 输出端口会返回来自转换的关系数据。

创建层次结构到关系转换时，Developer tool 将创建一个默认输入端口。 输入类型确定了数据集成服务传递给层次结构到关系转换的数据类型。 输入类型确定输入是数据还是源文件路径。

请配置以下输入类型之一：

缓冲区

层次结构到关系转换会在输入端口中接收到多行源数据。 在配置转换以便接收来自 Informatica 转换的数据时，请使用缓冲区输入类型。

文件

层次结构到关系转换会在输入端口中接收到源文件路径。 层次结构到关系转换将打开源文件。 对于可能需要大量系统内存来处理缓冲区输入端口的大文件，也可以使用文件输入类型。

使用“新建转换”向导创建转换时，可以定义一个示例输入文件。 示例输入文件是一个小的输入文件示例。 创建层次结构到关系转换时将引用示例输入文件。 在**数据查看器**视图中测试转换时也将使用示例输入文件。

转换中包含一组或多组返回关系数据的端口。

架构引用

层次结构到关系转换需要使用一个层次结构架构来定义转换中的输入层次结构。要在转换中使用架构，可以定义架构引用。

可以在转换的**引用**视图中定义转换架构引用。

层次结构到关系转换会引用模型存储库中的架构对象。在创建转换之前，存储库中可以存在架构对象。您还可以在转换的**引用**视图中导入架构。

一个架构可引用其他多个架构。**引用**视图显示了层次结构到关系转换所引用的每个架构的命名空间和前缀。引用命名空间为空的多个架构时，转换无效。

端口配置

在**端口**面板中，该转换显示了层次结构架构节点与关系端口之间的映射。该转换使用一个架构来定义层次结构输入。如果架构具有多个可作为根元素的元素，请选择一个节点作为根元素。

向导会在层次结构架构节点与关系端口之间生成链接。如果要更改已生成的链接，您可以使用**端口**面板来添加、删除或编辑链接。您可以将若干节点链接至若干端口，然后创建一个端口。

将多个节点链接至**转换输出**区域时，Developer tool 会将位置字段更新为这些节点在层次结构中的位置。如果手动创建端口，则必须将节点映射至端口。更新**位置**列，然后从列表中选择一个节点。

将某个多次出现的节点链接至一个包含父元素的组时，您可以配置要包含的子元素的出现次数。也可以将父组替换为转换输出中多次出现的子组。

要创建组，请将节点链接至**转换输出**区域中的空列。如果将某个多次出现的子节点链接至一个空的输入或输出列，则 Developer tool 会要求您将该组与其他输出组相关联。选择一个组后，Developer tool 将创建用于关联组的键。

在**转换输出**区域中配置关联的输出端口组。Developer tool 提示您关联输出组时，将向这些组添加键。还可以手动添加端口以表示键。

层次结构到关系转换开发

使用“新建转换”向导自动生成层次结构到关系转换。 选择一个架构或层级结构示例文件来定义输入层次结构。

1. 在 Developer tool 中创建转换。
2. 配置输入端口和映射。
3. 测试该转换。

创建层次结构到关系转换

1. 在 Developer tool 中，依次单击**文件** > **新建** > **转换**。
2. 选择层次结构到关系转换，然后单击**下一步**。
3. 输入转换的名称并浏览要放置该转换的模型存储库位置，然后单击**下一步**。

4. 要选择一个架构，请选用下列方法之一：
   - 要使用模型存储库中的某个架构来定义输入层次结构，请在**架构对象**字段的附近，浏览存储库以从中选择该架构文件。
   - 要导入架构文件，请单击**创建新的架构对象**。在**新建架构对象**窗口中，您可以浏览并选择一个架构文件，也可以选择通过某个示例层次结构文件创建一个架构。
5. 选择输出层次结构的根元素。在**层次结构根元素**对话框中，选择架构中作为输出层次结构文件的根元素的元素。为了帮助选择根对象，您可以添加一个示例层次结构文件。要添加一个示例文件，请在**示例文件**字段的附近，浏览文件系统并从中选择该文件。
6. 单击**完成**。

   向导将在存储库中创建该转换。

配置端口和映射

在**概览**视图中配置输入和输出端口。

1. 选择输入端口数据类型、端口类型、精度和小数位数。
2. 要查看映射，请在**概览**视图的**端口**区域中，选择**输出映射**。
3. 展开**端口**网格中的树。 在左侧，**转换输入**面板会显示所需层次结构输入；在右侧，**转换输出**面板会显示关系输出。
4. 要将节点定义为根元素，请单击**选择层次结构**。

   Developer tool 仅在**转换输入**区域中显示根级别中的节点以及根级别下的节点。
5. 要查看连接端口与层次结构节点的行，请单击**显示行**。选择查看所有连接行或仅查看选定端口的行。
6. 要将输入组或端口添加到**转换输出**区域，请使用以下方法之一：
   - 将**转换输入**区域中的简单或复杂元素拖至**转换输出**区域的空列中。 如果该节点是组节点，则 Developer tool 将添加关系组，而不添加端口。
   - 要添加关系组，请选择一行，右键单击并选择**新建** > **组**。
   - 要添加关系端口，请右键单击并选择**新建** > **字段**。
7. 要清除层次结构节点设置的端口位置，请使用以下方法之一：
   - 在**转换输入**区域中选择一个或多个节点，右键单击并选择**清除**。
   - 选择将关系端口连接到层次结构节点的一个或多个行，然后右键单击并选择**删除**。
8. 要显示层次结构中的输出端口，请单击**显示为层次结构**。 每个子组都显示在其父组下方。

测试转换

在**数据查看器**视图中测试层次结构到关系转换。

测试转换之前，请确认您已定义文件输入位置。可以在**概览**视图的**端口**面板的“输入位置”列中定义数据集成服务计算机上的输入位置。

1. 打开**数据查看器**视图。
2. 单击**运行**。

   Developer tool 将验证转换。如果没有错误，则 Developer tool 将在**输出**面板中显示层次结构文件内容。

第 19 章

Java 转换

本章包括以下主题：

Java 转换概览

使用 Java 转换扩展 Developer 工具的功能。

Java 转换提供了一个简单的本地编程接口，用于使用 Java 编程语言定义转换功能。使用 Java 转换，无需高深的 Java 编程语言知识，也无需具备外部 Java 开发环境，即可定义简单或相对复杂的转换功能。Java 转换是主动或被动转换。

Developer 工具可使用 Java 开发工具包 (JDK) 编译 Java 代码，并为转换生成字节代码。Developer 工具会将字节代码存储在模型存储库中。

数据集成服务使用 Java 运行时环境 (JRE) 在运行时运行生成的字节代码。当数据集成服务使用 Java 转换运行映射时，该数据集成服务会使用 JRE 运行字节代码、处理输入行并生成输出行。

编写 Java 代码段来定义转换逻辑，以创建 Java 转换。请根据以下事件定义 Java 转换的转换行为：

- 转换接收输入行。
- 转换已处理所有输入行。

在运行于 Spark 引擎上的映射中，可以在 Java 转换中使用复杂数据类型来处理层次结构数据。使用复杂数据类型时，Spark 引擎可以直接在 Avro、Parquet 和 JSON 复杂文件中读取、处理和写入层次结构数据。

可重用和不可重用 Java 转换

可以创建可重用或不可重用的 Java 转换。

可重用转换可存在于多个映射中。 不可重用转换存在于单个映射内。

在用于定义属性和创建 Java 代码的编辑器中，视图会根据要创建的是可重用 Java 转换还是不可重用 Java 转换而有所不同。

主动和被动 Java 转换

Java 转换根据转换是主动转换还是被动转换而以不同的方式生成输出行。

创建转换后，无法更改转换是主动转换还是被动转换。

主动 Java 转换

主动转换可更改它传递的行数。

要定义输出中的行数，请在代码中调用 generateRow() 方法来生成每个输出行。可以选择从单个输入行生成多个输出行，或者从多个输入行生成单个输出行。例如，如果转换包含两个输入端口，这两个端口分别代表开始日期和结束日期，则可以调用 generateRow() 方法为开始日期和结束日期之间的每个日期生成一个输出行。

被动 Java 转换

被动转换不能更改转换传递的行数。转换调用 generateRow() 方法，在处理每个输入行之后生成一个输出行。

数据类型转换

Java 转换会根据 Java 转换端口类型将 Developer tool 数据类型转换为 Java 数据类型。

当 Java 转换读取输入行时，它会将输入端口数据类型转换为 Java 数据类型。

当 Java 转换写入输出行时，它会将 Java 数据类型转换为输出端口数据类型。

例如，在 Java 转换中，会对 Integer 数据类型的输入端口进行以下处理：

1. Java 转换会将输入端口的 Integer 数据类型转换为 Java 基元 Int 数据类型。
2. 在转换过程中，转换会将输入端口的值视为 Java 基元 Int 数据类型。
3. 当转换生成输出行时，它会将 Java 基元 Int 数据类型转换为 Integer 数据类型。

下表显示了 Java 转换是如何将 Developer tool 数据类型映射到 Java 基元和复杂数据类型的：

Developer Tool 数据类型	Java 数据类型
array*	java.util.List
bigint	long
binary	byte[]
date/time	如果启用了纳秒处理，则为具有纳秒精度的 BigDecimal 如果禁用了纳秒处理，则为具有毫秒精度的 long（自格林尼治标准时间 1970 年 1 月 1 日 00:00:00.000 起的毫秒数）
decimal	如果禁用了高精度处理，则为精度为 15 的 double 如果启用了高精度处理，则为 BigDecimal

Developer Tool 数据类型	Java 数据类型
double	double
integer	int
map*	java.util.Map
string	String
struct*	自定义的 JavaBean 类，具有适用于 struct 字段元素的 getter 和 setter
text	String
* 仅在 Spark 引擎上受支持。	

在 Java 中，java.util.List、java.util.Map、String、byte[] 和 BigDecimal 数据类型为复杂数据类型。Double、int 和 long 数据类型为基元数据类型。

在 Developer tool 中，array、struct 和 map 数据类型为复杂数据类型。

Java 转换会将基元数据类型中的空值设置为零。您可以使用**输入**选项卡上的 isNull 和 setNull API 方法将输入端口中的空值设置为输出端口中的空值。有关示例，请参阅“setNull” 页面上 296。

注意: 启用高精度时，decimal 数据类型将映射到 BigDecimal。BigDecimal 不能与某些运算符结合使用，例如 + 运算符。如果 Java 代码包含的表达式使用 decimal 端口或具有 decimal 数据类型元素的复杂端口，并且该端口与这些运算符之一结合使用，Java 代码将无法编译。

Spark 引擎上的复杂数据类型转换

可以向在 Spark 引擎上运行的映射中的 Java 转换添加复杂端口。复杂端口是一种分配了复杂数据类型的端口。Java 转换会将复杂数据类型转换为可比较的 Java 复杂数据类型。还可以将复杂数据类型的元素转换为可比较的 Java 数据类型，此类型为多种原始数据类型的装箱版本。

Array 数据类型

当 Java 转换读取输入行时，它会将 Array 数据类型转换为 Java List 数据类型。此转换会将 Array 元素的数据类型转换为 Java 数据类型的装箱版本。

例如，此转换会将 Developer tool 复杂数据类型 `array<integer>` 转换为 Java 数据类型 `List<Integer>`。

当 Java 转换写入输出行时，它会将 Java List 数据类型转换为 Array 数据类型。此转换会将 List 元素的数据类型转换为 Developer tool 数据类型的未装箱版本。

Struct 数据类型

当 Java 转换读取 Struct 数据类型的输入行时，会生成一个等效的 Java bean 类。Java bean 类的名称与 Struct 数据类型的名称相同。此转换会将 Struct 元素的数据类型转换为 Java 数据类型的装箱版本。

如果 Struct 数据类型使用了特殊字符或 Java 保留关键字，如 final 或 private，则 Java 转换会在类名称中使用下划线 (_) 替换这些特殊字符。您可以在转换属性选项卡的 Java 视图中打开完整代码来查看生成的类。

Java 转换会生成带下划线 (_) 前缀的类成员字段名称。还会为成员字段生成 getter 和 setter。

生成的 Java bean 类作为外部类，具有类型定义库名称的命名空间。外部类的名称与类型定义库的名称相同。Struct 端口的 Java 数据类型名称为以下格式：

```
type_library_name.struct_type_name
```

例如，类型库名称为 m_Type_Definition_Library。Struct 端口的复杂数据类型定义为：

```
Customer {
  name string
  age  integer
 }
```

Java 转换生成 Java bean 类，为成员字段生成 getter 和 setter。下列代码段显示了外部类和内部类：

```
public static final class m_Type_Definition_Library {
 public static final class Customer implements Serializable {
  private String _name;
  private Integer _age;

 public Customer() {}
```

下列代码段显示了 string 类型字符串的 Struct 元素名称的 getter 和 setter：

```
public String get_name() {
 return _name;
    }

public void set_name(String _name) {
 this._name = _name;
 }
```

当 Java 转换写入 Struct 数据类型的输出行时，会将 Java bean 类转换为 Struct 数据类型。此转换将类的成员字段转换为 Struct 元素。成员字段的数据类型被转换为 Developer tool 数据类型的未装箱版本。

Map 数据类型

当 Java 转换读取输入行时，它会将 Map 数据类型转换为 Java Map 数据类型。此转换会将 Map 元素的数据类型转换为 Java 数据类型的装箱版本。

例如，Developer tool 复杂数据类型 map<string，bigint> 被转换为 Java 数据类型 Map<String，long>。

当 Java 转换写入输出行时，它会将 Java Map 数据类型转换为 map 数据类型。此转换会将 Map 元素的数据类型转换为 Developer tool 数据类型的未装箱版本。

设计 Java 转换

设计 Java 转换时，必须考虑多个因素，例如，要创建的转换类型等。

设计 Java 转换时，请考虑以下问题：

- 是否需要创建主动或被动 Java 转换？

  被动 Java 转换会为转换中的每个输入行生成一个输出行。

  主动 Java 转换会为转换中的每个输入行生成多个输出行。
- 是否需要在 Java 转换中定义任何函数？ 如果需要，您希望在每个函数中包含哪些表达式？

  例如，您可以定义一个函数，该函数可调用一个表达式来查找输入或输出端口的值，或查找 Java 转换变量的值。
- 希望创建可重用还是不可重用 Java 转换？

  可重用转换可存在于多个映射中。

  不可重用转换只能存在于单个映射内。

Java 转换端口

Java 转换可以具有输入和输出端口。

要为不可重用 Java 转换创建和编辑端口，请使用编辑器中的**端口**选项卡。 要为可重用 Java 转换创建和编辑端口，请使用编辑器中的**概览**视图。

可为端口指定默认值。 在向转换添加端口后，可在 Java 代码段中使用端口名作为变量。

创建端口

创建 Java 转换时，该转换包含一个输入组和一个输出组。

创建端口时，Developer 工具会将该端口添加到当前所选行或组的下方。

设置默认端口值

可以为 Java 转换中的端口定义默认值。

Java 转换会根据端口的数据类型将端口变量初始化为默认端口值。

输入和输出端口

Java 转换将初始化未连接的输入端口或输出端口的值，这些端口在 Java 代码段中未分配值。

Java 转换根据以下 Java 数据类型初始化端口：

基元数据类型。

如果为不为空的端口定义默认值，转换会将端口变量的值初始化为默认值。否则，会将端口变量的值初始化为 0。

复杂数据类型

如果为端口定义默认值，转换则会创建新的对象，并将该对象初始化为默认值。否则，转换会将端口变量初始化为空值。例如，如果为字符串端口定义默认值，转换则会创建新的 String 对象，并将该 String 对象初始化为默认值。

注意: 如果在 Java 代码中访问具有空值的输入端口变量，将出现 NullPointerException。

您可以启用输入端口作为分区键和排序键，您可以指定排序方向。数据集成服务对数据进行分区，并按排序键和排序方向在每个分区中排序数据。当转换范围设置为“所有输入”时，分区键和排序键有效。

使用下列属性对数据进行分区和排序：

分区键

确定要分组到同一分区中的数据行的输入端口。

您可以启用一个或多个输入行作为分区键。数据集成服务在代码运行之前使用分区键将数据重新分区。如果不选择输入行作为分区键，则使用其默认分区方案处理数据。

排序键

确定每个分区内的排序条件的输入端口。

方向

升序或降序。 默认设置是升序。

Java 转换高级属性

Java 转换包含适用于转换代码和转换的高级属性。

如果在映射中使用转换，则可以替代转换属性。

您可以在**高级**选项卡上定义 Java 转换的以下高级属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

可分区

可以使用多个线程处理转换。如果希望数据集成服务使用一个线程来处理转换，请清除此选项。 数据集成服务可以使用多个线程处理余下的映射管道阶段。

当 Java 代码要求使用一个线程处理转换时，为 Java 转换禁用分区。

启用高精度

将精度小于或等于 38 的 Decimal 数据类型端口处理为 Java BigDecimal 数据类型端口。

禁用高精度以将 Decimal 数据类型端口处理为 Java Double 数据类型端口。

下表显示了 Java 转换是如何根据高精度选项的启用或禁用情况处理 Decimal 数据类型输入端口中的值的：

示例	已启用高精度处理	已禁用高精度处理
小数类型输入端口接到值 40012030304957666903。	Java 转换保留该值不变。	Java 转换将该值转换为以下值： 4.00120303049577 x 10^19

如果 Java 转换包含 Decimal 端口或带 Decimal 数据类型元素的复杂端口，则转换必须使用与映射相同的精度模式。例如，如果在 Java 转换中启用高精度，则在映射中也必须启用高精度。

在日期/时间中使用纳秒

将日期/时间数据类型端口转换为具有纳秒精度的 Java BigDecimal 数据类型端口。

禁用纳秒处理，以使生成的 Java 代码将日期/时间数据类型端口转换为具有毫秒精度的 Java Long 数据类型端口。

类路径

为与在**导入**选项卡上导入的非标准 Java 包相关联的 jar 或类文件目录设置类路径。

此 jar 或类文件目录必须能够在 Developer tool 客户端计算机上进行访问，才能编译 Java 代码。

根据不同的操作系统，按如下方法分隔类路径条目：

- 在 UNIX 上，使用冒号来分隔类路径条目。
- 在 Windows 上，使用分号来分隔类路径条目。

例如，如果要在**导入**选项卡上导入 Java 转换器包，并且在 converter.jar 中定义该包，则必须将 converter.jar 文件的位置添加到类路径中，才能为 Java 转换编译 Java 代码。

注意: 不需要为内置 Java 包设置类路径。例如，由于 java.io 是一个内置的 Java 包，因此不需要为 java.io 设置类路径。

是否主动

指示转换是否为主动转换。主动转换可更改它传递的行数。

创建转换后，无法更改此属性。如果需要更改此属性，请创建一个新的转换。

转换范围

定义数据集成服务用于将转换逻辑应用至传入数据的方法。 可以选择以下值之一：

- 行。 一次将逻辑转换应用至一行数据。 当过程结果取决于单行数据时选择“行”。
- 事务。 将转换逻辑应用至事务中的所有行。 当过程结果取决于同一事务中的所有行（但不取决于其他事务中的行）时选择“事务”。 选择“事务”时，必须将所有输入组连接至同一事务控制点。
- 全部输入。 将转换逻辑应用至所有传入数据。 选择“全部输入”时，数据集成服务将删除事务边界。 当过程结果取决于源中的所有数据行时选择“全部输入”。

无状态

保持转换输入数据的行顺序。如果数据集成服务不应执行任何可能改变行顺序的优化，请选择此选项。

数据集成服务执行优化时，可能会失去之前在映射中建立的顺序。您可以在具有已排序的平面文件源、已排序的关系源或排序器转换的映射中设置顺序。当您将转换配置为维持行顺序时，数据集成服务将在执行映射优化时考虑此配置。如果数据集成服务可以保持顺序，将为转换执行优化。如果优化会改变行顺序，数据集成服务将不会为转换执行优化。

为 Developer 工具客户端配置类路径

可以将 jar 文件或类文件目录添加到 Developer 工具客户端的类路径中。

要为运行 Developer tool 客户端的计算机设置类路径，请完成以下任务之一：

- 配置 CLASSPATH 环境变量。在 Developer tool 客户端计算机上设置 CLASSPATH 环境变量。该变量适用于在该计算机上运行的所有 Java 进程。
- 对于不可重用的 Java 转换，请在 Java 转换高级属性中配置类路径。此操作适用于包含此 Java 转换的映射。Developer tool 客户端在编译 Java 代码时会包含此类路径中的文件。

要向 Java 转换中的类路径添加 jar 或类文件目录，请完成以下步骤：

1. 在**高级**选项卡上，单击**类路径**旁边**值**列中的向下箭头图标。

   此时将显示**编辑类路径**对话框。
2. 要添加类路径，请完成以下步骤：
   a. 单击**添加**。

      此时将显示**另存为**窗口。
   b. 在**另存为**窗口中，导航到 jar 文件所在的目录。
   c. 单击**确定**。

   此时将在**编辑类路径**对话框中显示类路径。
3. 要删除 jar 文件或类文件目录，请选择 jar 文件或类文件目录，然后单击**删除**。

   该目录将不再显示在目录列表中。

为数据集成服务配置类路径

可以将运行时所需的 jar 文件或类文件目录添加到数据集成服务节点上的类路径中。

将运行时所需的 jar 文件放置在数据集成服务节点上的以下目录中：

```
$INFA_HOME/services/shared/jars
```

系统将动态加载此位置上的 jar 文件。单个映射在运行时所需的所有类文件都能从此目录中找到，并会从此目录中加载。

注意: Java 转换会将此目录中的所有 jar 文件添加到映射级类路径中。

开发 Java 代码

使用 **Java** 视图中的代码输入选项卡编写并编译为特定转换事件定义转换行为的 Java 代码。

您可以按任何顺序在代码输入选项卡上开发代码段。 您可以在**完整代码**选项卡上查看完整的 Java 代码，但无法进行编辑。

开发完代码段后，可以编译这些代码段或完整的 Java 代码，并在 **Java** 视图的**编译**属性中的**结果**窗口中查看编译结果。

每个代码输入选项卡都包含若干部分，可用于编写、查看和编译 Java 代码：

代码属性

提供一些控件，用于查看和输入 Java 代码，包括 Java 转换 API 方法。 下表介绍了在**代码**属性中提供的控件：

控件	说明
导航器	显示输入端口、输出端口以及可调用的 Java 转换 API 方法。 在导航器中单击某一项可显示该项的说明。 双击某一项可将其添加到 Java 代码窗口。 或者，您也可以将某一项从导航器拖入 Java 代码窗口。 以下代码输入选项卡可提供导航器： - 帮助程序 - 输入 - 结尾
Java 代码窗口	可用于查看或输入转换的 Java 代码。 Java 代码窗口采用基本 Java 语法突出显示方法来显示 Java 代码。 **注意**: 在**完整代码**选项卡上，您可以查看 Java 转换的完整类代码，但无法进行编辑。 以下代码输入选项卡可提供 Java 代码窗口： - 导入 - 帮助程序 - 输入 - 结尾 - 函数 - 优化器接口 - 完整代码
新建函数命令	打开**定义函数**对话框，此对话框可用于定义调用 Java 表达式的函数。 **函数**命令在**函数**选项卡上可用。
编辑工具栏	可用于单击诸如剪切、复制和粘贴等工具图标以编辑 Java 代码。 以下代码输入选项卡可提供编辑工具栏： - 导入 - 帮助程序 - 输入 - 结尾 - 函数 - 优化器接口

“编译”属性

提供一些控件，用于编译和调试 Java 代码。 下表介绍了**编译**属性中的控件：

控件	说明
编译命令	编译转换的 Java 代码。
结果窗口	显示 Java 转换类的编译结果，并可用于查找代码中的错误来源。 要查找代码中的错误，请在**结果**窗口中右键单击某个错误消息，然后选择查看一段代码或完整代码中的错误。 您也可以在**结果**窗口中双击某个错误消息以查找其来源。

创建 Java 代码段

要创建用于定义转换行为的 Java 代码段，请使用代码输入选项卡上的 **Java 代码**窗口。

1. 单击相应的代码输入选项卡。

   下表介绍了可在 **Java** 视图的代码输入选项卡上完成的任务：

选项卡	说明
导入	为主动或被动 Java 转换导入第三方、内置和自定义 Java 包。 导入包后，可在其他代码输入选项卡上使用这些包。
帮助程序	在主动或被动 Java 转换中为 Java 转换类声明用户定义的变量和方法。 声明变量和方法后，可在除**导入**选项卡之外的任何其他代码输入选项卡中使用这些变量和方法。
输入	定义主动或被动 Java 转换在收到输入行时的行为。 在此选项卡上定义的 Java 代码会对每个输入行运行一次。 在此选项卡上，您还可以访问并使用输入和输出端口数据、变量和 Java 转换 API 方法。
结尾	定义主动或被动 Java 转换在处理完所有输入数据后的行为。 在此选项卡上，您还可以为主动转换设置输出数据，并调用 Java 转换 API 方法。
函数	使用 Java 编程语言定义在 Java 转换中调用表达式的函数。 例如，您可以定义一个函数，该函数可调用一个表达式来查找输入或输出端口的值，或查找 Java 转换变量的值。 在**函数**选项卡上，您可以手动定义函数，也可以单击**新建函数**调用**定义函数**对话框，此对话框可用于轻松定义函数。
优化器接口	定义早期选择或推入筛选器优化。 在导航器中选择优化方法。 更新代码段以启用优化。 定义输入端口以及关联的输出端口来推送筛选器逻辑。
完整代码	只读。 在此选项卡上，可以查看并编译 Java 转换的完整类代码。

2. 要访问代码段中的输入或输出列变量，请展开导航器中的**输入**或**输出**列表，然后双击端口名称。
3. 要调用代码段中的 Java 转换 API，请展开导航器中的**可调用 API** 列表，然后双击方法的名称。 如有需要，请为方法配置相应的输入值。
4. 根据代码输入选项卡类型编写相应的 Java 代码。

   在**完整代码**选项卡的 **Java 代码**窗口中查看 Java 转换的完整类代码。

导入 Java 包

在**导入**选项卡上，可以为主动或被动 Java 转换导入 Java 包。

可以导入第三方、内置或自定义 Java 包。 导入 Java 包后，可在其他代码输入选项卡上使用这些导入的包。

注意: 在**导入**选项卡上，不能声明或使用静态变量、实例变量或用户方法。

在 Developer 工具中，当导出或导入包含 Java 转换的元数据时，不会导出或导入包含此 Java 转换所需的第三方或自定义包的 jar 文件或类文件。

如果要导入包含 Java 转换的元数据，则必须将包含所需的第三方或自定义包的 jar 文件或类文件复制到 Developer 工具客户端和数据集成服务节点。

例如，要导入 Java I/O 包，请在**导入**选项卡上输入以下代码：

```
import java.io.*;
```

如果要导入非标准 Java 包，请将包或类添加到 Java 转换的类路径中。

定义帮助程序代码

在**帮助程序**选项卡上，可以为主动或被动 Java 转换中的 Java 转换类声明用户定义的变量和方法。

在**帮助程序**选项卡上声明变量和方法后，可在除**导入**选项卡之外的任何其他代码输入选项卡中使用这些变量和方法。

在**帮助程序**选项卡上，可以声明以下类型的代码、变量和方法：

- 静态代码和静态变量。

  在静态块中，可以声明静态变量和静态代码。 某个可重用 Java 转换在映射中的所有实例会共享静态代码和变量。 在 Java 转换中，静态代码会在任何其他代码之前运行。

  例如，以下代码用于声明一个静态变量来存储某个 Java 转换在映射中的所有实例的错误阈值：

  ```
  static int errorThreshold;
  ```

  使用此变量可存储该转换的错误阈值，并从该 Java 转换在映射中的所有实例访问该阈值。

  注意: 必须一个可重用 Java 转换中同步静态变量。

- 实例变量。

  某个可重用 Java 转换在映射中的多个实例不会共享实例变量。 请使用前缀来声明实例变量以防止发生冲突，并初始化非原始实例变量。

  例如，以下代码使用一个布尔变量来确定是否生成输出行：

  ```
  // boolean to decide whether to generate an output row
  // based on validity of input
  private boolean generateRow;
  ```

- 用户定义的静态方法或实例方法。

  扩展 Java 转换的功能。 在**帮助程序**选项卡上声明的 Java 方法可使用或修改输出变量或本地声明的实例变量。 不能在**帮助程序**选项卡上从 Java 方法中访问输入变量。

  例如，在**帮助程序**选项卡上使用以下代码可声明一个函数以添加两个整数：

  ```
  private int myTXAdd (int num1,int num2)
  {
       return num1+num2;
  }
  ```

Java 转换 Java 属性

使用 **Java** 视图中的代码输入选项卡编写并编译为特定转换事件定义转换行为的 Java 代码。

以下选项卡为代码输入选项卡：

- **导入**
- **帮助程序**
- **输入**
- **结尾**
- **函数**
- **优化器接口**

在**完整代码**选项卡上可查看 Java 转换的完整类代码。

“导入”选项卡

在**导入**选项卡上，可以为主动或被动 Java 转换导入第三方、内置或自定义 Java 包。

要导入 Java 包，请在**导入**选项卡上**代码**属性的 **Java 代码**窗口中输入代码以导入包。

例如，可以输入以下代码来导入 java.io 包：

```
import java.io.*;
```

要编译用于导入 Java 包的代码，请在**导入**选项卡的**编译**属性中单击**编译**。 编译结果将显示在**导入**选项卡的**结果**窗口中。

导入 Java 包后，可在其他代码输入选项卡上使用这些包。

“帮助程序”选项卡

在**帮助程序**选项卡上，可以为主动或被动 Java 转换中的 Java 转换类声明用户定义的变量和方法。

要声明用户定义的变量和方法，请在**帮助程序**选项卡上**代码**属性的 **Java 代码**窗口中输入代码。

要编译 Java 转换的帮助程序代码，请在**帮助程序**选项卡的**编译**属性中单击**编译**。 编译结果将显示在**帮助程序**选项卡的**结果**窗口中。

声明变量和方法后，可在除**导入**选项卡之外的任何其他代码输入选项卡中使用这些变量和方法。

“输入”选项卡

在**输入**选项卡上，可以定义主动或被动 Java 转换在收到输入行时的行为。 在此选项卡上，您还可以访问并使用输入和输出端口数据、变量和 Java 转换 API 方法。

在此选项卡上定义的 Java 代码会对每个输入行运行一次。

要定义 Java 转换在收到输入行时的行为，请在**输入**选项卡上**代码**属性的 **Java 代码**窗口中输入代码。

从**输入**选项卡的导航器中，可以访问和定义以下变量和 API 方法：

- 输入端口和输出端口变量。 使用端口名称作为变量名称，以变量的形式访问输入和输出端口数据。 例如，如果“in_int”是一个整数输入端口，则可以使用 Java 原始数据类型 Int 以变量的形式引用“in_int”来访问该端口的数据。 无需将输入和输出端口声明为变量。

  请不要为输入端口变量赋值。 如果在**输入**选项卡上为输入变量赋值，则无法在当前行上获取相应端口的输入数据。

- 实例变量和用户定义的方法。 使用在**帮助程序**选项卡上声明的任何实例或静态变量或者用户定义的方法。

  例如，一个主动 Java 转换具有两个 Integer 数据类型的输入端口：BASE_SALARY 和 BONUSES，以及一个 Integer 数据类型的输出端口：TOTAL_COMP。 可以在**帮助程序**选项卡上创建一个用户定义的方法 myTXAdd，以添加两个整数并返回结果。 在**输入**选项卡上使用以下 Java 代码可将输入端口的总值分配给输出端口并生成输出行：

  ```
  TOTAL_COMP = myTXAdd (BASE_SALARY,BONUSES);
  generateRow();
  ```

  当 Java 转换收到输入行时，它会添加 BASE_SALARY 和 BONUSES 输入端口的值，将该值分配给 TOTAL_COMP 输出端口，并生成输出行。

- Java 转换 API 方法。 可以调用 Java 转换提供的 API 方法。

要编译 Java 转换的代码，请在**输入**选项卡的**编译**属性中单击**编译**。 编译结果将显示在**输入**选项卡的**结果**窗口中。

“结尾”选项卡

在**结尾**选项卡上，可以定义主动或被动 Java 转换处理完所有输入数据后的行为。 在此选项卡上，您还可以为主动转换设置输出数据，并调用 Java 转换 API 方法。

要定义 Java 转换处理完所有输入数据后的行为，请在**结尾**选项卡上**代码**属性的 **Java 代码**窗口中输入代码。

可在**结尾**选项卡上访问和定义以下变量和 API 方法：

- 输出端口变量。 可以使用在**端口**选项卡上定义的任何输出端口的名称作为变量，或者为主动 Java 转换设置输出数据。
- 实例变量和用户定义的方法。 使用在**帮助程序**选项卡上声明的任何实例变量或用户定义的方法。
- Java 转换 API 方法。 调用 Java 转换提供的 API 方法。

  例如，使用以下 Java 代码可在到达数据结尾时将信息写入日志：

  ```
  logInfo("Number of null rows for partition is: " + partCountNullRows);
  ```

要编译 Java 转换的代码，请在**结尾**选项卡的**编译**属性中单击**编译**。 编译结果将显示在**结尾**选项卡的**结果**窗口中。

“函数”选项卡

在**函数**选项卡上，可以使用 Java 编程语言定义在 Java 转换中调用表达式的函数。

例如，您可以定义一个函数，该函数可调用一个表达式来查找输入或输出端口的值，或查找 Java 转换变量的值。

要定义函数，可以在**函数**选项卡上**代码**属性的 **Java 代码**窗口中手动定义函数，也可以单击**新建函数**调用**定义函数**对话框，此对话框可用于轻松定义函数。

要编译代码，请在**函数**选项卡的**编译**属性中单击**编译**。 编译结果将显示在**函数**选项卡的**结果**窗口中。

“完整代码”选项卡

在**完整代码**选项卡上，可以查看和编译 Java 转换的完整类代码，但无法编辑此代码。

可以在**代码**属性的 **Java 代码**窗口中查看完整类代码。

要编译 Java 转换的完整代码，请在**完整代码**选项卡的**编译**属性中单击**编译**。 编译结果将显示在**完整代码**选项卡的**结果**窗口中。

Java 转换的筛选器优化

数据集成服务可对主动 Java 转换应用筛选器优化。 定义 Java 转换时，可以在 Java 转换**优化器接口**选项卡上添加筛选器优化代码。

通过 Java 转换执行早期选择优化

如果 Java 转换没有副作用，您可以为早期选择优化启用主动或被动 Java 转换。 优化器通过 Java 转换传递筛选器逻辑，并根据需要修改筛选条件。

要查看早期选择优化的代码段，请在**优化器接口**选项卡的导航器中选择“PredicatePushOptimization”。

allowPredicatePush

布尔型。 启用早期选择。 请将此函数更改为返回 true 结果和消息，以启用早期选择。 默认值为 false，并且函数将返回一条消息，指出不支持优化。

```
public ResultAndMessage allowPredicatePush(boolean ignoreOrderOfOp) {
    // To Enable PredicatePushOptimization, this function should return true
    //return new ResultAndMessage(true, "");
    return new ResultAndMessage(false, "Predicate Push Optimization Is Not Supported");
}
```

canGenerateOutputFieldEvalError

布尔型。 指示 Java 转换是否能够返回输出字段错误，例如除数为零错误。 如果 Java 转换不生成输出字段错误，请将函数更改为返回 false。 Java 转换能够生成字段错误时，数据集成服务无法使用早期选择优化。

```
public boolean canGenerateOutputFieldEvalError() {
    // If this Java transformation can never generate an output field evaluation error,
    // return false.
    return true;
}
```

getInputExpr

返回一个 Informatica 表达式，用于描述输入字段中的哪些输入值组成输出字段。 优化器需要了解哪些输入字段组成输出字段，才能通过转换推送筛选器逻辑。

```
public InfaExpression getInputExpr(TransformationField field,
        TransformationDataInterface group) {
    // This should return an Informatica expression for output fields in terms of input fields
    // We will only push predicate that use fields for which input expressions are defined.
    // For example, if you have two input fields in0 and in1 and three output fields out0, out1, out2
    // out0 is the pass-through of in1, out2 is sum of in1 and in2, and out3 is unknown, the code should
be:
    //if (field.getName().equals("out0"))
    //  return new InfaExpression("in0", instance);
    //else if (field.getName().equals("out1"))
    //    return new InfaExpression("in0 + in1", instance);
    //else if (field.getName().equals("out2"))
    //    return null;
    return null;
}
```

例如，某个映射中包含筛选器表达式 "out0 > 8"。 Out0 是指 Java 转换中 out0 输出端口的值。 您可以将 out0 的值定义为 in0 输入端口 + 5 的值。 优化器能够通过早期选择优化将以下表达式 "(in0 + 5) > 8" 推送到 Java 转换之外。 如果输出字段中不包含输入字段表达式，则可以返回空值。 优化器不会将筛选器表达式推送到不包含输入表达式的输出字段之外。
您可能会将以下代码包括在内：

```
if (field.getName().equals("out0"))
     return new InfaExpression("in0 + 5", instance);
else if (field.getName().equals("out2"))
     return null;
```

inputGroupsPushPredicateTo

返回能够接收筛选器逻辑的组列表。 Java 转换包含一个输入组。 请不要为 Java 转换修改此函数。

```
public List<TransformationDataInterface> inputGroupsPushPredicateTo(
        List<TransformationField> fields) {
    // This functions returns a list of input data interfaces to push predicates to.
    // Since JavaTx only has one input data interface, you should not have to modify this function
    AbstractTransformation tx = instance.getTransformation();
    List<DataInterface> dis = tx.getDataInterfaces();
    List<TransformationDataInterface> inputDIs = new ArrayList<TransformationDataInterface>();
    for (DataInterface di : dis){
        TransformationDataInterface tdi = (TransformationDataInterface) di;
        if (tdi.isInput())
            inputDIs.add(tdi);
    }
    if(inputDIs.size() == 1)
        return inputDIs;
    else
        return null;
}
```

通过 Java 转换执行推入优化

如果推入优化的主动 Java 转换没有副作用，并且优化不会影响映射结果，您可以启用此转换。

为 Java 转换配置推入优化时，请为 Java 转换定义存储此转换接收来自优化器的筛选条件的方式。 添加用于检查筛选条件的代码。 如果 Java 转换能够吸收筛选器逻辑，则会将 true 条件传递回优化器。 优化器将从优化的映射中删除筛选器转换。

配置 Java 转换时，您需要编写用于在优化期间将筛选条件存储为转换元数据的代码。 还需要编写用于在运行时检索筛选条件以及根据筛选器逻辑删除行的代码。

定义 Java 转换时，请在 Java 转换**优化器接口**选项卡上添加推入优化的代码。 要访问推入优化的代码段，请在转换**优化器接口**选项卡的导航器中选择“FilterPushdownOptimization”。

Developer 工具将显示代码段，以启用推入优化以及接收来自优化器的筛选条件。 请更新代码段，以启用优化并将筛选器逻辑另存为转换元数据。

isFilterSupported

返回 true 表示启用推入优化。 返回 false 表示禁用推入优化。
请将此函数更改为返回 true 以启用推入优化。

```
public ResultAndMessage isFilterSupported() {
    // To enable filter push-into optimization this function should return true
    // return new ResultAndMessage(true, "");
    return new ResultAndMessage(false, "Filter push-into optimization is not supported");
}
```

pushFilter

接收来自优化器的筛选条件。
添加用于检查筛选器以及确定筛选器逻辑是否能够在转换中使用的代码。 如果转换能够吸收筛选器，请使用以下方法将筛选条件存储为转换元数据：

```
storeMetadata(字符串键, 字符串数据)
```

字符串键是指元数据的标识符。 您可以将任何字符串定义为字符串键。 字符串数据是指您要存储的数据，以便确定运行时要删除的行。 例如，字符串数据可以是 Java 转换接收的来自优化器的筛选条件。

```
public ResultAndMessage pushFilter(InfaExpression condition) {

    // Add code to absorb the filter
    // If filter is successfully absorbed return new ResultAndMessage(true, ""); and the optimizer
    // will remove the filter from the mapping
```

```
        // If the filter is not absorbed, return new ResultAndMessage(false, msg);
        return new ResultAndMessage(false, "Filter push-into optimization is not supported");
    }
```

创建 Java 转换

在 Developer 工具中，可以创建可重用或不可重用 Java 转换。

创建可重用 Java 转换

可重用转换可存在于多个映射中。

在 Developer 工具中创建可重用 Java 转换。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择 Java 转换。
4. 单击**下一步**。
5. 输入转换的名称。
6. 要创建主动转换，请选择**创建为主动**选项。
7. 单击**完成**。

   此时，转换将显示在编辑器中。
8. 在**端口**视图中，单击**新建**按钮向转换添加端口。
9. 编辑端口以设置名称、数据类型和精度。

   在 Java 代码段中使用端口名称作为变量。
10. 在 **Java** 视图中，可以使用代码输入选项卡来编写和编译转换的 Java 代码。
11. 在 **Java** 视图中，可以使用**函数**选项卡来定义调用表达式的函数。
12. 在任何代码输入选项卡上，双击**编译**属性的**结果**窗口中的错误消息，找到并修复转换的 Java 代码中的编译错误。
13. 在**高级**视图中，编辑转换属性。

创建不可重用 Java 转换

不可重用转换存在于单个映射内。

在 Developer 工具中创建不可重用 Java 转换。

1. 在映射或 Mapplet 中，将“转换”选项板中的 Java 转换拖动到编辑器中。
2. 在**新建 Java 转换**对话框中，输入转换的名称。
3. 要创建主动转换，请选择**创建为主动**选项。
4. 单击**完成**。

   此时，转换将显示在编辑器中。
5. 在**常规**选项卡中，编辑转换名称和说明。

6. 在**端口**选项卡中，单击**新建**按钮将端口添加到转换。
7. 编辑端口以设置名称、数据类型和精度。

   在 Java 代码段中使用端口名称作为变量。
8. 在 **Java** 视图中，可以使用代码输入选项卡来编写和编译转换的 Java 代码。
9. 在 **Java** 视图中，可以使用**函数**选项卡来定义调用表达式的函数。
10. 在任何代码输入选项卡上，双击**编译**属性的**结果**窗口中的错误消息，找到并修复转换的 Java 代码中的编译错误。
11. 在**高级**视图中，编辑转换属性。

编译 Java 转换

Developer 工具可使用 Java 编译器编译 Java 代码，并为转换生成字节代码。

Java 编译器将编译 Java 代码，并在代码输入选项卡上**编译**属性的**结果**窗口中显示编译结果。 Java 编译器会与 Developer 工具一起安装在 java/bin 目录中。

要编译 Java 转换的完整代码，请在**完整代码**选项卡的**编译**属性中单击**编译**。

创建 Java 转换时，此转换将包含一个 Java 类，用于定义 Java 转换的基本功能。 该 Java 类的完整代码包含此转换的模板类代码以及在代码输入选项卡上定义的 Java 代码。

编译 Java 转换时，Developer 工具会将代码输入选项卡中的代码添加至此转换的模板类，以为此转换生成完整类代码。 然后，Developer 工具会调用 Java 编译器来编译完整类代码。 Java 编译器将编译此转换，并为此转换生成字节代码。

编译结果将显示在**结果**窗口中。 使用编译结果可识别并查找 Java 代码错误。

Java 转换故障排除

在任意代码输入选项卡的**编译**属性的**结果**窗口中，可以找到并修复 Java 代码错误。

由于在代码输入选项卡的代码中出现错误，或在 Java 转换类的完整代码中出现错误，均会在 Java 转换中发生错误。

要为 Java 转换排除故障，请完成以下高级别步骤：

1. 在 Java 代码段代码中或在转换的完整类代码中找到错误源。
2. 识别错误类型。 通过**结果**窗口中的编译结果以及错误的位置来识别错误类型。
3. 在代码输入选项卡上修复 Java 代码。
4. 重新编译转换。

查找编译错误来源

要查找编译错误来源，请使用**结果**窗口中代码输入选项卡或**完整代码**选项卡上**编译**属性中显示的编译结果。

在**结果**窗口中双击某条错误消息时，在 **Java 代码**窗口的代码输入选项卡或**完整代码**选项卡上会突出显示导致此错误的源代码。

您可以在**完整代码**选项卡上查找错误，但不能在**完整代码**选项卡上编辑 Java 代码。 要修复在**完整代码**选项卡上发现的错误，请在相应的代码输入选项卡上修改代码。 您可能需要使用**完整代码**选项卡查看由于向转换的完整类代码添加用户代码而导致的错误。

在代码输入选项卡或“完整代码”选项卡中查找错误

可以在代码输入选项卡或**完整代码**选项卡中查找编译错误。

1. 在**结果**窗口任何代码输入选项卡或**完整代码**选项卡**的编译**属性中，右键单击错误消息。
2. 单击**显示位置** > **代码段**或**显示位置** > **完整代码选项卡**。

   Developer 工具将在所选选项卡中突出显示错误来源。

   注意: 在**完整代码**选项卡中可以查看错误，但不能更正错误。 要更正错误，必须导航到相应的代码输入选项卡。

标识编译错误来源

编译错误可能会在用户代码中显示为错误的结果。

用户代码中的错误还可能会在该类的非用户代码中生成错误。 Java 转换的用户代码和非用户代码中均会产生编译错误。

用户代码错误

代码输入选项卡上的用户代码可能发生错误。 用户代码错误包含标准 Java 语法错误以及语言错误。

当 Developer 工具将用户代码从代码输入选项卡添加至完整类代码时，也可能发生用户代码错误。

例如，Java 转换有一个输入端口，名称为 int1，数据类型为 integer。 该类的完整代码使用以下代码声明输入端口变量：

```
int int1;
```

但是，如果在**输入**选项卡上使用了相同的变量名称，Java 编译器将发出一个错误，要求对变量进行重新声明。 要修复此错误，请在**输入**选项卡上重命名该变量。

非用户代码错误

代码输入选项卡上的用户代码可在非用户代码中导致错误。

例如，某个 Java 转换具有输入端口 int1 和输出端口 out1，数据类型为 Integer。 在**输入**代码输入选项卡上输入以下代码以计算输入端口 int1 的目标并将其指定给输出端口 out1：

```
int interest;
interest = CallInterest(int1); // calculate interest
out1 = int1 + interest;
}
```

编译转换时，Developer 工具会将**输入**代码输入选项卡上的代码添加到此转换的完整类代码中。 Java 编译器编译 Java 代码时，不匹配的大括号会导致完整类代码中的某个方法过早结束，从而导致 Java 编译器发出错误。

转换为结构数据示例

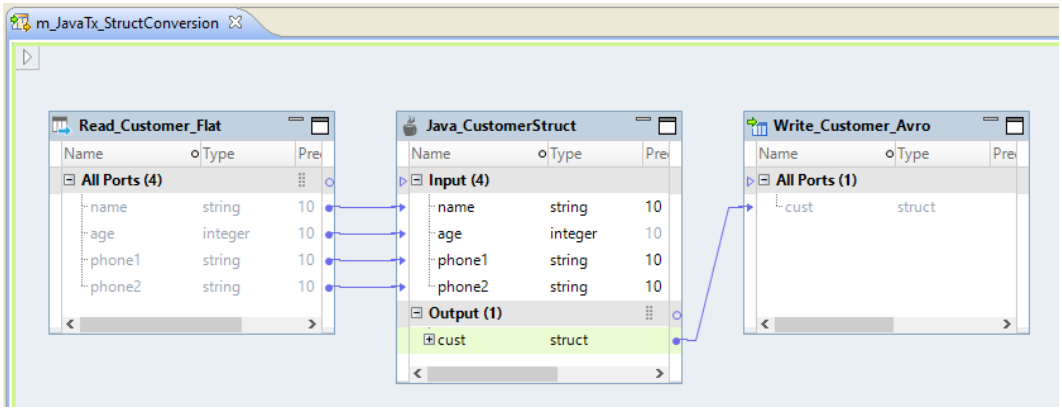
您的组织需要将平面文件中的大量客户数据转换为结构数据，并将其写入 Avro 文件。输入文件包含诸如姓名、年龄和电话号码等客户详细信息。如果输入文件中的客户姓名为空，则表示您不想将客户详细信息添加到输出文件中。

可以使用 Java 转换开发一个映射，以定义转换功能。在 Hadoop 环境下，在 Spark 引擎上运行映射，以转换数据并将结构数据写入 Avro 文件。

创建映射并配置以下转换：

- 读取转换，从平面文件源读取客户信息
- Java 转换，作为活动转换，将平面数据转换为结构数据并删除不一致的数据
- 写入转换，将结构数据写入 Avro 文件

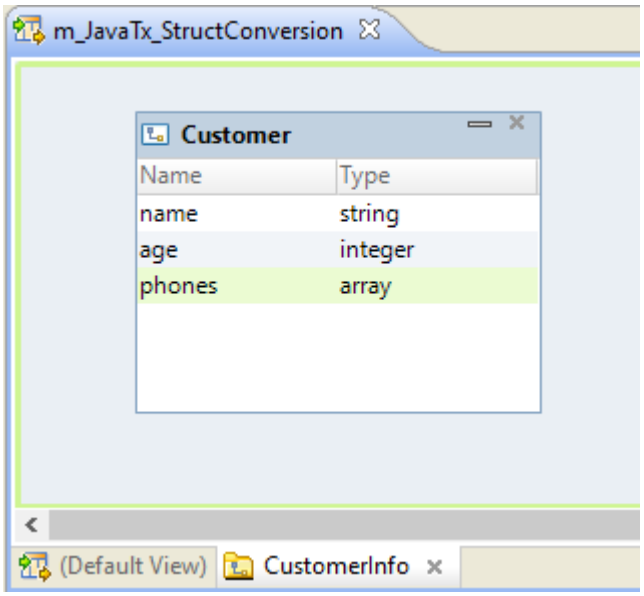
下图显示了包含读取转换、Java 转换和写入转换的映射。



在映射编辑器的“类型定义库”选项卡上，创建复杂的数据类型定义 Customer。复杂数据类型定义表示结构数据的架构。将类型定义库重命名为 CustomerInfo。将以下元素添加到复杂的数据类型定义：

- 字符串类型的名称
- 整数类型的年龄
- 具有字符串元素的数组类型的电话

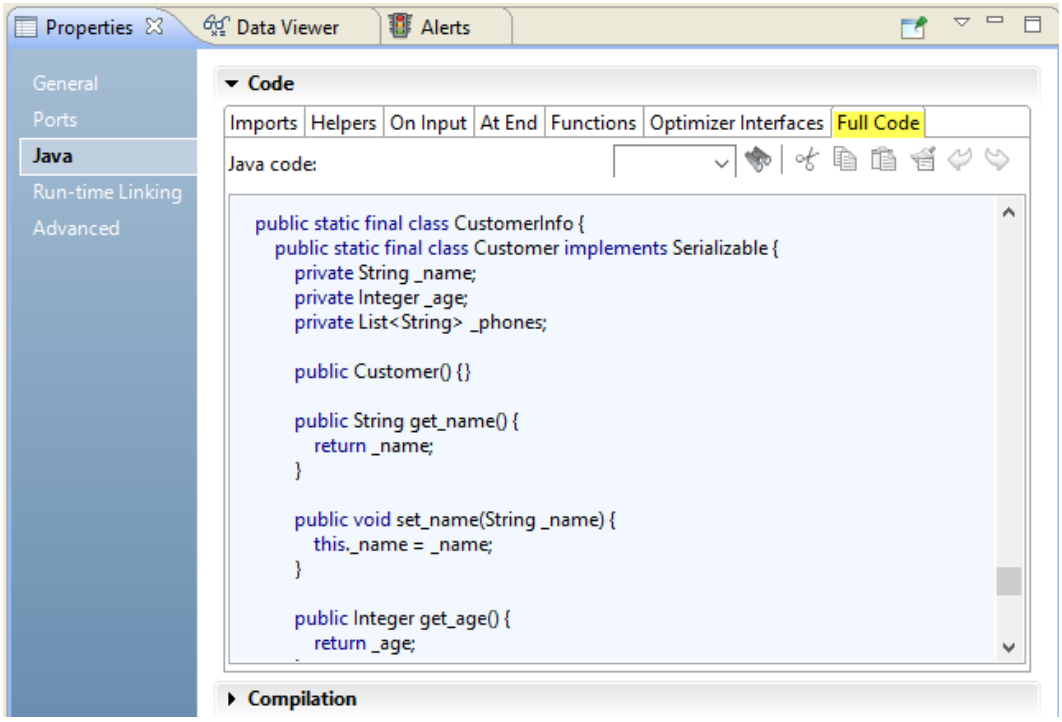
下图显示了类型定义库中的复杂数据类型定义：



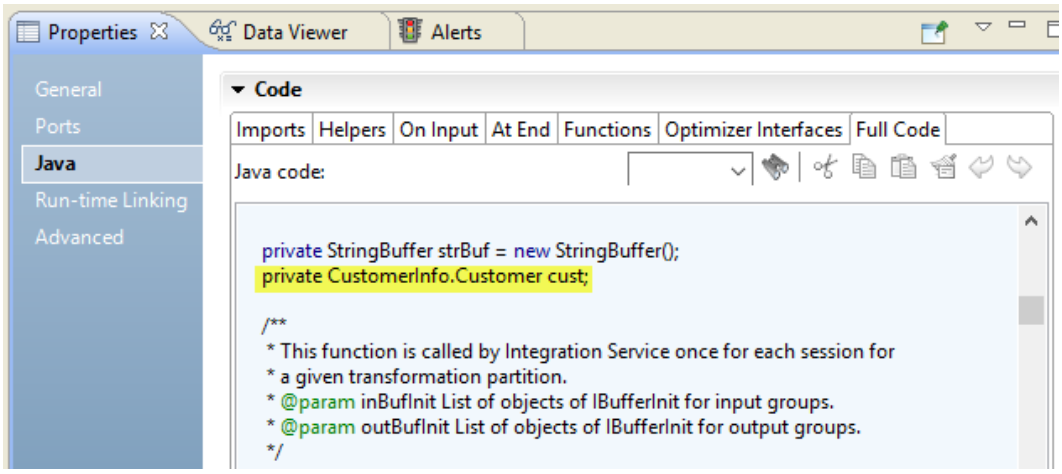
在 Java 转换中，添加一个结构输出端口并指定该端口的类型配置，以引用您创建的复杂数据类型定义。Java 转换生成具有 setter 和 getter 的 Customer 类，以读取和设置成员字段。该类包含以下成员字段：

- _name
- _age
- _phones

下图显示了在 **Java** 视图的**完整代码**选项卡中为结构端口创建的类：

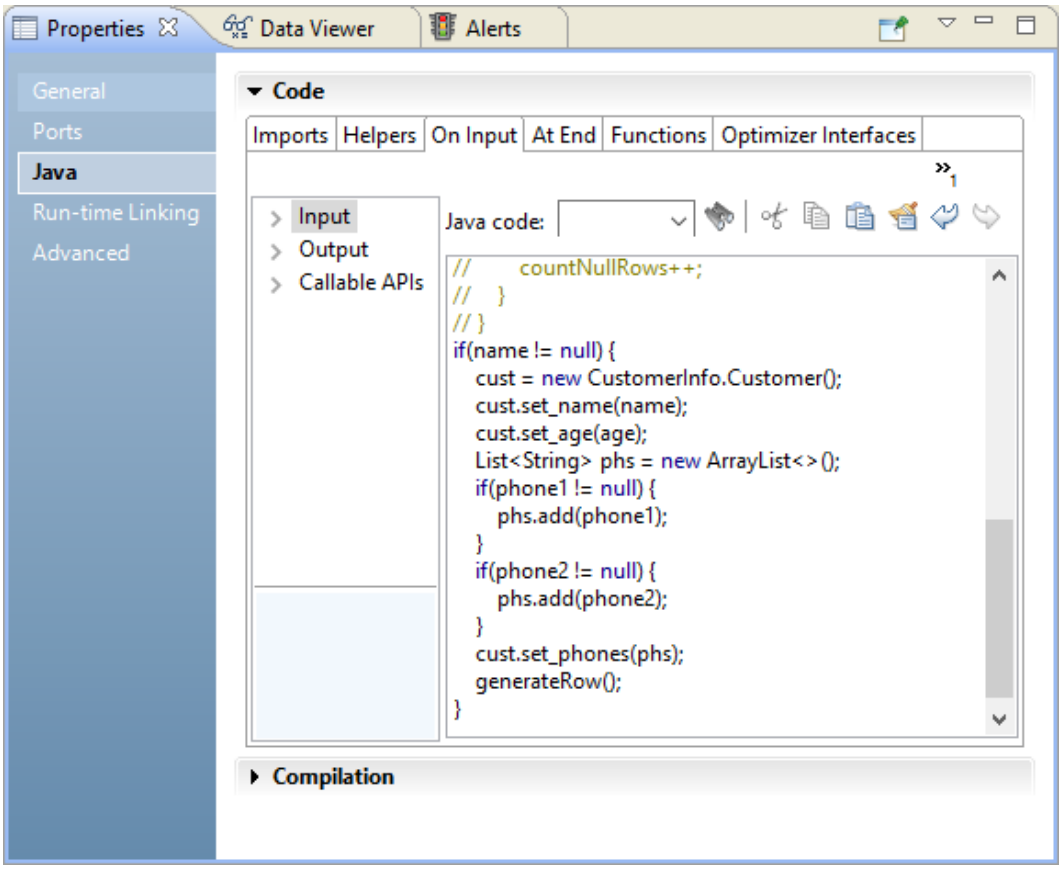


结构端口的 Java 数据类型使用类型定义库的名称和复杂数据类型定义。下图显示了生成代码中 cust 字段的 Java 数据类型名称 CustomerInfo.Customer：



在 Java 转换的 **Java** 视图中，导入转换需要的第三方、内置的或者自定义 Java 包。写入和编译 Java 代码，将平面数据转换为结构数据，在客户名称为空时删除客户行。

下图显示了**输入**选项卡中的代码：



在 Spark 引擎上验证映射并运行映射，将转换后的数据写入 Avro 文件输出。

非本地环境中的 Java 转换

非本地环境中的 Java 转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。不受支持。

Blaze 引擎上的 Java 转换

要在 Java 转换中使用外部 .jar 文件，请执行以下步骤：

1. 将外部 .jar 文件复制到数据集成服务计算机上的 Informatica 安装目录中的以下位置：<Informatica 安装目录>/services/shared/jars。然后再次运行数据集成服务。
2. 在托管 Developer tool 的计算机（也是您开发和运行含有 Java 转换的映射的计算机）上：
   a. 将外部 .jar 文件复制到本地计算机的目录中。
   b. 编辑 Java 转换，使其包含指向本地 .jar 文件的导入语句。
   c. 更新 Java 转换中的类路径。
   d. 编译转换。

Blaze 引擎的某些处理规则与数据集成服务的处理规则不同。

分区

- “转换范围”属性有以下限制：
  - “事务”值对转换范围无效。
  - 如果为分区键启用输入端口，则必须将转换范围设置为“全部输入”。
  - 如果转换范围为行，则必须启用“无状态”。

Spark 引擎上的 Java 转换

Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

一般限制

Spark 引擎对 Java 转换的支持存在以下限制：

- 当您将转换逻辑推送至 Hadoop 时，转换中的 Java 代码无法将输出写入至标准输出。Java 代码可以将输出写入至标准错误，该错误显示在日志文件中。
- 对于日期/时间值，Spark 引擎支持的最高精度为微秒。如果日期/时间值包含纳秒，尾数会被截断。

分区

Java 转换与分区配合使用时存在以下限制：

- 必须在 Java 转换中启用“可分区”属性。转换无法在一个分区中运行。
- “转换范围”属性有以下限制：
  - “事务”值对转换范围无效。
  - 如果为分区键启用输入端口，则必须将转换范围设置为“全部输入”。

- 如果转换范围为行，则必须启用“无状态”。

映射验证

在下列情况下，映射验证会失败：

- 从 Java 转换内的表达式中引用未连接的查找转换。
- 选择复杂数据类型端口作为分区或排序键。
- 在日期/时间中启用纳秒处理，而 Java 转换包含带 Date/Time 类型元素的复杂数据类型端口。例如，如果在日期/时间中启用纳秒处理，则 array<data/time> 类型端口无效。

在下列情况下，映射会失败：

- 当 Java 转换包含 Decimal 端口或带 Decimal 数据类型元素的复杂端口时，Java 转换和映射使用不同的精度模式。

  即使在映射中启用高精度，某些情况下映射也会以低精度模式处理数据，譬如映射包含带 Decimal 数据类型元素的复杂端口或者映射为流映射时。如果在 Java 转换和映射中均启用高精度，但映射以低精度模式处理数据，映射则会失败。
- 将二进制空字符传递至输出端口。为了避免映射失败，可以向 Java 转换添加代码，在将数据写入输出端口之前，将二进制空字符替换为其他字符。

使用外部 .jar 文件

要在 Java 转换中使用外部 .jar 文件，请执行以下步骤：

1. 将外部 .jar 文件复制到数据集成服务计算机上的 Informatica 安装目录中的以下位置：
   <Informatica 安装目录>/services/shared/jars
2. 再次应用数据集成服务。
3. 在托管 Developer tool 的计算机（也是您开发和运行含有 Java 转换的映射的计算机）上：
   a. 将外部 .jar 文件复制到本地计算机的目录中。
   b. 编辑 Java 转换，使其包含指向本地 .jar 文件的导入语句。
   c. 更新 Java 转换中的类路径。
   d. 编译转换。

流映射中的 Java 转换

流映射具有不适用于批处理映射的其他处理规则。

数据集成服务会忽略以下属性：

- 可分区。数据集成服务会忽略属性，并且可以使用多个线程处理转换。
- 无状态。数据集成服务会忽略属性，并且保留输入数据的行顺序。

第 20 章

Java 转换 API 引用

本章包括以下主题：

Java 转换 API 方法概览

在编辑器的 **Java** 视图的代码输入选项卡上，可以将 API 方法添加到 Java 代码中以定义转换行为。

要将 API 方法添加到代码中，请在代码输入选项卡上的导航器中展开**可调用 API** 列表，然后双击要添加到代码中的方法的名称。

此外，也可以将方法从导航器拖动到 Java 代码段中或在 Java 代码段中手动输入 API 方法。

可以将以下 API 方法添加至 Java 转换中的 Java 代码：

defineJExpression

定义 Java 表达式。

failSession

引发异常并显示错误消息，映射失败。

generateRow

为主动 Java 转换生成输出行。

getInRowType

返回转换中当前行的输入类型。

incrementErrorCount

递增映射的错误计数。

invokeJExpression

调用使用 defineJExpression 方法定义的 Java 表达式。

isNull

检查输入列中是否存在空值。

logError

将错误消息写入日志中。

logInfo

将信息性消息写入日志中。

resetNotification

如果数据集成服务计算机在重启模式下运行，则在映射运行后将重置 Java 代码中使用的变量。

setNull

将主动或被动 Java 转换中输出列的值设置为空值。

defineJExpression

定义表达式，包括表达式字符串和输入参数。 defineJExpression 方法的参数包括：一个 JExprParamMetadata 对象数组（包含输入参数）以及一个用于定义表达式语法的字符串值。

请使用以下语法：

```
defineJExpression(
    String expression,
    Object[] paramMetadataArray
    );
```

下表介绍了参数：

参数	类型	数据类型	说明
表达式	输入	String	表示表达式的字符串。
paramMetadataArray	输入	Object[]	JExprParaMetadata 对象数组，其中包含表达式的输入参数。

可以将 defineJExpression 方法添加到除**导入**和**函数**选项卡之外的任何代码输入选项卡上的 Java 代码中。

要使用 defineJExpression 方法，必须实例化 JExprParamMetadata 对象数组，该数组代表表达式的输入参数。 可以为这些参数设置元数据值，然后将该数组作为参数传递给 defineJExpression 方法。

例如，以下 Java 代码可创建一个表达式，用于查找两个字符串的值：

```
JExprParaMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata(EDataType.STRING, 20, 0);
params[1] = new JExprParamMetadata(EDataType.STRING, 20, 0);
defineJExpression(":lkp.mylookup(x1,x2)",params);
```

注意: 必须对连续传递给表达式的参数进行编号，这些参数必须以字母 x 开头。例如，要将三个参数传递给表达式，请将这些参数分别命名为 x1、x2 和 x3。

failSession

引发异常并显示错误消息，映射失败。

请使用以下语法：

```
failSession(String errorMessage);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
errorMessage	输入	String	错误消息字符串。

可使用 failSession 方法结束映射。 不要在代码输入选项卡上的 try/catch 块中使用 failSession 方法。

可以将 failSession 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

以下 Java 代码显示了如何测试 input1 输入端口是否为空值，如果是空值，则映射将失败：

```
if(isNull(" input1" )) {
     failSession( "Cannot process a null value for port input1." );
}
```

generateRow

为主动 Java 转换生成输出行。

请使用以下语法：

```
generateRow();
```

调用 generateRow 方法时，Java 转换将使用输出端口变量的当前值生成输出行。 如果要生成与某输入行对应的多个行，可以为每个输入行多次调用 generateRow 方法。 如果未在主动 Java 转换中使用 generateRow 方法，则该转换不会生成输出行。

可以将 generateRow 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

只能在主动转换中调用 generateRow 方法。 如果在被动转换中调用 generateRow 方法，则数据集成服务将生成错误。

使用以下 Java 代码可实现生成一个输出行、修改输出端口的值和生成其他输出行：

```
// Generate multiple rows.
if(!isNull("input1") && !isNull("input2"))
{
     output1 = input1 + input2;
     output2 = input1 - input2;
}
generateRow();
// Generate another row with modified values.
output1 = output1 * 2;
output2 = output2 * 2;
generateRow();
```

getInRowType

返回转换中当前行的输入类型。 该方法返回要插入、更新、删除或拒绝的值。

请使用以下语法：

```
rowType getInRowType();
```

下表介绍了参数：

参数	参数类型	数据类型	说明
rowType	输出	String	返回更新策略类型，该类型是以下值之一： - DELETE - INSERT - REJECT - UPDATE

可以将 getInRowType 方法添加到**输入**代码输入选项卡上的 Java 代码中。

可以在配置为设置更新策略的主动转换中使用 getInRowType 方法。 如果在未配置为设置更新策略的主动转换中调用此方法，则数据集成服务将生成错误。

getMetadata

在运行时检索 Java 转换元数据。 getMetadata 方法检索使用 storeMetadata 方法保存的元数据，例如从优化器传递至 pushFilter 函数中 Java 转换的筛选条件。

请使用以下语法：

```
getMetadata (String key);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
键	输入	String	确定元数据。 getMetadata 方法使用键来确定要检索的元数据。

可以将 getMetadata 方法添加到以下代码输入选项卡上的 Java 代码中：

- 帮助程序
- 输入
- 结尾
- 优化器接口
- 函数

可以配置 getMetadata 方法检索用于推入优化的筛选条件。 getMetadata 方法可以从优化器中检索存储的每个筛选条件。

```
// Retrieve a filter condition
String mydata = getMetadata ("FilterKey");
```

incrementErrorCount

递增的错误计数。 如果错误计数达到错误阈值，则映射将失败。

请使用以下语法：

```
incrementErrorCount(int nErrors);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
nErrors	输入	Integer	用于递增错误计数的数字。

可以将 incrementErrorCount 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

以下 Java 代码显示了转换的输入端口存在空值时如何递增错误计数：

```
// Check if input employee id and name is null.
if (isNull ("EMP_ID_INP") || isNull ("EMP_NAME_INP"))
{
     incrementErrorCount(1);
     // if input employee id and/or name is null, don't generate a output row for this input row
     generateRow = false;
}
```

invokeJExpression

调用表达式并返回该表达式的值。

请使用以下语法：

```
(datatype)invokeJExpression(
                        String expression,
                        Object[] paramMetadataArray);
```

invokeJExpression 方法的输入参数是字符串值，该值表示表达式和包含表达式输入参数的对象数组。

下表介绍了参数：

参数	参数类型	数据类型	说明
表达式	输入	String	表示表达式的字符串。
paramMetadataArray	输入	Object[]	包含表达式的输入参数的对象数组。

可以将 invokeJExpression 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

使用 invokeJExpression 方法时，请遵循以下规则和准则：

- 返回数据类型。 invokeJExpression 方法的返回数据类型是一个对象。 您必须使用相应的数据类型转换函数的返回值。

  可以返回数据类型为 Integer、Double、String 和 byte[] 的值。
- 行类型。 invokeJExpression 方法的返回值的行类型为 INSERT。

  要为该返回值使用其他行类型，请使用高级接口。

- 空值。 如果将空值作为参数传递或者 invokeJExpression 方法的返回值为空，则将该值视为空指示符。

  例如，如果表达式的返回值为空且返回数据类型为 String，则返回一个具有空值的字符串。
- Date 数据类型。 必须将数据类型为 Date 的输入参数转换为 String 数据类型。

  要将表达式中的字符串用作 Date 数据类型，请使用 to_date() 函数将字符串转换为 Date 数据类型。

  或者，您必须将返回 Date 数据类型的所有表达式的返回类型转换为 String 数据类型。

以下示例将字符串“John”与“Smith”连接，并返回字符串“John Smith”：

```
(String)invokeJExpression("concat(x1,x2)", new Object [] { "John ", "Smith" });
```

注意: 必须对连续传递给表达式的参数进行编号，这些参数必须以字母 x 开头。例如，要将三个参数传递给表达式，请将参数分别命名为 x1、x2 和 x3。

isNull

检查输入列的值是否为空值。

请使用以下语法：

```
Boolean isNull(String satrColName);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
strColName	输入	String	输入列的名称。

可以将 isNull 方法添加到**输入**代码输入选项卡上的 Java 代码中。

以下 Java 代码显示了在将 SALARY 输入列添加至 totalSalaries 实例变量之前如何检查其值是否为空值：

```
// if value of SALARY is not null
if (!isNull("SALARY")) {
     // add to totalSalaries
     TOTAL_SALARIES += SALARY;
}
```

或者，也可以使用以下 Java 代码获得相同结果：

```
// if value of SALARY is not null
String strColName = "SALARY";
if (!isNull(strColName)) {
     // add to totalSalaries
     TOTAL_SALARIES += SALARY;
}
```

logError

将错误消息写入日志中。

请使用以下语法：

```
logError(String msg);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
msg	输入	String	错误消息字符串。

可以将 logError 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

以下 Java 代码显示了输入端口为空值时如何记录错误：

```
// check BASE_SALARY
if (isNull("BASE_SALARY")) {
     logError("Cannot process a null salary field.");
}
```

当代码运行时，日志中将显示以下消息：

```
[JTX_1013] [ERROR] Cannot process a null salary field.
```

logInfo

将信息性消息写入日志中。

请使用以下语法：

```
logInfo(String msg);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
msg	输入	String	信息消息字符串。

可以将 logInfo 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

以下 Java 代码显示了在 Java 转换处理达到消息阈值（1000 行）后如何将消息写入日志：

```
if (numRowsProcessed == messageThreshold) {
     logInfo("Processed " + messageThreshold + " rows.");
}
```

resetNotification

如果数据集成服务计算机在重启模式下运行，则在映射运行后将重置 Java 代码中使用的变量。

在重新启动模式中，数据集成服务无法取消初始化，但可以在请求后对其进行重置以便数据集成服务可以处理下一个请求。

对于 Java 转换，可以在运行映射之后使用 resetNotification 方法来重置 Java 代码中的变量。

请使用以下语法：

```
public int resetNotification(IGroup group) {
    return EStatus.value;
}
```

下表介绍了参数：

参数	参数类型	数据类型	说明
int	输出	EStatus.*value*	返回值，其中 *value* 是以下值之一： - SUCCESS. 成功。 - FAILURE. 失败。 - NOIMPL. 未实现。
组	输入	IGroup	输入组。

在**帮助程序**选项卡的代码输入选项卡上，可以将 resetNotification 方法添加到 Java 代码。

resetNotification 方法不会显示在“可调用 API”列表中。

例如，假设 Java 代码声明名为 out5_static 的静态变量并将其初始化为 1，则下次运行映射后，以下 Java 代码会将 out5_static 变量重置为 1。

```
public int resetNotification(IGroup group) {
    out5_static=1;
    return EStatus.SUCCESS;
}
```

该方法不是必需的。 但是，如果数据集成服务在重新启动模式中运行，并且映射包含未实现 resetNotification 方法的 Java 转换，则日志中将显示 JSDK_42075 警告消息。

setNull

将主动或被动 Java 转换中的输出列值设置为空值。

请使用以下语法：

```
setNull(String strColName);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
strColName	输入	String	输出列名称。

setNull 方法可以将主动或被动 Java 转换中的输出列值设置为空值。 如果将输出列值设置为空值，则生成输出行后才能修改该值。

可以将 setNull 方法添加到任意代码输入选项卡（除**导入**和**函数**选项卡之外）上的 Java 代码。

以下 Java 代码显示了如何检查输入列值并将相应的输出列值设置为空值：

```
// check value of Q3RESULTS input column
if(isNull("Q3RESULTS")) {
     // set the value of output column to null
     setNull("RESULTS");
}
```

或者，可以使用以下 Java 代码获得相同效果：

```
// check value of Q3RESULTS input column
String strColName = "Q3RESULTS";
if(isNull(strColName)) {
     // set the value of output column to null
```

```
        setNull(strColName);
}
```

storeMetadata

存储可以在运行时使用 getMetadata 方法检索的 Java 转换元数据。

请使用以下语法：

```
storeMetadata (String key String data);
```

下表介绍了参数：

参数	参数类型	数据类型	说明
键	输入	String	确定元数据。 storeMetadata 方法需要一个用于标识元数据的键。 将该键定义为任何字符串。
数据	输入	String	要存储为 Java 转换元数据的数据。

可以将 storeMetadata 方法添加到以下代码输入选项卡上的 Java 代码：

- 帮助程序
- 输入
- 结尾
- 优化器接口
- 函数

通过在主动转换中配置 storeMetadata 方法可以接受用于推入优化的筛选条件。 storeMetadata 方法会存储优化器从映射推送到 Java 转换的筛选条件。

```
// Store a filter condition
storeMetadata ("FilterKey", condition);
```

第 21 章

Java 表达式

本章包括以下主题：

Java 表达式概览

可以在 Java 转换中使用 Java 编程语言调用表达式。

使用表达式可扩展 Java 转换的功能。 例如，可以通过在 Java 转换中调用表达式来查找输入或输出端口的值，或者查找 Java 转换变量的值。

要在 Java 转换中调用表达式，请生成 Java 代码或使用 Java 转换 API 方法来调用表达式。 调用表达式后，可在相应的代码输入选项卡上使用该表达式的结果。 可以生成调用表达式的 Java 代码，或使用 API 方法编写调用表达式的 Java 代码。

下表介绍了可用于在 Java 转换中创建和调用表达式的方法：

方法	说明
定义函数对话框	可用于创建函数以调用表达式，并为表达式生成代码。
简单接口	可用于调用一个 API 方法以调用表达式并获取该表达式的结果。
高级接口	可用于定义表达式、调用表达式和使用表达式的结果。 如果您熟悉面向对象的编程，并且希望对表达式调用操作进行更多控制，请使用高级接口。

表达式函数类型

、使用**定义函数**对话框或使用简单或高级接口，可以为 Java 转换创建表达式。

可以在输入的表达式中使用输入或输出端口变量或 Java 代码中的变量作为输入参数。

如果使用**定义函数**对话框，则在 Java 转换中使用表达式之前可以验证该表达式。

可以在 Java 转换中调用以下类型的表达式函数：

表达式函数类型	说明
转换语言函数	类似于 SQL 的函数，用于处理通用表达式。
用户定义的函数	基于转换语言函数在 Developer 工具中创建的函数。
自定义函数	使用自定义函数 API 创建的函数。

还可以在表达式中使用未连接的转换和内置变量。 例如，可以在表达式中使用未连接的查找转换。

使用定义函数对话框来定义表达式

定义 Java 表达式时，请配置函数、创建表达式并生成调用该表达式的代码。

可在**定义函数**对话框中定义函数和创建表达式。

要创建一个表达式函数并在 Java 转换中使用该表达式，请完成以下高级别任务：

1. 配置调用表达式的函数，包括函数名称、说明和参数。 创建表达式时，请使用函数参数。
2. 创建表达式语法并验证该表达式。
3. 生成调用表达式的 Java 代码。

   Developer 将该代码放置在**函数**代码输入选项卡上。

生成 Java 代码后，基于您使用的是简单接口还是高级接口，在相应的代码输入选项卡上调用生成的函数，以便调用一个表达式或获得一个 JExpression 对象。

注意: 要在创建表达式时验证该表达式，必须使用**定义函数**对话框。

步骤 1。 配置函数

为调用表达式的 Java 函数配置函数名称、说明和输入参数。

配置函数时，请使用以下规则和准则：

- 使用不与转换或预留的 Java 关键字中现有 Java 函数冲突的唯一函数名称。
- 必须配置参数名称、Java 数据类型、精度和小数位数。 输入参数是调用转换的 Java 代码中的函数时传递的值。
- 要将 Date 数据类型传递给表达式，请对输入参数使用 String 数据类型。

  如果表达式返回 Date 数据类型，可以将返回值用作简单接口中的 String 数据类型以及高级接口中的 String 或 long 数据类型。

步骤 2。 创建并验证表达式

创建表达式时，请使用为函数配置的参数。

您还可以在表达式中使用转换语言函数、自定义函数或其他用户定义函数。 您可以在**定义函数**对话框中创建并验证表达式。

步骤 3。 生成表达式的 Java 代码

配置函数和函数参数并定义和验证表达式后，可以生成调用表达式的 Java 代码。

Developer 将生成的 Java 代码置于**函数**代码输入选项卡上。 使用生成的 Java 代码调用函数，该函数调用代码输入选项卡中的表达式。 您可以生成简单或高级 Java 代码。

生成调用表达式的 Java 代码后，无法编辑表达式或重新对其进行验证。 要在代码生成后修改表达式，必须重新创建表达式。

使用定义函数对话框创建表达式并生成 Java 代码

使用**定义函数**对话框可以创建调用表达式的函数。

要创建调用表达式的函数，请完成以下步骤：

1. 在 Developer 中，打开 Java 转换或创建新的 Java 转换。
2. 在 **Java 代码**选项卡中，单击**新建函数**。

   此时将显示**定义函数**对话框。
3. 输入函数名称。
4. 或者，输入表达式的说明。

   最多可输入 2,000 个字符。
5. 为函数创建参数。

   创建参数时，请配置参数名称、数据类型、精度和小数位数。
6. 在**表达式**选项卡上，使用创建的参数创建表达式。
7. 要验证表达式，请单击**验证**。
8. 或者，在**表达式**框中输入表达式。 然后单击**验证**来验证表达式。
9. 要使用高级接口生成 Java 代码，请选择**生成高级代码**选项。 然后单击**生成**。

   Developer 将在**函数**代码输入选项卡中生成调用表达式的函数。

Java 表达式模板

可以使用表达式的简单或高级 Java 代码为表达式生成 Java 代码。

表达式的 Java 代码是基于表达式的模板生成的。

以下示例显示了为简单 Java 代码生成的 Java 表达式的模板：

```
Object function_name (Java datatype x1[,
                                    Java datatype x2 ...] )
                                    throws SDK Exception
{
return (Object)invokeJExpression( String expression,
                                  new Object [] { x1[, x2, ... ]}  );
}
```

以下示例显示了使用高级接口生成的 Java 表达式的模板：

```
JExpression function_name ()  throws SDKException
{
    JExprParamMetadata params[] = new JExprParamMetadata[number of parameters];
    params[0] = new JExprParamMetadata (
                    EDataType.STRING,  // data type
                    20,  // precision
                    0    // scale
                    );
...
    params[number of parameters - 1] = new JExprParamMetadata (
```

```
                    EDataType.STRING,  // data type
                    20,  // precision
                    0    // scale
                    );
...
    return defineJExpression(String expression,params);
}
```

使用简单接口

使用 invokeJExpression Java API 方法调用简单接口中的表达式。

invokeJExpression

调用表达式并返回该表达式的值。

请使用以下语法：

```
(datatype)invokeJExpression(
                    String expression,
                    Object[] paramMetadataArray);
```

invokeJExpression 方法的输入参数是字符串值，该值表示表达式和包含表达式输入参数的对象数组。

下表介绍了参数：

参数	参数类型	数据类型	说明
表达式	输入	String	表示表达式的字符串。
paramMetadataArray	输入	Object[]	包含表达式的输入参数的对象数组。

可以将 invokeJExpression 方法添加到除**导入**和**函数**选项卡之外的任意代码输入选项卡上的 Java 代码中。

使用 invokeJExpression 方法时，请遵循以下规则和准则：

- 返回数据类型。 invokeJExpression 方法的返回数据类型是一个对象。 您必须使用相应的数据类型转换函数的返回值。

  可以返回数据类型为 Integer、Double、String 和 byte[] 的值。
- 行类型。 invokeJExpression 方法的返回值的行类型为 INSERT。

  要为该返回值使用其他行类型，请使用高级接口。
- 空值。 如果将空值作为参数传递或者 invokeJExpression 方法的返回值为空，则将该值视为空指示符。

  例如，如果表达式的返回值为空且返回数据类型为 String，则返回一个具有空值的字符串。
- Date 数据类型。 必须将数据类型为 Date 的输入参数转换为 String 数据类型。

  要将表达式中的字符串用作 Date 数据类型，请使用 to_date() 函数将字符串转换为 Date 数据类型。

  或者，您必须将返回 Date 数据类型的所有表达式的返回类型转换为 String 数据类型。

注意: 必须对连续传递给表达式的参数进行编号，这些参数必须以字母 x 开头。例如，要将三个参数传递给表达式，请将参数分别命名为 x1、x2 和 x3。

简单接口示例

您可以定义并调用表达式，这些表达式使用**帮助程序**和**输入**代码输入选项卡上的 invokeJExpression API 方法。

以下示例显示如何在 Java 转换的 NAME 和 ADDRESS 输入端口完成查找，并将返回值分配给 COMPANY_NAME 输出端口。

在**输入**代码输入选项卡上输入以下代码：

```
COMPANY_NAME = (String)invokeJExpression(":lkp.my_lookup(X1,X2)", new Object [] {str1 ,str2} );
generateRow();
```

使用高级接口

在高级接口中，可以使用面向对象的 API 方法来定义、调用并获得表达式的结果。

下表介绍了高级接口中可用的类和 API 方法：

类或 API 方法	说明
EDataType 类	枚举表达式的数据类型。
JExprParamMetadata 类	包含一个表达式中每个参数的元数据。 参数元数据包含数据类型、精度和小数位数。
defineJExpression API 方法	定义表达式。 包含表达式字符串和参数。
invokeJExpression API 方法	调用表达式。
JExpression 类	包含用于创建、调用、获得元数据、获得表达式结果以及检查返回数据类型的方法。

通过高级接口调用表达式

可以使用高级接口来定义和调用表达式以及获取表达式的结果。

1. 在**帮助程序**或**输入**代码输入选项卡上，为表达式的每个参数创建一个 JExprParamMetadata 类实例并设置元数据的值。 或者，可以在 defineJExpression 方法中实例化 JExprParamMetadata 对象。
2. 使用 defineJExpression 方法获取表达式的 JExpression 对象。
3. 在相应的代码输入选项卡上，使用 invokeJExpression 方法调用表达式。
4. 使用 isResultNull 方法检查返回值的结果。
5. 可以使用 getResultDataType 和 getResultMetadata 方法获取返回值的数据类型或返回值的元数据。
6. 使用相应的 API 方法获取表达式的结果。 可以使用 getInt、getDouble、getStringBuffer 和 getBytes 方法。

使用高级接口的规则和准则

使用高级接口时，您必须了解规则和准则。

使用以下规则和准则：

- 如果您将空值作为参数传递或表达式结果为空值，该值将被视为空指示器。 例如，如果表达式的结果为空值且返回的数据类型为 String，则返回的字符串为空值。 您可以使用 isResultNull 方法检查表达式的结果。
- 您必须先将 Date 数据类型的输入参数转换为 String，才可以将其用于表达式。 要将表达式中的字符串用作 Date 数据类型，请使用 to_date() 函数将字符串转换为 Date 数据类型。

  您可以获取表达式的结果，该表达式返回的 Date 数据类型为 String 或 long 数据类型。

  使用 getStringBuffer 方法可以获取表达式的结果，该表达式返回的 Date 数据类型为 String 数据类型。 使用 getLong 方法可以获取表达式的结果，该表达式返回的 Date 数据类型为 long 数据类型。

EDataType 类

枚举表达式中所使用的 Java 数据类型。 获取表达式的返回数据类型，或分配 JExprParamMetadata 对象中参数的数据类型。 无需实例化 EDataType 类。

下表列出了表达式中 Java 数据类型的枚举值：

数据类型	枚举值
INT	1
DOUBLE	2
STRING	3
BYTE_ARRAY	4
DATE_AS_LONG	5

以下 Java 代码示例显示如何使用 EDataType 类向 JExprParamMetadata 对象分配 String 数据类型：

```
JExprParamMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata (
                EDataType.STRING,  // data type
                20,  // precision
                0  // scale
            );
...
```

JExprParamMetadata 类

实例化代表表达式参数的对象，并为这些参数设置元数据。

可以使用 JExprParamMetadata 对象数组作为 defineJExpression 方法的输入来为输入参数设置元数据。 可以在**函数**代码输入选项卡上或在 defineJExpression 中创建 JExprParamMetadata 对象的实例。

请使用以下语法：

```
JExprParamMetadata paramMetadataArray[] = new JExprParamMetadata[numberOfParameters];
paramMetadataArray[0] = new JExprParamMetadata(datatype, precision, scale);
...
paramMetadataArray[numberofParameters - 1] = new JExprParamMetadata(datatype, precision, scale);;
```

下表介绍了以下参数：

参数	参数类型	参数数据类型	说明
数据类型	输入	EDataType	参数的数据类型。
精度	输入	整数	参数的精度。
小数位数	输入	整数	参数的小数位数。

例如，通过以下 Java 代码可使用数据类型 String、精度 20、小数位数 0 实例化由两个 JExprParamMetadata 对象组成的数组：

```
JExprParamMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata(EDataType.STRING, 20, 0);
params[1] = new JExprParamMetadata(EDataType.STRING, 20, 0);
return defineJExpression(":LKP.LKP_addresslookup(X1,X2)",params);
```

defineJExpression

定义表达式，包括表达式字符串和输入参数。 defineJExpression 方法的参数包括：一个 JExprParamMetadata 对象数组（包含输入参数）以及一个用于定义表达式语法的字符串值。

请使用以下语法：

```
defineJExpression(
    String expression,
    Object[] paramMetadataArray
    );
```

下表介绍了参数：

参数	类型	数据类型	说明
表达式	输入	String	表示表达式的字符串。
paramMetadataArray	输入	Object[]	JExprParaMetadata 对象数组，其中包含表达式的输入参数。

可以将 defineJExpression 方法添加到除**导入**和**函数**选项卡之外的任何代码输入选项卡上的 Java 代码中。

要使用 defineJExpression 方法，必须实例化 JExprParamMetadata 对象数组，该数组代表表达式的输入参数。 可以为这些参数设置元数据值，然后将该数组作为参数传递给 defineJExpression 方法。

例如，以下 Java 代码可创建一个表达式，用于查找两个字符串的值：

```
JExprParaMetadata params[] = new JExprParamMetadata[2];
params[0] = new JExprParamMetadata(EDataType.STRING, 20, 0);
params[1] = new JExprParamMetadata(EDataType.STRING, 20, 0);
defineJExpression(":lkp.mylookup(x1,x2)",params);
```

注意: 必须对连续传递给表达式的参数进行编号，这些参数必须以字母 x 开头。例如，要将三个参数传递给表达式，请将这些参数分别命名为 x1、x2 和 x3。

JExpression 类

包含多个方法，可用于创建和调用表达式、返回表达式的值以及检查返回的数据类型。

下表列出了 JExpression 类包含的方法：

方法名称	说明
invoke	调用表达式。
getResultDataType	返回表达式结果的数据类型。
getResultMetadata	返回表达式结果的元数据。
isResultNull	检查表达式结果的结果值。
getInt	以 Integer 数据类型返回表达式结果的值。
getDouble	以 Double 数据类型返回表达式结果的值。
getStringBuffer	以 String 数据类型返回表达式结果的值。
getBytes	以 byte[] 数据类型返回表达式结果的值。

高级接口示例

可以使用高级接口在 Java 转换中创建并调用查找表达式。

以下 Java 代码示例说明了如何创建一个函数来调用表达式以及如何调用该表达式以获取返回值。 本示例将具有 String 数据类型的两个输入端口（NAME 和 COMPANY）的值传递到函数 myLookup。 myLookup 函数使用查找表达式来查找 ADDRESS 输出端口的值。

注意: 本示例假定在名为 LKP_addresslookup 的映射中有一个未连接的查找转换。

在**帮助程序**选项卡上使用以下 Java 代码：

```
JExpression addressLookup()  throws SDKException
{
     JExprParamMetadata params[] = new JExprParamMetadata[2];
     params[0] = new JExprParamMetadata (
                    EDataType.STRING,         // data type
                    50,                  // precision
                    0                    // scale
                    );
     params[1] = new JExprParamMetadata (
                    EDataType.STRING,     // data type
                    50,                   // precision
                    0                     // scale
                    );
     return defineJExpression(":LKP.LKP_addresslookup(X1,X2)",params);
}
JExpression lookup = null;
boolean isJExprObjCreated = false;
```

在**输入**选项卡上使用以下 Java 代码调用表达式并返回 ADDRESS 端口的值：

```
...
if(!iisJExprObjCreated)
{
    lookup = addressLookup();
    isJExprObjCreated = true;
}
lookup = addressLookup();
```

```
lookup.invoke(new Object [] {NAME,COMPANY}, ERowType.INSERT);
EDataType addressDataType = lookup.getResultDataType();
if(addressDataType == EDataType.STRING)
{
    ADDRESS = (lookup.getStringBuffer()).toString();
} else {
    logError("Expression result datatype is incorrect.");
}
...
```

JExpression 类 API 引用

JExpression 类包含多个 API 方法，可用于创建和调用表达式、返回表达式的值以及检查返回值的数据类型。

JExpression 类包含以下 API 方法：

- getBytes
- getDouble
- getInt
- getLong
- getResultDataType
- getResultMetadata
- getStringBuffer
- invoke
- isResultNull

getBytes

以 byte[] 数据类型返回表达式结果的值。 获取一个表达式的结果，该表达式可使用 AES_ENCRYPT 函数对数据进行加密。

请使用以下语法：

```
objectName.getBytes();
```

使用下面的示例 Java 代码可获取以下表达式的结果：该表达式使用 AES_ENCRYPT 函数对二进制数据进行加密（其中，JExprEncryptData 是一个 JExpression 对象）：

```
byte[] newBytes = JExprEncryptData.getBytes();
```

getDouble

以 Double 数据类型返回表达式结果的值。

请使用以下语法：

```
objectName.getDouble();
```

使用下面的示例 Java 代码可获取以下表达式的结果：该表达式以 Double 数据类型返回薪资值（其中，JExprSalary 是一个 JExpression 对象）：

```
double salary = JExprSalary.getDouble();
```

getInt

以 Integer 数据类型返回表达式结果的值。

请使用以下语法：

```
objectName.getInt();
```

例如，使用下面的 Java 代码可获取以下表达式的结果：该表达式以 Integer 数据类型返回员工 ID 编号（其中，findEmpID 是一个 JExpression 对象）：

```
int empID = findEmpID.getInt();
```

getLong

以 Long 数据类型返回表达式结果的值。 获取使用 Date 数据类型的表达式的结果。

请使用以下语法：

```
objectName.getLong();
```

使用下面的示例 Java 代码可获取以下表达式的结果：该表达式以 Long 数据类型返回日期值（其中，JExprCurrentDate 是一个 JExpression 对象）：

```
long currDate = JExprCurrentDate.getLong();
```

getResultDataType

返回表达式结果的数据类型。 返回 EDataType 的值。

请使用以下语法：

```
objectName.getResultDataType();
```

使用下面的示例 Java 代码可调用表达式并将结果的数据类型赋值给变量 dataType：

```
myObject.invoke(new Object[] { NAME,COMPANY }, ERowType INSERT);
EDataType dataType = myObject.getResultDataType();
```

getResultMetadata

返回表达式结果的元数据。 可以使用 getResultMetadata 获取表达式结果的精度、小数位数和数据类型。 可以将表达式返回值的元数据赋值给 JExprParamMetadata 对象。 使用 getScale、getPrecision 和 getDataType 对象方法可检索结果元数据。

请使用以下语法：

```
objectName.getResultMetadata();
```

使用下面的示例 Java 代码可将 myObject 返回值的小数位数、精度和数据类型赋值给变量：

```
JExprParamMetadata myMetadata = myObject.getResultMetadata();
int scale = myMetadata.getScale();
int prec = myMetadata.getPrecision();
int datatype = myMetadata.getDataType();
```

注意: getDataType 对象方法将按 EDataType 中的枚举内容返回数据类型的整数值。

getStringBuffer

以 String 数据类型返回表达式结果的值。

请使用以下语法：

```
objectName.getStringBuffer();
```

使用下面的示例 Java 代码可获取以下表达式的结果：该表达式将返回两个连接的字符串（其中，JExprConcat 是一个 JExpression 对象）：

```
String result = JExprConcat.getStringBuffer();
```

invoke

调用表达式。invoke 的参数中包括用于定义输入参数和行类型的对象。必须先实例化 JExpression 对象，才可以使用 invoke 方法。对于行类型，请使用 ERowType.INSERT、ERowType.DELETE 和 ERowType.UPDATE。

请使用以下语法：

```
objectName.invoke(
    new Object[] { param1[, ... paramN ]},
    rowType
    );
```

下表介绍了以下参数：

参数	数据类型	输入/输出	说明
objectName	JExpression	输入	JExpression 对象名称。
参数	-	输入	包含表达式输入值的对象数组。

例如，在**函数**代码输入选项卡上创建了名为 address_lookup() 的函数，该函数返回表示该表达式的 JExpression 对象。使用下面的代码可调用使用输入端口 NAME 和 COMPANY 的表达式：

```
JExpression myObject = address_lookup();
myObject.invoke(new Object[] { NAME,COMPANY }, ERowType INSERT);
```

isResultNull

检查表达式结果值。

请使用以下语法：

```
objectName.isResultNull();
```

使用下面的示例 Java 代码可调用一个表达式，并且在表达式的返回值不为空的情况下将返回值赋值给变量 address：

```
JExpression myObject = address_lookup();
myObject.invoke(new Object[] { NAME,COMPANY }, ERowType INSERT);
if(!myObject.isResultNull()) {
     String address = myObject.getStringBuffer();
}
```

第 22 章

联接器转换

本章包括以下主题：

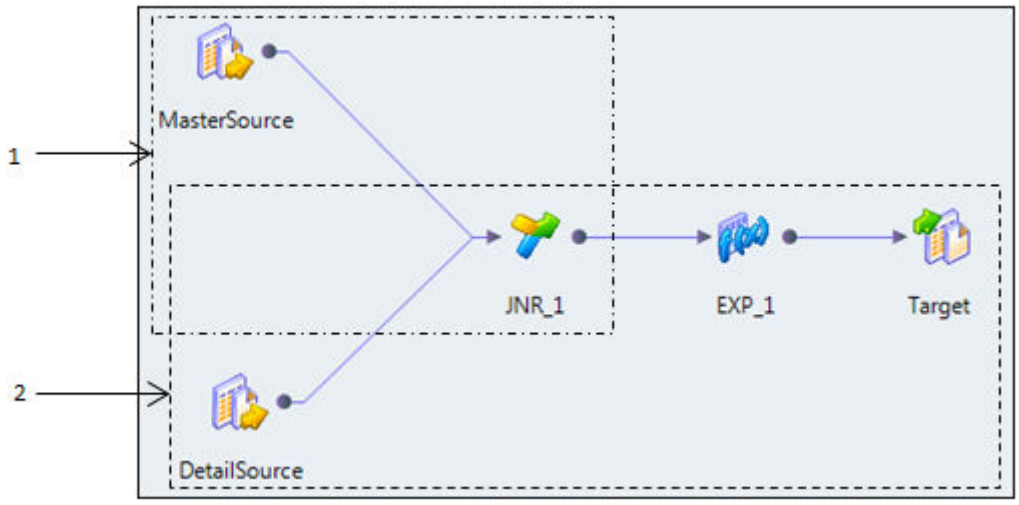
联接器转换概览

联接器转换联接来自不同位置或不同文件系统的两个相关异构源中的源数据。还可以从同一源联接数据。联接器转换属于多组主动转换。

联接器转换会将源与至少一个匹配列联接。联接器转换将使用与两个源之间的一个或多个列对相匹配的条件。

两个输入管道中包括一个主管道和一个详细管道或一个主管道和一个详细分支。主管道结束于联接器转换，但详细管道将继续连接到目标。

下图显示了具有联接器转换的映射中的主管道和详细管道：



1. 主管道
2. 详细管道

要联接映射中两个以上的源，请将联接器转换中的输出与其他源管道联接。将联接器转换添加到映射中，直到您已联接所有源管道。

联接器转换高级属性

配置有助于确定数据集成服务如何为联接器转换处理数据的属性。

在**高级**选项卡上配置以下属性：

联接器数据缓存大小

数据集成服务在映射运行开始时为转换的数据缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

联接器索引缓存大小

数据集成服务在映射运行开始时为转换的索引缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

缓存目录

数据集成服务创建索引缓存文件和数据缓存文件的目录。请确认该目录存在并含有足够的磁盘空间可用于缓存文件。

输入多个以分号分隔的目录，以提高缓存分区期间的性能。缓存分区会为处理转换的每个分区创建一个单独的缓存。

默认值为 CacheDir 系统参数。您可以为此属性配置另一个系统参数或用户定义的参数。

已排序输入

指示输入数据已按组预先排序。 选择“已排序输入”联接已排序数据。 使用已排序输入可以提高性能。

主排序顺序

指定主源数据的排序顺序。 如果主源数据按升序排列，则选择“升序”。 如果选择“升序”，也会启用已排序输入。 默认为“自动”。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

相关主题：

- “缓存大小” 页面上 67

联接器缓存

当您运行使用联接器转换的映射时，数据集成服务将在内存中创建索引缓存和数据缓存以运行该转换。如果数据集成服务需要的空间大于内存缓存中的可用空间，它会将溢出数据存储在缓存文件中。

当您运行使用联接器转换的映射时，数据集成服务将同时从主源和详细信息源中读取行，并基于主行来构建索引缓存和数据缓存。数据集成服务基于详细信息源数据和已缓存的主数据执行联接。

联接器转换的类型决定了数据集成服务在缓存中存储的行数。

下表介绍了数据集成服务在缓存中为不同类型的联接器转换存储的信息：

联接器转换类型	索引缓存	数据缓存
未排序输入	将具有唯一索引键的所有主行存储在联接条件中。	存储所有主行。
具有不同源的已排序输入	将具有唯一索引键的 100 条主行存储在联接条件中。	存储与存储在索引缓存中的行对应的主行。 如果主数据包含多个具有相同键的行，数据集成服务在数据缓存中存储的行数将超过 100 行。
具有相同源的已排序输入	将具有唯一键的所有主行或详细信息行存储在联接条件中。 如果数据集成服务处理详细信息管道的速度比处理主管道的速度快，则存储详细信息行。否则，存储主行。 集成服务存储的行数取决于主管道和详细信息管道的处理速率。 如果一个管道处理其数据行的速度比另一管道快，数据集成服务将缓存所有已处理的行。服务将保持缓存的行，直到另一管道完成对自己行的处理。	存储已存储在索引缓存中的行的数据。 如果索引缓存存储主管道的键值，则数据缓存会存储主管道的数据。 如果索引缓存存储详细信息管道的键值，则数据缓存会存储详细信息管道的数据。

联接器转换端口

联接器转换具有不同的端口类型，这些端口类型可确定数据集成服务执行联接的方式。

联接器转换具有以下端口类型：

主端口

链接到映射中主源的端口。

详细端口

链接到映射中详细源的端口。

动态端口

在动态映射中接收或返回端口。 动态端口可以从上游转换接收一个或多个列并为每个列创建一个生成的端口。 动态输出端口可以返回一个或多个生成的端口。 可以定义输入规则以确定动态端口接收的列。

可以将端口从主端口更改为详细端口。 也可以将端口从详细端口更改为主端口。 更改一个端口的端口类型时，将更改所有端口的端口类型。 因此，将主端口更改为详细端口时，会将所有主端口更改为详细端口，将所有详细端口更改为主端口。

动态映射中的联接器转换

可以在动态映射中使用联接器转换。 可以在联接条件中引用动态端口和生成的端口。

动态映射是指在运行时源、目标及转换逻辑可能发生变化的映射。 您可以设置参数和规则来更改数据的结构。 在动态映射中使用联接器转换时，源的数据结构可以更改。 输入端口和联接条件中的端口也会更改。

可以执行以下任务来配置动态映射中的联接器转换：

定义动态端口。

定义动态端口和生成的端口以接受来自动态源的不同输入列。 可以在联接条件中包括动态端口或生成的端口。

定义端口选择器。

定义一个包含要在联接条件中使用的端口的端口选择器。 配置用于确定要在端口选择器中包括的端口的选择规则。 可以参数化端口选择器，以在运行时包括特定端口。

参数化联接条件。

您可以参数化整个联接条件。 为可能需要的每个联接条件配置表达式参数。

有关动态映射的详细信息，请参阅*《Informatica Developer 映射指南》*。

联接器转换中的端口选择器

联接器转换具有生成的端口时，您需要配置一个联接条件并确保当转换在运行时具有不同的生成端口时该条件有效。

例如，动态映射包括一个具有以下联接条件的联接器转换：

```
CustomerID = CustomerNo
```

CustomerID 是联接器转换中的生成的端口。 由于映射具有动态源，因此，映射可以运行多个不同的源文件格式。 包含客户编号的列在每个源文件中都有一个不同的名称：CustomerID、CustomerNum 或 CustNO。

您可以在联接器转换中创建一个端口选择器来接受动态源中的不同客户列名称。 使用一个包括前缀为“Cust”的任何端口名称的选择规则来配置端口选择器。

然后，将联接条件配置为包括端口选择器名称，而非“CustomerID”列名称：

```
Customer_PortSelector = CustomerNo
```

该联接条件对于以“Cust”开头的任何端口名称都有效。

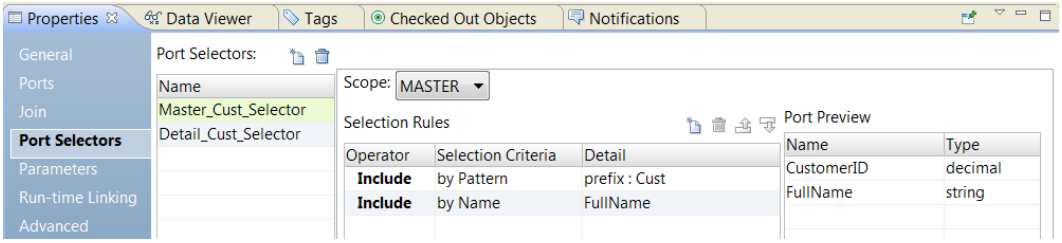
一个端口选择器中可以包含一个或多个端口。 如果该联接条件中的主组和详细组包含相同的端口数，该联接条件可以包含多个端口。

选择规则

配置端口选择器时，可以定义选择规则来确定要包括哪些生成的端口。 选择规则与您可以为动态端口配置的输入规则类似。

端口选择器可以包括端口或生成的端口。 在**端口选择器**选项卡上配置端口选择器。

下图显示了**端口选择器**选项卡：



配置端口选择器的以下属性：

名称

标识端口选择器。 可以在一个转换中创建多个端口选择器，然后在表达式中引用它们。

范围

标识要应用端口选择器的一组端口。 选择“主”或“详细信息”范围。

选择规则

确定要包括在端口选择器中的端口。 创建选择规则时，**端口预览**面板显示当前输入端口中符合条件的端口。 这些端口可能会发生变化。 配置选择规则以接受来自不同源的端口。

可以根据以下条件创建选择规则：

运算符

包括或排除选择规则返回的端口。 默认为包括。 必须先包括端口，然后才能排除端口。

选择条件

要创建的选择规则的类型。 可根据端口类型或列名称创建规则。 要根据列名称包括端口，请搜索特定名称或搜索名称中的字符模式。

详细信息

要应用到选择条件的值。 如果选择条件是按列名称，则配置要搜索的字符串或名称。 如果选择条件是按端口类型，则选择要包括的端口类型。

下表介绍了选择条件以及如何指定条件的详细信息：

选择条件	说明	详细信息
全部	包括所有端口。	无需详细信息。
名称	根据端口名称筛选端口。	从值列表中选择端口名称或使用一个类型为“端口”或“端口列表”的参数。

选择条件	说明	详细信息
类型	根据每个端口的数据类型筛选端口。	从列表中选择数据类型。
模式	按照名称中的字符串或按照正则表达式来筛选端口。	选择前缀、后缀或正则表达式作为端口名称的模式类型。然后，输入模式值或使用类型为“字符串”的参数。

创建端口选择器

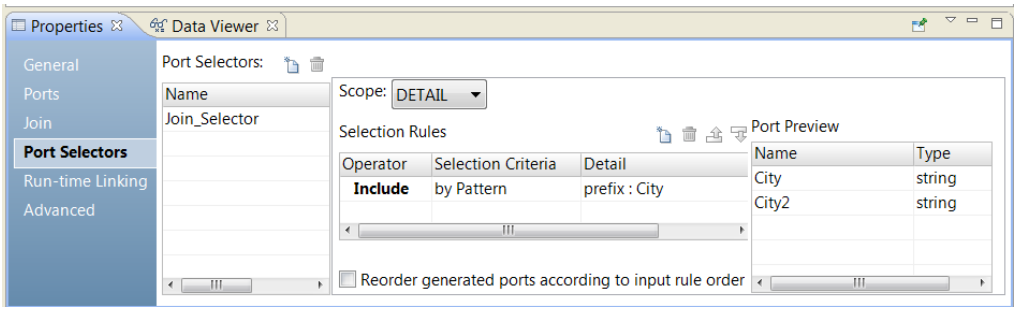
使用端口选择器可确定要在动态表达式、查找条件或联接器条件中使用的端口。

1. 单击**端口选择器**选项卡。
2. 在**端口选择器**区域中，单击**新建**。

   Developer tool 使用包含所有端口的默认选择规则来创建端口选择器。
3. 在**端口选择器**区域内，将端口选择器名称更改为唯一名称。
4. 如果您要使用联接器转换或查找转换，请选择范围。

   可用端口会根据所选的端口组发生相应变化。
5. 在**选择规则**区域内，选择**运算符**。
   - 包含。创建一个包含端口选择器端口的规则。必须先包括端口，然后才能排除端口。
   - 排除。创建一个在端口选择器中排除特定端口的规则。
6. 选择**选择条件**。
   - 按名称。按名称选择特定端口。您可以在范围内从端口列表中选择端口名称。
   - 按类型。按类型选择端口。可以选择一个或多个数据类型。
   - 按模式。按端口名称中的字符模式选择端口。可以使用特定字符进行搜索，也可以创建正则表达式。

   下图显示了“端口选择器”选项卡：



7. 单击**详细信息**列。

   输入规则详细信息对话框随即打开。
8. 选择作为端口筛选依据的值。
   - 按名称。选择按值或参数来创建端口列表。单击**选择**在列表中选择端口。
   - 按类型。从列表中选择一个或多个数据类型。**端口预览**区域显示所选类型的端口。
   - 按模式。选择在端口名称的前缀或后缀中搜索特定字符模式。或者，选择创建在搜索时使用的正则表达式。配置参数或者配置在搜索时使用的模式。

   配置规则时，**端口预览**区域会显示端口选择器中的端口。
9. 要重新排序端口选择器中的端口，请选择**根据输入规则顺序重新排序生成的端口**。

定义联接条件

联接条件包含两个输入源中的端口，数据集成服务使用这两个输入源联接两个行。

根据选定联接的类型，数据集成服务将行添加到结果集中，或者放弃该行。联接器转换将生成基于联接类型、条件和输入数据源的结果集。

定义联接条件之前，请验证是否已配置主源和详细源以实现最佳性能。映射运行期间，数据集成服务将比较主源和详细源中的每一行。要提高未排序联接器转换的性能，请将行数较少的源作为主源。要提高已排序联接器转换的性能，请将重复键值数较少的源作为主源。

在联接条件中使用来自联接器转换的输入源的一个或多个端口。使用其他端口将增加联接两个源所需的时间。条件中端口的顺序会影响联接器转换的性能。如果在联接条件中使用多个端口，则数据集成服务将按照您指定的顺序来比较端口。

如果联接 Char 和 Varchar 数据类型，则数据集成服务会将填充 Char 值的所有空格计入字符串：

```
Char(40) = "abcd"
Varchar(40) = "abcd"
```

Char 值是使用 36 个空格填充的“abcd”，数据集成服务不会联接这两个字段，因为 Char 字段的结尾包含空格。

注意: 联接器转换不会匹配空值。例如，如果 EMP_ID1 和 EMP_ID2 都包含具有空值的行，则数据集成服务不会将其视作匹配项，并且不会联接这两个行。要联接具有空值的行，请将空输入替换为默认值，然后联接默认值。

可以定义简单或高级条件类型。您还可以定义表达式参数。表达式参数是指包含联接表达式的参数。可以使用映射参数在运行时更改参数值。

简单条件类型

为已排序或未排序的联接器转换定义一个简单条件类型。

简单条件包含将指定的主源和详细信息源相比较的一个或多个条件。 简单条件必须采用以下格式：

```
<master_port> operator <detail_port>
```

对于已排序的联接器转换，此条件必须使用等式运算符。

对于未排序的联接器转换，此条件可以使用以下任一运算符： =, !=, >, >=, <, <=.

例如，如果带有名为 EMPLOYEE_AGE 和 EMPLOYEE_POSITION 的表的两个源均包含员工身份证号码，则以下条件将匹配员工在两个源中均列出的行：

```
EMP_ID1 = EMP_ID2
```

Developer tool 验证简单条件中的数据类型。 条件中的两个端口必须具有相同的数据类型。 如果需要在条件中使用数据类型不匹配的两个端口，请转换相应数据类型，以使数据类型相匹配。

可以在简单条件中配置一个联接条件列表。 配置多个联接条件时，所有条件必须为 true，才能进行联接。

例如，可以在简单条件中配置以下语句：

```
StoreID = StoreNO
Dept    = Department
Salary  > Commission
```

如果将相同的语句视为一个高级条件，联接条件将显示为下列表达式：

```
StoreID = StoreNO AND Dept = Department AND (Salary > Commission)
```

高级条件类型

为未排序联接器转换定义高级条件类型。

高级条件中可能包含任何计算结果为布尔值或数字值的表达式。高级条件可以包含以下任何运算符： =, !=, >, >=, <, <=.

可以为联接条件输入一个常量。与 FALSE 等效的数值为零 (0)。任何非零值均等效于 TRUE。例如，转换包含一个名为 NUMBER_OF_UNITS 的端口，且该端口的数据类型为数值。可以配置筛选条件，使其在 NUMBER_OF_UNITS 的值等于零时返回 FALSE。否则，该条件将返回 TRUE。

注意: 不能使用单个动态端口或端口选择器作为某个联接条件的布尔值。

要在联接条件中输入表达式，请在**联接**选项卡上选择“高级”条件类型。使用“表达式编辑器”将端口、参数、表达式、端口选择器和运算符包含在条件中。可以使用生成的端口。如果端口类型为数值型，您可以在表达式编辑器中输入一个端口。但是，您不能输入一个端口选择器作为表达式。

例如，希望通过匹配员工的全名来联接源。主源包括一个 FirstName 端口和一个 LastName 端口。详细源包括一个 FullName 端口。定义以下条件以连接主端口并匹配两个源中的全名：

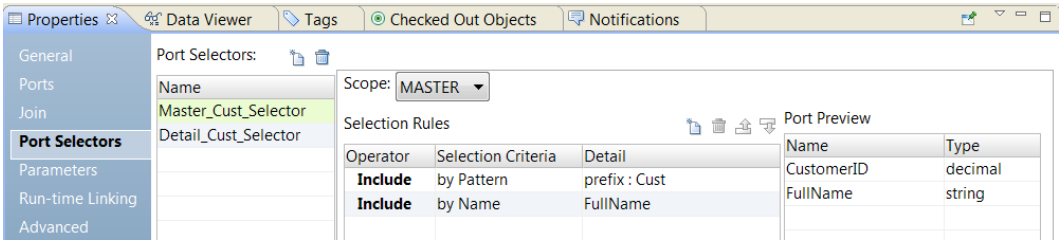
```
CONCAT(FirstName, LastName) = FullName
```

联接条件中的端口选择器

可以在联接条件中包含端口选择器。 联接条件必须引用主组中的端口选择器以及详细组中的端口选择器。

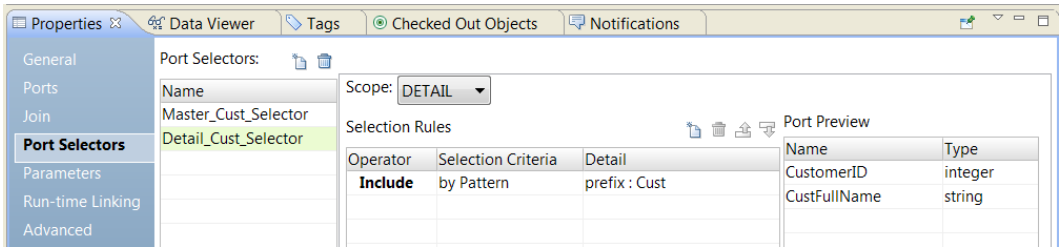
例如，联接器转换包含动态端口。 您可能需要比较联接条件中的多个生成的端口。

下图显示了主组的端口选择器中的字段：



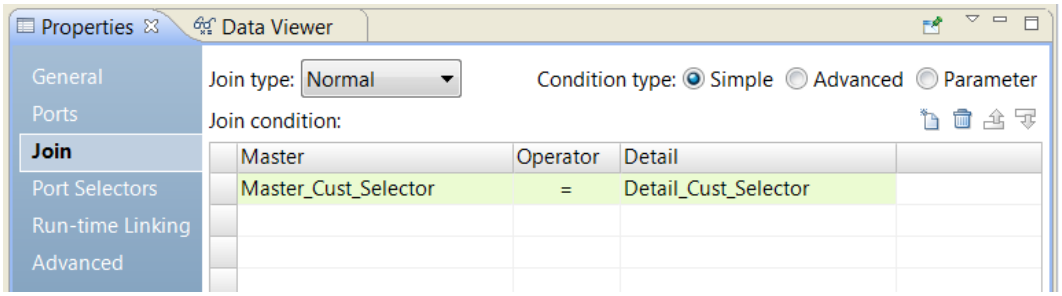
Master_Cust_Selector 包含 CustomerID 和 FullName 端口。

下图显示了详细组的端口选择器中的字段：



Detail_Cust_Selector 包含 CustomerNo 和 CustFullName 端口。 这些端口的前缀为 Cust。

创建以下简单联接条件：



联接条件将 Master_Cust_Selector 与 Detail_Cust_Selector 中的每个端口进行比较。 联接条件为：`CustomerID = CustomerNo AND FullName = CustFullName`.

每个端口选择器必须包含相同的端口数。 端口的类型必须相同。

注意: 如果更改端口选择器的范围并且简单类型的联接条件不再有效，Developer tool 可能会将条件类型切换为高级。 可以在**联接**选项卡上将联接条件类型切换回简单类型。

联接条件中的动态端口

可以在端口选择器中引用动态端口。

动态端口可以包含一个或多个生成的端口。 如果联接条件包含动态端口，则主端口数必须与详细端口数相同。

例如，动态端口 A 具有 2 个生成的端口：

```
CustomerID
OrderID
```

动态端口 B 也有 2 个生成的端口：

```
CustomerNo
OrderNo
```

以下联接条件有效：

DynamicPortA = DynamicPortB

联接条件扩展为以下表达式：

```
CustomerID = CustomerNo AND OrderID = OrderNo
```

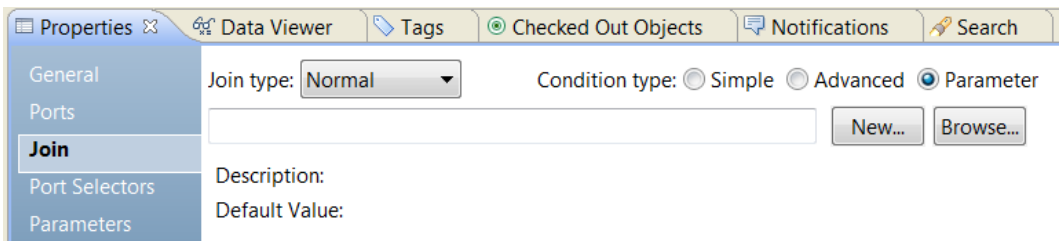
如果端口选择器包含的端口数与动态端口相同，则可以在联接条件中引用端口选择器和动态端口。

表达式参数

可以定义一个包含联接条件的表达式参数。 可以选择该参数作为联接器转换中的联接条件。

要将参数用于联接条件，请在“联接”选项卡上选择参数条件类型。

下图显示了选择参数条件类型的位置：



可以浏览现有参数，也可以创建参数。 要创建参数，请单击**新建**，然后定义参数。 在表达式编辑器中创建表达式。

注意: 表达式参数不能包含其他参数。 如果将参数嵌入到表达式参数中，数据集成服务将发出运行时验证错误。

联接类型

在联接器转换中，联接可源自不同类型的源。

联接器转换支持以下类型的联接：

- 普通
- 主外部联接
- 详细外部联接
- 完整外部联接

注意: 普通或主外部联接的执行速度比完整外部联接或详细外部联接的执行速度快。

如果结果集中包括不包含任一源中的数据的字段，则联接器转换将使用空值填充空字段。如果知道有字段返回 NULL 但不希望在目标中插入 NULL，可以为相应端口设置默认值。

普通联接

通过普通联接，数据集成服务将根据条件丢弃主源和详细源中所有不匹配的数据行。

例如，具有两个用于自动部分的数据源，称为 PARTS_SIZE 和 PARTS_COLOR。

PARTS_SIZE 数据源是主源，包含以下数据：

PART_ID1	DESCRIPTION	SIZE
1	Seat Cover	Large
2	Ash Tray	Small
3	Floor Mat	Medium

PARTS_COLOR 数据源是详细源，包含以下数据：

PART_ID2	DESCRIPTION	COLOR
1	Seat Cover	Blue
3	Floor Mat	Black
4	Fuzzy Dice	Yellow

要通过匹配两个源中的 PART_ID 联接两个表，请按如下所示设置条件：

```
PART_ID1 = PART_ID2
```

通过普通联接来联接这些表时，结果集中将包括以下数据：

PART_ID	DESCRIPTION	SIZE	COLOR
1	Seat Cover	Large	Blue
3	Floor Mat	Medium	Black

以下示例显示了等价 SQL 语句：

```
SELECT * FROM PARTS_SIZE, PARTS_COLOR WHERE PARTS_SIZE.PART_ID1 = PARTS_COLOR.PART_ID2
```

主外部联接

主外部联接将保留详细源的所有数据行和主源的匹配行。 它将丢弃主源中不匹配的行。

通过主外部联接和相同条件来联接示例表时，结果集将包括以下数据：

PART_ID	DESCRIPTION	SIZE	COLOR
1	Seat Cover	Large	Blue
3	Floor Mat	Medium	Black
4	Fuzzy Dice	NULL	Yellow

因为没有为 Fuzzy Dice 指定大小，所以数据集成服务将使用 NULL 填充该字段。

以下示例显示了等价 SQL 语句：

```
SELECT * FROM PARTS_SIZE RIGHT OUTER JOIN PARTS_COLOR ON (PARTS_COLOR.PART_ID2 = PARTS_SIZE.PART_ID1)
```

详细外部联接

详细外部联接将保留主源的所有数据行和详细源的匹配行。 它将丢弃详细源中不匹配的行。

通过详细外部联接和相同条件来联接示例表时，结果集将包括以下数据：

PART_ID	DESCRIPTION	SIZE	COLOR
1	Seat Cover	Large	Blue
2	Ash Tray	Small	NULL
3	Floor Mat	Medium	Black

因为没有为 Ash Tray 指定颜色，所以数据集成服务将使用 NULL 填充该字段。

以下示例显示了等价 SQL 语句：

```
SELECT * FROM PARTS_SIZE LEFT OUTER JOIN PARTS_COLOR ON (PARTS_SIZE.PART_ID1 = PARTS_COLOR.PART_ID2)
```

完整外部联接

完整外部联接将保留主源和详细源中的所有数据行。

通过完整外部联接和相同条件来联接示例表时，结果集将包括以下数据：

PARTED	DESCRIPTION	SIZE	Color
1	Seat Cover	Large	Blue
2	Ash Tray	Small	NULL
3	Floor Mat	Medium	Black
4	Fuzzy Dice	NULL	Yellow

因为没有为 Ash Tray 指定颜色并且没有为 Fuzzy Dice 指定大小，所以数据集成服务将使用 NULL 填充这些字段。

以下示例显示了等价 SQL 语句：

```
SELECT * FROM PARTS_SIZE FULL OUTER JOIN PARTS_COLOR ON (PARTS_SIZE.PART_ID1 = PARTS_COLOR.PART_ID2)
```

联接器转换的已排序输入

您可以通过已排序输入选项提高联接器转换的性能。 如果数据已排序，则可以使用已排序输入。

将联接器转换配置为使用已排序数据时，数据集成服务会通过将磁盘输入和输出降至最低来提高性能。 如果使用的是大型数据集，将看到最显著的性能提高。

要配置映射以使用已排序数据，需要在映射中建立并维护排序顺序，以便数据集成服务在处理联接器转换时可以使用已排序数据。 完成以下步骤以配置映射：

1. 配置要联接的数据的排序顺序。
2. 添加维护已排序数据顺序的转换。
3. 配置联接器转换以使用已排序数据并配置联接条件，从而来使用排序来源端口。 排序来源表示已排序数据的源。

配置排序顺序

配置排序顺序以确保数据集成服务将已排序数据传递给联接器转换。

要配置排序顺序，请使用以下方法之一：

- 使用排序的平面文件。 平面文件包含排序数据时，请验证排序列的顺序在每个源文件中是否匹配。
- 使用排序的关系数据。 在关系数据对象中使用排序端口可以对源数据库中的列进行排序。 在每个关系数据对象中配置的排序端口的顺序应相同。
- 使用排序器转换对关系数据或平面文件数据进行排序。 将排序器转换置于主管道和详细信息管道中。 将每个排序器转换配置为使用相同顺序的排序键端口和排序顺序方向。

如果向配置为使用已排序数据的联接器转换传递未排序或排序不正确的数据，则映射运行将失败。 数据集成服务会将该错误记录到日志文件中。

将转换添加到映射

向映射中添加转换，该映射可维护联接器转换中已排序数据的顺序。

可以在排序源后直接放置联接器转换以维护已排序数据。

在排序源与联接器转换之间添加转换时，请遵循以下准则维护已排序数据：

- 请勿在排序源与联接器转换之间放置以下任何转换：
  - 等级
  - 联合
  - 未排序汇总器
  - 包含上述转换中任一转换的 Mapplet
- 如果遵循以下准则，则可以在排序源与联接器转换之间放置已排序汇总器转换：
  - 为已排序输入配置汇总器转换。
  - 在汇总器转换中使用的分组依据列的端口与排序源上的端口相同。
  - 分组依据端口的顺序必须与排序源上的端口顺序相同。
- 将联接器转换的结果集与其他管道联接在一起时，请验证第一个联接器转换中的数据输出是否已排序。

联接条件的规则和准则

为已排序联接器转换创建联接条件时，适用特定规则和准则。

请在创建联接条件时遵循以下准则：

- 必须定义使用等式运算符的简单条件类型。
- 如果在排序来源和联接器转换之间使用了已排序的汇总器转换，请在定义联接条件时将已排序汇总器转换视作排序来源。
- 在联接条件中使用的端口必须与排序来源中的端口匹配。
- 配置多个联接条件时，第一个联接条件中的端口必须与排序来源中的前几个端口匹配。
- 配置多个条件时，条件的顺序必须与排序来源中的端口顺序匹配，并且不得跳过任何端口。
- 排序来源中已排序端口的编号可以大于或等于联接条件中端口的编号。
- 如果联接数据类型为 Decimal 的端口，则每个端口的精度必须属于相同的精度范围。

  可以使用以下有效精度范围之一：
  - 小数 0-18
  - 小数 19-28
  - 十进制，29-38 位数
  - 十进制，39 位数及以上

  例如，如果定义了条件 `DecimalA = DecimalB`（其中，DecimalA 精度为 15，DecimalB 精度为 25），则该条件无效。

联接条件和排序顺序的示例

本示例显示了一个联接器转换，该转换可将主管道和详细信息管道与已排序端口相联接。

可以使用以下已排序端口，在主管道和详细信息管道中配置排序器转换：

- ITEM_NO

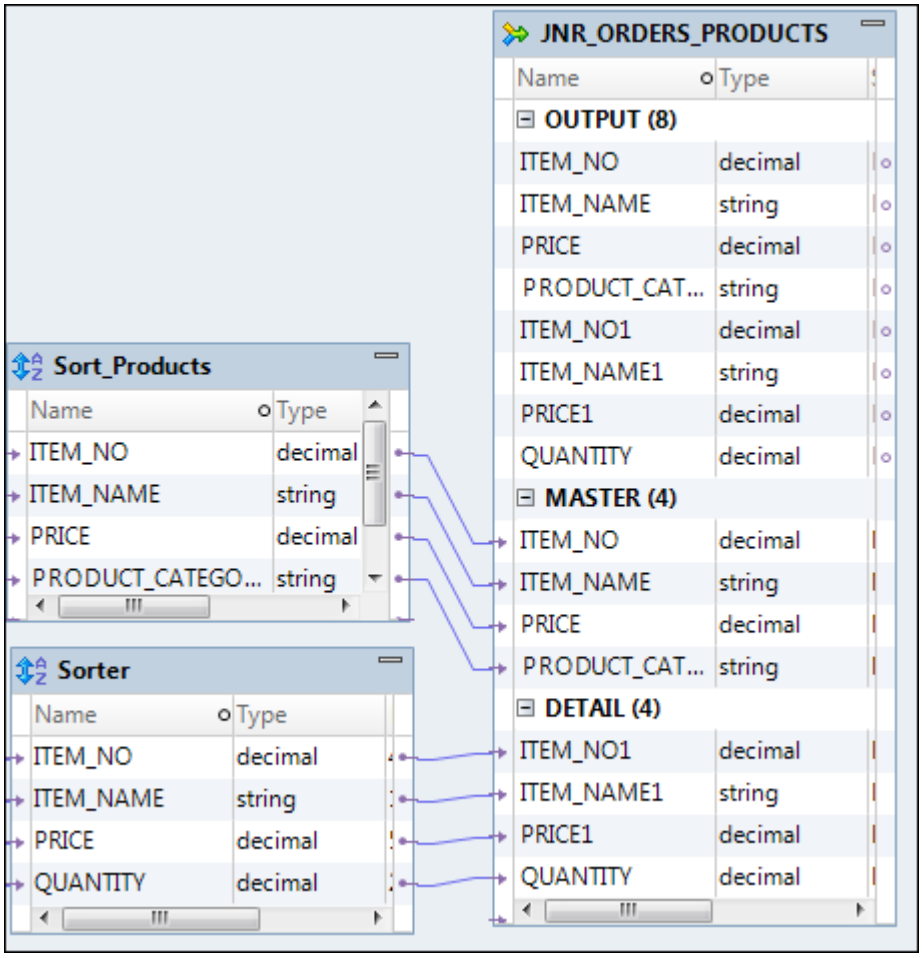
- ITEM_NAME
- PRICE

配置联接条件时，请遵循以下准则维护排序顺序：

- 必须在第一个联接条件中使用 ITEM_NO。
- 如果要添加第二个联接条件，则必须使用 ITEM_NAME。
- 如果要在联接条件中使用 PRICE，则必须同时在第二个联接条件中使用 ITEM_NAME。

如果跳过 ITEM_NAME 并联接 ITEM_NO 和 PRICE，则会失去排序顺序，并且数据集成服务将无法运行映射。

下图显示了一个映射，该映射已配置为对端口 ITEM_NO、ITEM_NAME 和 PRICE 进行排序和联接：



使用联接器转换联接主管道和详细信息管道时，可以配置以下任一联接条件：

```
ITEM_NO = ITEM_NO
```

或者

```
ITEM_NO = ITEM_NO1
ITEM_NAME = ITEM_NAME1
```

或者

```
ITEM_NO = ITEM_NO1
ITEM_NAME = ITEM_NAME1
```

```
PRICE = PRICE1
```

从同一源联接数据

如果希望对部分数据执行计算并联接已转换数据与原始数据，可以从同一源联接数据。

从同一源联接数据时，可以维护原始数据并在某个映射内转换部分数据。 可以通过以下方式从同一源联接数据：

- 联接同一管道的两个分支。
- 联接同一源的两个实例。

联接同一管道的两个分支

从同一源联接数据时，可以创建该管道的两个分支。

使管道产生分支时，必须至少将映射输入和联接器转换之间的转换添加到管道的一个分支中。 必须联接已排序数据并为已排序输入配置联接器转换。

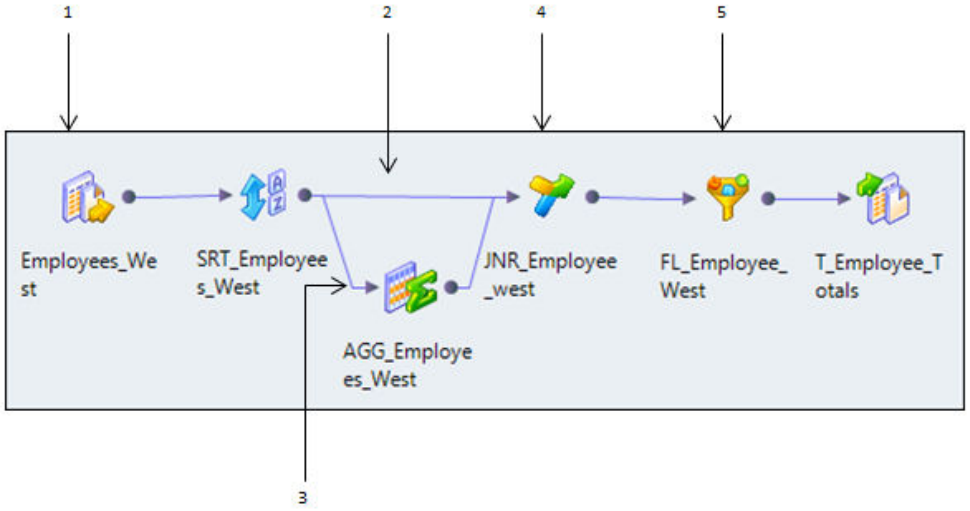
例如，具有使用以下端口的源：

- Employee
- 部门
- 销售总额

在目标中，希望查看生成的销售额高于部门平均销售额的员工。 要执行该操作，请使用以下转换创建映射：

- 排序器转换。 对数据进行排序。
- 已排序汇总器转换。 计算销售数据的平均值并按部门进行分组。 执行此汇总时，将丢失单个员工的数据。 要维护员工数据，必须将管道的分支传递给汇总器转换，并将具有相同数据的分支传递给联接器转换以维护原始数据。 联接管道的两个分支时，将联接汇总数据与原始数据。
- 已排序联接器转换。 联接已排序汇总数据与原始数据。

- 筛选器转换。 比较平均销售数据和每个员工的销售数据，并筛选销售额低于上述平均值的员工。



1. Employees_West 源
2. 管道分支 1
3. 管道分支 2
4. 已排序联接器转换
5. 筛选销售额低于上述平均值的员工

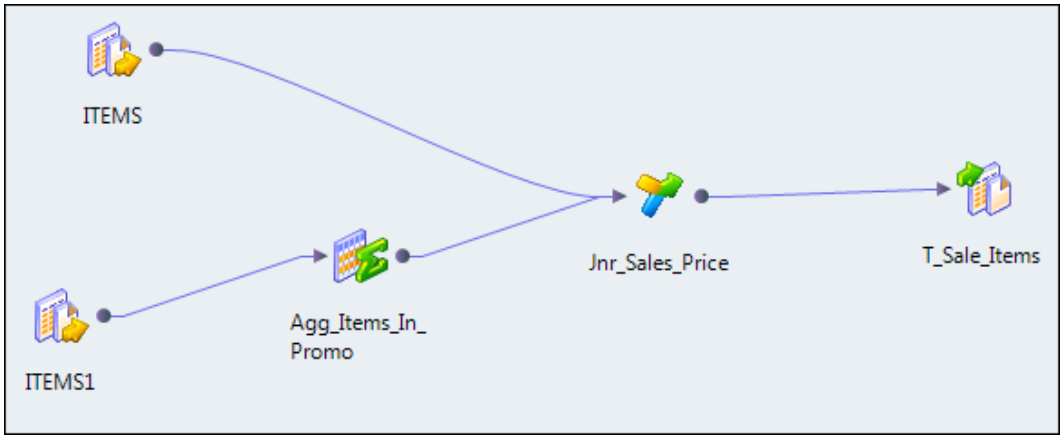
如果联接器转换从一个分支接收到数据的时间远远晚于从另一个分支接收到数据的时间，则联接这两个分支可能会降低性能。 联接器转换将缓存第一个分支中的所有数据，并在缓存已满时将其写入磁盘。 然后，联接器转换接收第二个分支中的数据时，必须从磁盘中读取数据。

联接同一源的两个实例

可以通过创建该源的其他实例从同一源联接数据。

创建其他源实例后，可以联接两个源实例中的管道。 如果希望联接未排序数据，则必须创建同一源的两个实例并联接管道。

下图显示了同一源中通过联接器转换联接的两个实例：



联接同一源的两个实例时，数据集成服务将读取每个源实例的源数据。 与联接管道的两个分支相比，性能可能更低。

从同一源联接数据的准则

决定是联接管道的分支还是联接源的两个实例时，请遵循特定准则。

决定是联接管道的分支还是联接源的两个实例时，请遵循以下准则：

- 源较大或者如果只能读取源数据一次时，请联接管道的两个分支。
- 使用已排序数据时，请联接管道的两个分支。 如果源数据未排序，而您使用排序器转换对该数据进行排序，请在数据排序完成后对管道进行分支。
- 需要将编块转换添加到源和联接器转换之间的管道中时，请联接源的两个实例。
- 如果某个管道的处理速度慢于其他管道，请联接源的两个实例。
- 如果需要联接未排序数据，请联接源的两个实例。

阻止源管道

使用联接器转换运行映射时，根据映射配置以及您是否为已排序输入配置联接器转换，数据集成服务将阻止和接受源数据。

未排序联接器转换

数据集成服务处理未排序的联接器转换时，它将先读取所有主行，之后再读取详细信息行。 数据集成服务将在缓存主源中的行时阻止详细信息源。

数据集成服务读取并缓存所有主行后，将取消阻止详细信息源并读取详细信息行。 某些包含未排序联接器转换的映射将违反数据流验证。

已排序联接器转换

数据集成服务处理已排序的联接器转换时，它将基于映射配置阻止数据。 如果联接器转换的主输入和详细信息输入源自不同的源，阻止逻辑是有可能的。

如果可以在不同时阻止目标加载顺序组中所有源的情况下执行操作，数据集成服务将使用阻止逻辑来处理联接器转换。 否则，它将不使用阻止逻辑。 相反，它将在缓存中存储更多行。

数据集成服务可以使用阻止逻辑来处理联接器转换时，它将在缓存中存储较少行，以提高性能。

缓存主行

数据集成服务处理联接器转换时，会同时从两个源读取行，并基于主行建立索引和数据缓存。

然后数据集成服务会基于详细源数据和缓存数据执行联接。 数据集成服务在缓存中存储的行数取决于源数据以及是否为已排序输入配置联接器转换。

要提高未排序联接器转换的性能，请使用行数较少的源作为主源。 要提高已排序联接器转换的性能，请使用重复键值数较少的源作为主源。

联接器转换性能提示

请参考提示来提高联接器转换的性能。

联接器转换会降低性能，因为这些转换在运行时需要额外的空间来保存中间结果。可以查看联接器性能计数器信息以确定是否需要优化联接器转换。

请参考以下提示来提高联接器转换的性能：

指定主源作为具有较少重复键值的源。

处理已排序联接器转换时，数据集成服务一次缓存一百个唯一键对应的行。如果主源包含许多具有相同键值的行，则数据集成服务必须缓存多个行，该操作可能会降低性能。

指定主源作为具有较少行的源。

联接器转换将比较详细源和主源的每一行。主源中的行数越少，联接比较发生的迭代数越少，从而加快联接进程的速度。

尽可能在数据库中执行联接。

在数据库中执行联接的速度快于在映射运行期间执行联接的速度。使用的数据库联接类型可能会影响性能。普通联接的速度比外部联接的速度快，且产生的行数较少。有时，无法在数据库中执行联接，例如联接两个不同数据库或平面文件系统中的表。

尽可能联接已排序数据。

配置联接器转换以使用已排序输入。通过最大程度地减少磁盘输入和磁盘输出，数据集成服务可提高性能。如果使用的是大型数据集，将呈现出最显著的性能提高。对于未排序联接器转换，请将具有较少行的源指定为主源。

优化联接条件。

数据集成服务尝试通过读取较小组中的行、查找较大组中的匹配行并执行联接操作来减小一个联接操作的数据集的大小。减小数据集的大小可提高映射性能，因为数据集成服务不再从较大的组源中读取不必要的行。数据集成服务将联接条件移至较大的组源，并仅读取与较小的组匹配的行。

使用半联接优化方法。

在以下情况下，请使用半联接优化方法以提高映射性能：一个输入组具有的行数远大于其他输入组，以及根据联接条件，较大的组具有许多与较小的组不匹配的行。

联接器转换的规则和准则

使用联接器转换时，需要应用一定的规则和准则。

联接器转换接受来自大多数转换的输入。但是，如果任一输入管道包含更新策略转换，则不能使用联接器转换。

非本地环境中的联接器转换

非本地环境中的联接器转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。

- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的联接器转换

在下列情况下，映射验证会失败：

- 转换包含不相等联接且映射端联接处于禁用状态。
- 联接器转换表达式引用未连接的查找转换。

当为详细信息外部联接或完整外部联接联接配置了联接器转换时，映射端联接处于禁用状态。

Spark 引擎上的联接器转换

在下列情况下，映射验证会失败：

- 联接条件为二进制数据类型或包含二进制表达式。

source: Transformations in the Non-native Environment (Conrefs) conref: Databricks and Spark. This conref is reused in the Transformations topic in the Developer Transformation Guide and BDM User Guide.

流映射中的联接器转换

流映射具有不适用于批处理映射的其他处理规则。

映射验证

在下列情况下，映射验证会失败：

- 联接器转换在汇总器转换的下游。
- 联接器转换在等级转换的下游。
- 流管道中包含多个联接器转换。
- 联接器转换联接流管道和非流管道的数据。

一般准则

要指定联接条件，可从“联接”选项卡上的高级条件类型中选择 TIME_RANGE 函数，然后输入联接条件表达式。TIME_RANGE 函数确定要联接的流事件的时间范围。

Databricks Spark 引擎上的联接器转换

在下列情况下，映射验证会失败：

- 联接条件为二进制数据类型或包含二进制表达式。

source: Transformations in the Non-native Environment (Conrefs) conref: Databricks and Spark. This conref is reused in the Transformations topic in the Developer Transformation Guide and BDM User Guide.

第 23 章

键生成器转换

本章包括以下主题：

键生成器转换概览

键生成器转换是一种主动转换，用于根据选定列中的数据值将记录组织成组。 使用该转换可以在将记录传递给匹配转换之前对记录进行分类。

键生成器转换使用分组策略为选定列创建组键。 这些策略是字符串、Soundex 和 NYSIIS。 选定字段中具有通用值的记录具有一个通用的组键值。 匹配转换会将具有通用组键值的记录一起处理。 这样就可以加快匹配转换中的重复项分析。

匹配转换需要执行的比较运算次数随着数据集中的记录数呈指数增长。 这种指数增长可能会占用大量的计算资源。 通过创建组键，键生成器转换可以使匹配转换以较小的组为单位来比较记录，从而缩短了处理时间。

执行字段匹配时，请选择一个可能为匹配提供有用组的列，以便生成组键。 例如，姓氏列可能会提供比名字列更有意义的组键数据。 但是，如果要选择姓氏列在匹配转换中进行重复项分析，则不要使用此列。

键生成器转换还可以为每个记录创建唯一 ID。 进入匹配转换的每个记录都必须包含唯一 ID。 如果数据不存在 ID，则可以使用键生成器转换为数据创建 ID。

Soundex 策略

Soundex 策略使用表示字发音的代码分析字并创建组键。

Soundex 代码以字的第一个字母开头，后跟表示连续辅音的一系列数字。 使用 Soundex 策略将相同代码分配给发音类似的字。 配置 Soundex 深度以定义策略返回的字母数字字符数量。

此策略侧重于字的发音而不是拼写，它可以对其他拼写和细微拼写差异进行分组。 例如，Smyth 和 Smith 的 Soundex 代码相同。

Soundex 策略也可以对发音错误的字进行分组。 例如，名称 Edmonton 和 Edmonson 的 Soundex 代码相同。

Soundex 策略属性

配置 Soundex 策略属性来确定键生成器转换用于创建组键的 Soundex 设置。

下表介绍了 Soundex 策略属性：

属性	说明
Soundex 深度	确定 Soundex 策略返回的字母数字字符数。 默认深度为 3。 此深度将创建 Soundex 代码，包含字符串中第一个字母和代表接下来两个相异辅音发音的两个数字。

相关主题：

- “字符串策略属性” 页面上 329
- “键创建属性” 页面上 331
- “ 配置分组策略” 页面上 330

字符串策略

字符串策略根据输入数据中的子字符串创建组键。

可以指定输入列中子字符串的长度和位置。 例如，可以配置此策略以根据输入字符串中的前四个字符创建键。

字符串策略属性

配置字符串策略属性来确定键生成器转换用于创建组键的子字符串。

下表介绍了字符串策略属性：

属性	说明
从左开始	配置转换，以便从左至右读取输入字段。
从右开始	配置转换，以便从右至左读取输入字段。
起始位置	指定要跳过的字符数。 例如，如果为**起始位置**输入 3，子字符串将从输入字段的第四个字符（从您指定的一侧开始算起）开始。
长度	指定用作组键的字符串的长度。 输入 0，则可使用整个输入字段。

相关主题：

- “Soundex 策略属性” 页面上 329
- “键创建属性” 页面上 331
- “ 配置分组策略” 页面上 330

NYSIIS 策略

NYSIIS 策略根据表示字发音的字母分析字并创建组键。

Soundex 策略仅会考虑字符串中的第一个元音，而 NYSIIS 策略则会分析整个字符串中的元音。 NYSIIS 策略将所有字母转换为六个字符之一，并将大多数元音转换为字母 A。

键生成器输出端口

键生成器转换输出端口将创建 ID 和组键，匹配转换会使用这些 ID 和组键来处理记录。

下表介绍了键生成器转换的输出端口：

属性	说明
SequenceID	创建一个 ID 以标识源数据集中的每个记录。
GroupKey	创建匹配转换用于处理记录的组键。

创建可重用的键生成器转换时，可使用**概览**视图查看端口。 将不可重用的转换添加到映射中时，可使用**属性**视图的**端口**选项卡查看端口。

配置分组策略

要配置分组策略，请编辑**策略**视图中的属性。

配置键生成器策略之前，请先向键生成器转换添加输入端口。

1. 选择**策略**视图。
2. 单击**新建**按钮。
3. 选择分组策略。
4. 单击**确定**。
5. 在**输入**列中，选择一个输入端口。
6. 通过单击属性字段中的选择箭头，配置策略属性。
7. 配置键创建属性。

相关主题：

- “Soundex 策略属性” 页面上 329
- “字符串策略属性” 页面上 329
- “键创建属性” 页面上 331

键创建属性

配置适用于所分析数据的键创建属性。

下表介绍了键创建属性：

属性	说明
排序结果	使用 GroupKey 字段对键生成器转换输出进行排序。 对于字段匹配操作，必须选择此选项或者验证是否为匹配转换提供了排序数据。 对于标识匹配操作，请不要选择此选项。
自动生成序列键	按照输入数据的顺序生成序列键字段。
使用字段作为序列键	生成指定列的序列字段。
序列键字段	指定序列键字段的名称。

相关主题：

- “Soundex 策略属性” 页面上 329
- “字符串策略属性” 页面上 329
- “ 配置分组策略” 页面上 330

键生成器转换高级属性

键生成器转换包含确定高速缓存行为和跟踪级别的高级属性。

您可以配置以下高级属性：

缓存文件目录

指定数据集成服务将当前转换的临时数据写入到的目录。当输入数据量超出可用系统内存时，数据集成服务会将临时文件写入该目录。数据集成服务会在运行映射后删除这些临时文件。

您可以在该属性上输入目录路径，也可以使用参数标识目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。默认值为 CacheDir 系统参数。

缓存文件大小

确定数据集成服务对转换的输入数据排序使用的系统内存量。可以使用参数指定缓存文件大小。

在对数据排序之前，数据集成服务会分配您指定的内存量。如果排序操作生成的数据较多，数据集成服务会将超出的数据写入缓存目录。如果排序操作所需的内存超出系统内存和文件存储可以提供的内存，映射会失败。

如果未指定缓存文件大小，转换会对数据集成服务的执行选项应用最大内存值。

转换以字节为单位读取值。默认值为 400,000 字节。最大值为 2,147,483,647 字节。可以使用参数指定缓存文件大小。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的键生成器转换

非本地环境中的键生成器转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的键生成器转换

支持键生成器转换，但存在以下限制：

- 当您在多个节点上运行具有“键生成器转换”的映射时，转换中的序列生成选项可能不会产生唯一的 ID 值。

Spark 引擎上的键生成器转换

支持键生成器转换，但存在以下限制：

- 当您在多个节点上运行具有“键生成器转换”的映射时，转换中的序列生成选项可能不会产生唯一的 ID 值。

Databricks Spark 引擎上的键生成器转换

支持键生成器转换，但存在以下限制：

- 当您在多个节点上运行具有“键生成器转换”的映射时，转换中的序列生成选项可能不会产生唯一的 ID 值。

第 24 章

标签创建器转换

本章包括以下主题：

标签创建器转换概览

标签创建器转换是一种被动转换，用于分析输入端口字段并写入描述每个字段数据的文本标签。

如果要了解端口所包含的信息类型，可以使用标签创建器转换。 如果不了解端口上的信息类型，或者要标识端口上不包含所需信息类型的记录，可以使用标签创建器转换。

标签是由一个或多个字符组成的字符串，用于描述输入字符串。 通过配置标签创建器转换，可以基于每个字符串所包含的数据为输入字符串分配标签。

配置转换时，请指定要搜索的字符类型或字符串类型，以及在转换找到关联的字符或字符串后作为输出写入的标签。 配置添加标签操作时，请输入要搜索的字符类型和字符串类型，以及要使用的标签。 或者，使用引用数据对象来指定字符、字符串和标签。

配置转换以便为字符或标志添加标签：

为字符添加标签

写入用于描述输入字符串字符结构的标签，包括标点和空格。 转换将在列的每一行中写入一个标签。 例如，标签创建器转换可以为邮政编码 10028 添加标签“nnnnn”，其中，“n”表示一个数字字符。

为标志添加标签

写入用于描述输入字符串信息类型的标签。 转换将为输入数据中所标识的每个标志写入一个标签。 例如，可以配置标签创建器转换以使用标志“Word Init Word”为字符串“John J. Smith”添加标签。

标志是输入字符串中的一个分隔符值。

标签创建器找到与指定的标签匹配的字符或字符串后，会将标签名称写入新的输出端口。

标签创建器转换使用引用数据来标识字符和标志。 在标签创建器策略中配置操作时，需要选择引用数据对象。

何时使用标签创建器转换

标签创建器转换会为端口上的每个值写入描述性标签。

以下示例介绍了一些可以使用标签创建器转换执行的分析类型。

使用联系人数据查找记录

为此转换配置引用表，该引用表包含名字列表。 创建“为标志添加标签”策略，以便为任何与引用表值匹配的字符串添加标签。 查看输出数据时，任何包含标签的记录都可能会标识人员。

查找公司记录

使用包含公司后缀（例如 Inc、Corp 和 Ltd）列表的标志集配置该转换。创建“为标志添加标签”策略，以便为任何与引用表值匹配的字符串添加标签。 查看输出数据时，任何包含标签的记录都可能会标识公司。

注意: 使用要标识任何公司名称的公司后缀标志集。 如果确定引用表包含所有要标识的公司，则可以使用公司名称的引用表。 例如，可以使用列出了纽约证券交易所上市公司的引用表。

查找电话号码数据

为此转换配置字符集，该字符集定义了电话号码的字符结构。 例如，可以使用将不同模式的标点符号和数字识别为美国电话号码的字符集。 可以通过查看数据来查找包含错误的电话号码数字的记录。

字符标签可能会使用以下字符来分析列数据：

```
c=punctuation character n=digit s=space
```

下表显示了电话号码结构示例：

字符结构	电话号码
cnnncsnnncnnncnnnnn	(212) 555-1212
nnnnnnnnnn	2125551212
cnnncnnncnnnn	+212-555-1212

标签创建器转换中的引用数据使用

Informatica Developer 安装有不同类型的可与标签创建器转换一起使用的引用数据对象。 您还可以创建引用数据对象。

如果要将引用数据对象添加到标签创建器转换策略，则该转换将在策略的输入数据中搜索引用数据对象中的值。该转换会将任何找到的值替换为引用数据对象中的有效值，或替换为您指定的值。

下表介绍了可以使用的引用数据类型：

引用数据类型	说明
字符集	标识不同类型的字符，例如字母、数字和标点符号。 在“为字符添加标签”操作中使用。
概率模型	将模糊匹配功能添加到“为标志添加标签”操作。 该转换可以使用概率模型来推理字符串中的信息类型。 要启用模糊匹配功能，请在 Developer 工具中编译概率模型。 在“为标志添加标签”操作中使用。
引用表	查找与数据库表中的条目相匹配的字符串。 在“为标志添加标签”和“为字符添加标签”操作中使用。
正则表达式	标识与您所定义的条件相匹配的字符串。 可以使用正则表达式在一个较大的字符串中查找某一字符串。 在“为标志添加标签”操作中使用。
标志集	基于字符串所包含的信息类型来标识这些字符串。 在“为标志添加标签”操作中使用。 Informatica 将不同类型的标志定义（例如文字、电话号码、邮政编码以及产品代码定义）与标志集安装在一起。

字符集

字符集包含可标识特定字符和字符范围的表达式。 可以在使用标志解析模式的标签创建器转换和解析器转换中使用字符集。

字符范围指定字符代码的顺序范围。 例如，字符范围“[A-C]”与大写字符“A”、“B”和“C”相匹配。该字符范围与小写字符“a”、“b”或“c”不匹配。

使用字符集将特定字符或字符范围标识为标志解析或“为标志添加标签”操作的一部分。 例如，可以在包含电话号码的列中标注所有数字。 为所有数字添加标签后，可以使用解析器转换标识模式，并将问题模式写入单独的输出。

概率模型

概率模型可依据标志所包含的信息类型以及标志在输入字符串中的位置对标志进行标识。

概率模型包含引用数据值和标签值。引用数据值代表连接到转换的输入端口上的数据。标签值描述引用数据值包含的信息类型。请为模型中的每个引用数据值分配标签。

要将引用数据值链接到概率模型中的标签，您可以对该模型进行编译。编译过程会在数据值与标签之间生成一系列逻辑关联。当您运行读取模型的映射时，数据集成服务会将模型逻辑应用到转换输入数据。数据集成服务将返回最能准确描述输入数据值的标签。

您可以在 Developer tool 中创建概率模型。模型存储库存储概率模型对象。Developer tool 将数据值、标签和编译数据写入 Informatica 目录结构中的一个文件。

引用表

引用表是一个至少包含两列的数据库表。 一列包含标准版本或所需版本的数据值，另一列包含备选版本的数据值。 将引用表添加到转换时，该转换将在输入端口数据中搜索同时显示在引用表中的值。 可以使用任何对所处理的数据项目有用的数据来创建引用表。

正则表达式

在添加标签操作上下文中，正则表达式是可用于标识输入数据中特定字符串的表达式。 可以在使用“为标志添加标签”模式的标签创建器转换中使用正则表达式。

标签创建器转换使用正则表达式匹配输入模式并创建单个标签。 包含多个输出的正则表达式不会生成多个标签。

标志集

标志集包含能够标识特定标志的表达式。 您可以在使用“为标志添加标签”模式的标签创建器转换中使用标志集。

使用标志集可以将特定标志标识为“为标志添加标签”操作一部分。 例如，您可以使用标志集为所有采用“AccountName@DomainName”格式的电子邮件地址添加标签。 为标志添加标签后，您可以使用解析器转换将这些电子邮件地址写入指定的输出端口。

Developer 工具包括系统定义的标志集，您可以这些标志集标识各种模式。 系统定义的标志集示例包括：

- 文字
- 数字
- 电话号码
- 电子邮件地址
- 邮政编码
- 公民身份证号，例如社会保障号
- 信用卡号

标签创建器转换策略

使用添加标签策略向输入数据分配标签。 要配置添加标签策略，请在标签创建器转换的**策略**视图中编辑相应设置。

创建添加标签策略时，需要添加一个或多个操作。 每个操作均实现一个特定的添加标签任务。

标签创建器转换将提供一个用于创建策略的向导。 创建添加标签策略时，需要在“为字符添加标签”模式或“为标志添加标签”模式之间进行选择。 然后添加特定于该添加标签模式的操作。

重要说明: 可以更改操作和策略的顺序。 策略中的操作顺序会更改策略的输出，因为每个操作读取的是前一操作的结果。

“为字符添加标签”操作

使用“为字符添加标签”操作可以创建描述数据中字符模式的标签。

可以向“为字符添加标签”策略添加以下类型的操作：

使用字符集为字符添加标签

使用预定义的字符集（例如数字字符或字母字符）为字符添加标签。 可以选择 Unicode 字符集和非 Unicode 字符集。

使用引用表为字符添加标签

使用引用表中的自定义标签为字符添加标签。

“为标志添加标签”操作

执行“为标志添加标签”操作可创建用于说明数据中的字符串的标签。

标签创建器转换可以标识一个输入字符串中的多个标志以及为这些标志添加标签。 例如，可以将标签创建器转换配置为使用美国电话号码和电子邮件地址标志集。 当标签创建器转换处理输入字符串“555-555-1212 someone@somewhere.com”时，输出字符串为“USPHONE EMAIL”。

您可以将以下类型的“为标志添加标签”操作添加到标签策略：

使用引用表添加标签

为匹配引用表条目的字符串添加标签。

使用标志集为标志添加标签

为匹配标志集数据或概率模型数据的字符串模式添加标签。

标签创建器转换端口

为在转换中配置的添加标签操作选择所需的输入端口和输出端口。

标签创建器转换将使用以下端口：

输入端口

从上游对象读取字符串输入。

已添加标签的输出端口

写入转换操作所定义的标签。

标志化的输出端口

传递与输出中每个标签对应的输入字符串。 如果要在 Mapplet 或映射中为标签创建器转换添加下游解析器转换，并将该解析器转换配置为在基于模式的解析模式中运行，请选择此端口。 解析器转换将“为标志添加标签”输出与标志化输出端口上的数据相关联。

得分输出端口

选择此端口可在“为标志添加标签”操作中写入通过概率匹配技术而生成的得分值。

运行使用概率模型的“为标志添加标签”操作时，该操作将为每个添加标签的字符串生成一个数值得分。 该得分表示输入字符串和概率模型中所定义的模式之间的相似度。

“为字符添加标签”属性

在标签创建器转换的**策略**视图中配置“为字符添加标签”操作的属性。

常规属性

常规属性将应用于策略中定义的所有“为字符添加标签”操作。 使用常规属性命名策略，并指定输入端口和输出端口。

下表介绍了常规属性：

属性	说明
名称	提供策略名称。
输入	标识策略操作可以读取的输入端口。
输出	标识策略操作可以写入的输出端口。
说明	提供策略的文本说明。 这是可选属性。

引用表属性

定义“为字符添加标签”策略时，可以使用字符集和引用表将操作添加到标签。 使用引用表属性可以指定转换使用引用表的方式。

下表介绍了引用表属性：

属性	说明
名称	请提供操作名称。
引用表	指定转换用来向字符添加标签的引用表。
标签	指定与引用表条目匹配的输入字符的替换文本。
替代策略中的其他标签	确定此添加标签操作是否替代其他添加标签操作。

字符集属性

定义“为字符添加标签”策略时，可以使用字符集和引用表将操作添加到标签。使用字符集属性可以指定转换使用字符集的方式。

下表介绍了字符集属性：

属性	说明
名称	请提供操作名称。
选择字符集	指定转换用来向字符串添加标签的字符集。 可以替代与字符集匹配的输入字符串的替换文本。单击**标签**列中的选择箭头以输入自定义替换文本。
筛选文本	使用输入的字符或通配符来筛选字符集列表。
添加字符集	选择此项可定义自定义字符集。

属性	说明
编辑	编辑自定义字符集的内容。
导入	允许为存储在内容集中的字符集创建不可重用副本。对原始字符集的更改不会更新到存储在标签创建器转换中的副本中。
删除	删除自定义字符集。
指定执行顺序	设置操作将标志集应用于数据的顺序。使用向上箭头和向下箭头可更改此顺序。

筛选器属性

可以指定在添加标签操作期间要跳过的值。 使用**忽略文本**属性可以指定不会应用添加标签操作的值。

下表介绍了筛选器属性：

属性	说明
搜索术语	指定转换在执行添加标签操作之前筛选的字符串。 使用此功能可指定定义的添加标签策略的例外情况。
区分大小写	确定筛选字符串与搜索术语的大小写是否必须匹配。
大写	将筛选出的字符串转换为大写。
启动	指定要开始搜索筛选字符串的字符位置。
结束	指定要停止搜索筛选字符串的字符位置。

“为标志添加标签”属性

在标签创建器转换的**策略**视图中配置“为标志添加标签”操作属性。

常规属性

常规属性将应用于策略中定义的所有“为标志添加标签”操作。 使用常规属性命名策略、指定输入和输出端口并指定策略是否支持概率匹配技术。

下表介绍了常规属性：

属性	说明
名称	提供策略名称。
输入	标识策略操作可以读取的输出端口。
输出	标识策略操作可以写入的输出端口。

属性	说明
说明	描述策略。 该属性为可选。
使用概率匹配技术	指定策略可以使用概率模型来标识标志类型。
已启用反向	指示策略按从右到左的顺序读取输入数据。 已对概率匹配禁用此属性。
分隔符	指定转换用于计算输入数据内的子字符串的字符。 默认值为空格。 该属性在随机标签中处于禁用状态。
标志化的输出字段	指示策略将多个标签写入一个输出端口。 选择此字段可以为解析器转换中的基于模式的解析创建输入数据。
得分输出字段	标识包含在概率匹配中生成的得分值的字段。 如果选择选项以使用概率匹配技术，则设置得分输出字段。
输出分隔符	指定用于分隔输出端口数据值的字符。 默认是冒号。

标志集属性

当您配置添加标签操作以使用标志集时，将应用标志集属性。

下表介绍了常规属性：

属性	说明
选择标志集	指定转换用于向字符串添加标签的标志集。
筛选文本	筛选标志集或正则表达式的列表。使用文本字符和通配符作为筛选器。
添加标志集	用于定义自定义标志集。
添加正则表达式	用于定义与输入模式匹配的正则表达式。
编辑	编辑自定义标志集或正则表达式的内容。
导入	从模型存储库的文件夹中导入标志集或正则表达式的不可重用副本。如果更新标志集或正则表达式的源对象，数据集成服务不会更新不可重用副本。
删除	删除自定义标志集或正则表达式。
指定执行顺序	设置操作向数据应用标志集或正则表达式的顺序。使用向上箭头和向下箭头可更改此顺序。

自定义标签属性

配置标志标签操作时，可以选择**自定义标签**视图以创建特定搜索词的标签。

下表介绍了自定义标签属性：

属性	说明
搜索术语	标识要搜索的字符串。
区分大小写	指定输入数据是否必须匹配搜索词的大小写。
自定义标签	标识要应用的自定义标签。

概率匹配属性

选择选项以使用概率匹配技术时，可以将概率模型添加到“添加标签”操作。 不能将概率模型添加到使用标志集或引用表的策略。

下表介绍了与概率匹配关联的属性：

属性	说明
名称	请提供操作名称。
筛选文本	使用您输入的字符或通配符来筛选存储库中的概率模型列表。
概率模型	标识要在操作中使用的概率模型。

引用表属性

将添加标签操作配置为使用引用表时，将应用引用表属性。

下表介绍了引用表属性：

属性	说明
名称	请提供操作名称。
引用表	指定操作用来为标志添加标签的引用表。
标签	指定输入字符串与引用表条目相匹配时操作将写入新端口的文本。
区分大小写	确定输入字符串是否必须与引用表条目的大小写相匹配。
将匹配项替换为有效值	将已添加标签的字符串替换为引用表“有效”列中的条目。
模式	确定为标志添加标签的方法。 选择“包含”以为匹配引用表条目的输入字符串添加标签。选择“独占”以为不匹配引用表条目的输入字符串添加标签。
设置优先级	确定在策略中，为引用表添加标签的操作是否优先于为标志集添加标签的操作。 如果设置此属性，转换会先为引用表添加标签，然后再为标志集添加标签，且标志集分析无法覆盖引用表标签分析。

配置“为字符添加标签”策略

要配置添加标签策略，请在标签创建器转换的**策略**视图中编辑相应设置。

1. 选择**策略**视图，然后单击**新建**以创建策略。

   此时将打开**策略**向导。
2. 输入策略名称。
3. 单击**输入**和**输出**字段以定义策略端口。
4. （可选）输入策略的说明。
5. 选择“为字符添加标签”模式。
6. 单击**下一步**。
7. 选择要配置的“为字符添加标签”操作类型。 可以配置以下操作：
   - 使用引用表为字符添加标签。
   - 使用字符集为字符添加标签。
8. 单击**下一步**。
9. 配置操作属性，然后单击**下一步**。
10. （可选）配置**忽略文本**属性。
11. 单击**下一步**以将更多的操作添加到策略，或者单击**完成**。

可以更改转换处理策略和操作的顺序。 在**策略**视图中选择一个策略或操作，然后单击**上移**或**下移**。

配置“为标志添加标签”策略

要配置添加标签策略，请在标签创建器转换的**策略**视图中编辑相应设置。

1. 选择**策略**视图，然后单击**新建**以创建策略。

   此时将打开**策略**向导。
2. 输入策略名称。
3. 单击**输入**和**输出**字段以定义策略端口。
4. （可选）输入策略的说明。
5. 选择“为标志添加标签”模式。

   验证或编辑选定模式的属性。
6. 单击**下一步**。
7. 选择要配置的“为标志添加标签”操作类型。 可以配置以下操作：
   - 使用标志集为标志添加标签。
   - 使用引用表为标志添加标签。
   - 使用概率匹配为标志添加标签。
8. 单击**下一步**。
9. 配置操作属性，然后单击**下一步**。

   如果要配置策略以使用概率匹配，请输入一个标签以用于标识为剩余数据的标志。
10. （可选）配置**自定义标签**属性。

11. 单击**下一步**以将更多的操作添加到策略，或者单击**完成**。

可以更改转换处理策略和操作的顺序。 在**策略**视图中选择一个策略或操作，然后单击**上移**或**下移**。

标签创建器转换高级属性

配置属性以帮助确定数据集成服务如何处理标签创建器转换的数据。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的标签创建器转换

非本地环境中的标签创建器转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中无限支持。在流映射中不受支持。
- Databricks Spark 引擎。无限支持。

第 25 章

查找转换

本章包括以下主题：

查找转换概览

查找转换是指在平面文件、逻辑数据对象、引用表或关系表中查找数据的被动或主动转换。查找转换可以从一个查找中返回一行或多行。

创建查找转换之前，请先创建查找源。导入平面文件或关系数据库表作为物理数据对象。或者，创建逻辑数据对象或引用表以用作查找源。创建查找转换时，Developer tool 会添加数据对象或引用表中的列作为转换中的查找端口。创建转换后，配置一个或多个输出端口以返回查找结果。配置查找条件和其他查找属性。

运行映射或预览数据时，集成服务会查询查找源。集成服务根据转换中的查找端口、查找属性和查找条件查询查找源。查找转换会将查找的结果返回到目标或其他转换中。

可以配置已连接或未连接的查找转换。已连接的转换连接到映射中的其他转换。未连接的转换从其他转换中的 :LKP 表达式接收输入。如果查找转换对逻辑数据对象执行查找，则必须配置已连接的查找转换。将查找转换输入端口连接到上游转换或上游源。将输出端口连接到下游转换或下游目标。

您可以在一个映射中使用多个查找转换。

可以使用查找转换执行以下任务：

- 获取相关值。根据输入数据中的值从查找源中检索值。例如，输入数据中包含员工 ID。按员工 ID 从查找源中检索员工姓名。
- 从查找源中检索多个行。
- 执行计算。从查找表中检索值并在计算中使用该值。例如，检索销售税百分比、计算税款并将税款返回到目标。
- 在接受表达式的转换中使用 :LKP 表达式执行未连接的查找。在该转换中使用其他表达式来筛选结果。
- 参数化查找源和查找条件，以在动态映射中使用查找转换。

已连接和未连接的查找

可以配置已连接的查找转换或未连接的查找转换。 已连接的查找转换是具有连接到映射中其他转换的输入端口和输出端口的转换。 未连接的查找转换将显示在映射中，但未连接到其他转换。

未连接的查找转换将从表达式转换或汇总器转换等转换中的 :LKP 表达式的结果中接收输入。 :LKP 表达式会将参数传递至查找转换，并从查找转换中接收结果。:LKP 表达式可以将查找结果传递至表达式转换或汇总器转换中的其他表达式以筛选结果。

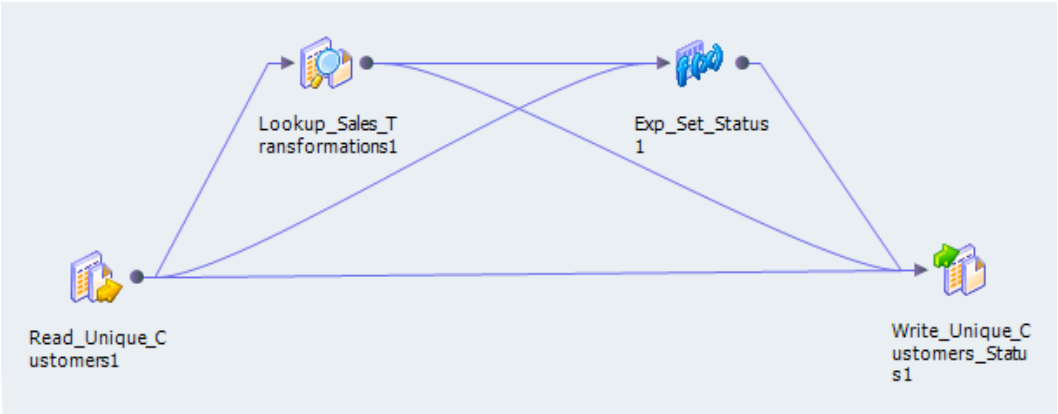
下表列出了已连接的查找与未连接的查找之间的差异：

已连接的查找	未连接的查找
直接从管道中接收输入值。	从其他转换的 :LKP 表达式的结果中接收输入值。
使用动态或静态缓存。	使用静态缓存。
缓存中包括查找条件中的查找源列和作为输出端口的查找源列。	缓存中包括查找条件中的所有查找端口和输出端口以及查找/返回端口。
从同一行返回多列或者插入到动态查找缓存中。	从每一行返回一列至返回端口。
如果不存在与查找条件匹配的项，则集成服务将为所有输出端口返回默认值。如果配置动态缓存，集成服务会将行插入到缓存中或者保持行不变。	如果不存在与查找条件匹配的项，则集成服务将返回 NULL。
如果存在与查找条件匹配的项，则集成服务将为所有查找/输出端口返回查找条件的结果。如果配置动态缓存，集成服务会更新缓存中的行或者保留行不变。	如果存在与查找条件匹配的项，则集成服务会将查找条件的结果返回至返回端口。
将多个输出值传递到其他转换。将查找/输出端口链接至其他转换。	将一个输出值返回至其他转换。查找转换返回端口会将值传递到其他转换中包含 :LKP 表达式的端口。
支持用户定义的默认值。	不支持用户定义的默认值。

已连接的查找

已连接的查找转换是连接到映射中的源或目标的查找转换。

下图显示了具有已连接的查找转换的映射：



当运行包含已连接查找转换的映射时，集成服务将执行以下步骤：

1. 集成服务会将值从其他转换传递至查找转换的输入端口。
2. 对于每个输入行，集成服务将根据转换中的查找端口和查找条件查询查找源或缓存。
3. 如果该转换未缓存，或者该转换使用静态缓存，则集成服务将返回查找查询中的值。

   如果转换使用动态缓存，则集成服务在缓存中找不到行时将行插入缓存。 集成服务在缓存中找到行时，会更新缓存中的行或者保留不更改。 它将行标记为插入、更新或未更改。
4. 集成服务将返回查询中的数据，并将该数据传递至映射中的下一转换。

   如果转换使用动态缓存，可以将行传递给筛选器或路由器转换以将新行筛选到目标中。

注意: 除非另行指定，否则本章将讨论已连接的查找转换。

未连接的查找

未连接的查找转换是指未连接到映射中的源或目标的查找转换。 在支持表达式的转换中使用 :LKP 表达式调用查找。

查找表达式的语法为 :LKP lookup_transformation_name(argument, argument, ...)

列出每个参数的顺序必须与查找转换中查找条件的顺序一致。查找转换通过查找转换返回端口返回查询的结果。调用此查找的转换会在包含 :LKP 表达式的端口中收到查找结果值。如果此查找查询无法返回值，则端口服务将收到空值。

执行未连接的查找时，可在同一映射中多次执行相同的查找。可在另一表达式中测试查找结果，并基于这些结果来筛选行。

如果运行的映射包含未连接的查找转换，则集成服务将执行以下步骤：

1. 未连接的查找转换将从汇总器转换、表达式转换或更新策略转换等其他转换中的 :LKP 表达式的结果中接收输入值。
2. 集成服务根据查找转换中的查找端口和条件来查询查找源或缓存。
3. 集成服务通过查找转换的返回端口返回一个值。
4. 集成服务将返回值传递到包含 :LKP 表达式的端口。

开发查找转换

开发查找转换时，需要考虑查找源类型和查找条件等因素。

开发查找转换时，请考虑以下因素：

- 您是否要从平面文件、逻辑数据对象、引用表或关系数据对象创建转换。 创建查找转换之前，请先创建查找源。导入平面文件或关系数据库表作为物理数据对象。或者，创建逻辑数据对象或引用表以用作查找源。
- 转换的输出端口。
- 转换中的查找条件。
- 是否希望集成服务对查找数据进行缓存。 集成服务可以缓存平面文件、引用表或关系数据对象的数据。

查找查询

集成服务根据在查找转换中配置的端口和属性来查询查找。 当第一行进入查找转换时，集成服务将运行默认查找查询。

如果对关系表使用查找，可以替代查找查询。可以使用替代来添加 WHERE 子句或在缓存查找数据之前转换该数据。

如果在查找查询上配置 SQL 替代和筛选器，集成服务将忽略筛选器。

默认查找查询

默认查找查询包含以下语句：

SELECT

SELECT 语句包括映射中的所有查找端口。要查看查找查询的 SELECT 语句，请选择使用自定义查询属性。

ORDER BY

ORDER BY 子句按照列在查找转换中出现的顺序对列进行排序。集成服务生成 ORDER BY 子句。生成默认 SQL 时不能对此进行查看。

查找查询的 SQL 替代

您可以替代关系查找的查找查询。可以添加 WHERE 子句并在缓存前转换查找数据。

您可以在表名称和列名称中使用预留字和正斜杠。

您可以输入一个查询来完全替代默认查找查询。或则，您可以查看和编辑默认查找查询。默认查找查询包括查找端口、输出端口和返回端口。

使用 SQL 替代时，查询会追加子句 ORDER BY 1。该子句将对数据进行排序，以便可靠地为其他子句提供第一个值和最后一个值。

注意: 在 Hive 命令行实用程序中运行以下查询可以手动验证 SQL：

```
CREATE VIEW <table name> (<port list>) AS <SQL>
```

其中：

- <table name> 是您选择的名称
- <port list> 是源中端口的逗号分隔列表

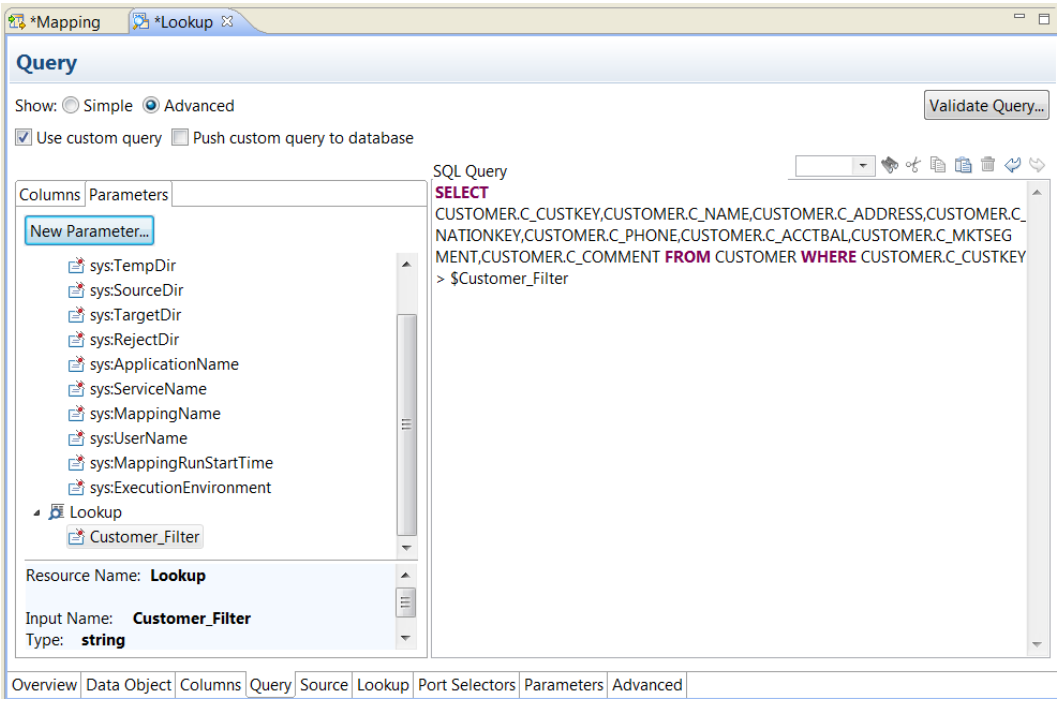
- <SQL> 是要验证的查询

SQL 替代查询中的参数

可以在查找转换的查找查询中使用系统参数或用户定义的参数。 SQL 编辑器提供了一个可供您选择的系统参数和用户定义参数的列表。

您可以浏览查找用户定义的参数，也可以在查找转换的**查询**选项卡中进行创建。 为每个参数定义一个默认值。 可以在向映射添加查找转换之后将 mapplet 或映射参数绑定到转换参数，从而替代默认值。

下图显示了查找转换的“查询”选项卡：



保留字

如果任何查找名称或列名称包含数据库预留字（例如 MONTH 或 YEAR），则当集成服对数据库执行 SQL 时映射会失败，并出现数据库错误。

您可以在集成服务安装目录中创建并维护一个预留字文件 reswords.txt。 当集成服务初始化映射时，它会搜索 reswords.txt 文件并在预留字两侧加上引号，然后对源、目标和查找数据库执行 SQL。

您可能需要启用某些数据库（如 Microsoft SQL Server 和 Sybase）才能使用关于加引号的标识符的 SQL-92 标准。 使用环境 SQL 发出命令。 例如，对于 Microsoft SQL Server，使用以下命令：

```
SET QUOTED_IDENTIFIER ON
```

替代查找查询的准则

替代查找查询时，应使用特定规则和准则。

替代查找 SQL 查询时，请考虑以下准则：

- 可以替代关系查找的查找 SQL 查询。

- 添加源查找筛选器以筛选添加到查找缓存中的行。这样可确保集成服务在与 WHERE 子句匹配的动态缓存和目标表中插入行。
- 如果多个查找转换共享一个查找缓存，请对每个查找转换使用相同的查找 SQL 替代。
- 如果查找查询中的表名或列名包含保留字，应为保留字加上引号。
- 要替代未缓存查找的查找查询，应选择在集成服务找到多个匹配项时返回任何值。
- 不能在默认 SQL 语句中添加或删除任何列。
- Developer tool 不会验证 SQL 查询的语法。如果未连接的查找查询中的 SQL 替代无效，映射将失败。

替代查找查询

可以替代默认查找 SQL 查询，以在查找源上创建自定义查询。

1. 在**属性**视图上，选择**查询**选项卡。
2. 选择**高级**。
3. 选择**使用自定义查询**。
4. 在 SQL 查询区域编辑查找查询。

   可以双击表名、列名或参数以将其添加到查询中。
5. 单击**验证查询**以验证查找查询。
6. 选择**将自定义查询推入数据库**以在数据库中运行查找查询。

查找源筛选器

您可以为已启用缓存的关系查找转换配置查找源筛选器。 添加查找源筛选器以限制集成服务在查找源表中执行的查找数量。

配置查找源筛选器时，集成服务基于筛选器语句的结果执行查找。 例如，您可能需要检索 ID 大于 510 的每位员工的姓氏。

在 EmployeeID 列上配置以下查找源筛选器：

```
EmployeeID >= 510
```

EmployeeID 是查找转换中的输入端口。当集成服务读取源行时，如果 EmployeeID 值大于 510 它将在缓存上执行查找。当 EmployeeID 小于或等于 510 时，查找转换不会检索姓氏。

当为针对下推优化配置的映射向查找查询添加查找源筛选器时，集成服务会创建一个视图来表示 SQL 替代。 集成服务针对此视图运行 SQL 查询以将转换逻辑推送到数据库中。

筛选查找中的源行

您可以为已启用缓存的关系查找转换配置查找源筛选器。 筛选查找中的源行以限制集成服务在查找源表中执行的查找数量。

1. 在**属性**视图上，选择**查询**选项卡。
2. 在**筛选器**选项中，单击**编辑**。
3. 在 SQL 编辑器中，选择输入端口或输入您想要筛选的任何查找转换端口。
4. 输入筛选条件。

不在筛选条件中包括关键字 WHERE。 将字符串映射参数和变量包含在字符串标识符内。

5. 单击**验证查询**验证筛选条件的语法。

查找条件

数据集成服务根据查找条件在查找源中查找数据。 配置查找转换中的查找条件时，将源数据中一列或多列的值与查找源或缓存中的值进行比较。

例如，源数据中包含 employee_number。 查找源表中包含 employee_ID、first_name 和 last_name。 配置以下查找条件：

```
employee_ID = employee_number
```

对于每个 employee_number，数据集成服务将从查找源中返回 employee_ID、last_name 和 first_name 列。

数据集成服务可以从查找源中返回多行。 配置以下查找条件：

```
employee_ID > employee_number
```

数据集成服务将返回所有大于源员工编号的 employee_ID 编号的行。

数据对象查找中的 Null 值

查找条件的输入为 NULL 时，数据对象查找转换会返回单一行，其中包含仅输出端口的空值以及来自传递端口的输入行的值。

例如，以下查找条件对数据源执行查找操作，数据源中有一行或多行的 employee_ID 值为 NULL：

```
employee_ID = employee_number
```

在本示例中，使用包含以下数据的查找表：

EMPLOYEE_ID	LAST_NAME
1294765	Hara
1356356	Carver
1407207	NULL
1570348	Draper
NULL	Limonov

将数据源中的以下输入值与查找表进行比较：

```
EMPLOYEE_NUMBER
---------------
1294765
1356356
1407207
1648246
NULL
```

在本示例中，查找条件生成以下结果：

```
1294765,Hara
1356356,Carver
1407207,NULL
NULL,NULL
NULL,NULL
```

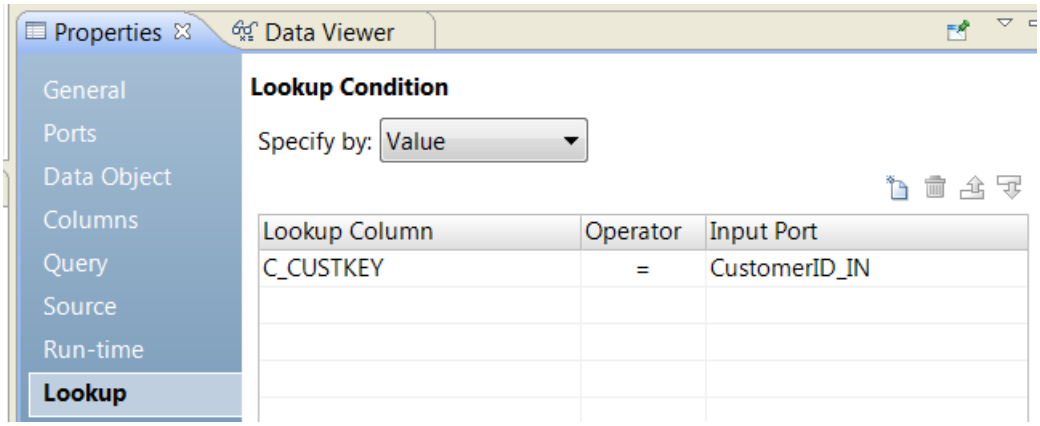
对于前两行，查找条件找到了 EMPLOYEE_ID 和 EMPLOYEE_NUMBER 匹配的项目。 对于第三行，查找源包含一个值为 NULL 但未加入查找条件的行。 它与查找条件相匹配，因此返回了一个结果，其中非查找列的值为 NULL。

对于第四行和第五行，查找条件未找到匹配项，因此返回的两个值均为 NULL。 请注意第五行，查找条件也未找到匹配项，因为 NULL 与任何值都不匹配，包括 NULL 本身。

配置查找条件

查找条件是用于确定要从查找源检索的行的表达式。 在**属性**视图的**查找**选项卡上配置查找条件。

下图显示了按客户编号执行查找的查找条件：



可以在**查找**选项卡上配置以下选项：

指定依据

选择**值**可选择查找列和输入端口名称。 选择**参数**可配置表达式参数以定义查找条件。

查找列

查找源中要与输入行中的某列相匹配的列。 可以在查找条件中包含多个列。

运算符

确定查找列和输入端口之间的搜索条件的运算符。 运算符包括 =、!=, >, < , >=, <=.

输入端口

包含要在查找源中搜索的值的输入端口。 可以将多个输入端口与查找源中的端口进行比较。

查找转换条件的规则和准则

输入查找转换的条件时，需要应用一定的规则和准则。

输入查找转换的条件时，请考虑以下规则和准则。

- 查找条件中列的数据类型必须匹配。
- 将一个输入端口用于查找条件中的每个查找端口。可以在转换的多个条件中使用同一输入端口。
- 如果在查找条件中使用端口选择器或动态端口，该查找条件将考虑表达式中的所有端口。
- 可以使用动态输入端口或端口选择器作为查找条件的输入端口。 输入端口中生成的端口数量必须与查找列中的端口数量相等。
- 处理具有多个查找条件的查找转换时，集成服务将重运行匹配所有查找条件的行。

- 可以创建一个表达式参数，以便在不可重用的查找转换中对查找条件进行参数化。
- 要提高查找性能，请按以下顺序输入条件：

  - 等于 (=)

  - 小于 (<)、大于 (>)、小于或等于 (<=)、大于或等于 (>=)

  - 不等于 (!=)
- 创建查找条件时，请使用以下运算符之一：=、>、<、>=、<= 或 !=.
- 集成服务将根据您是使用动态查找缓存、静态查找缓存还是未缓存查找配置查找转换，对查找匹配进行不同处理。
- 集成服务会匹配空值。例如，如果查找端口和输入端口都具有空值，集成服务会将其视为等同。
- 如果查找条件中列的数据类型是 Decimal，则每列的精度都必须属于同一精度范围。 有效的精度范围包括：

  - 小数 0-18

  - 小数 19-28

  - 十进制，29-38 位数

  - 十进制，39 位数及以上

  例如，如果定义了条件 `DecimalA = DecimalB`（其中，DecimalA 精度为 15，DecimalB 精度为 25），则该查找条件无效。

查找缓存

通过缓存大查找源或小查找源，可以提高性能。缓存查找源时，集成服务将查询查找缓存，而不是查询每个输入行的查找源。

根据业务需求，您可以创建不同类型的查找缓存。可以创建静态或动态缓存。可以创建持久性缓存或非持久性缓存。可以在多个查找转换之间共享缓存。

如果查找转换位于动态映射中，您可以创建持久或非持久缓存。 保留缓存并使用参数更改查找源时，映射将失败。 如果更改平面文件查找源的控制文件，映射也将失败。

注意: 查找转换包含动态端口或参数化查找源时，不能使用动态查找缓存或持久查找缓存。

查询属性

配置查询属性以查看或更改关系查找表上的查找查询。 可以对查找应用筛选器或者自定义查找查询。

下表介绍了执行关系查找的查找转换的查询属性：

属性	说明
简单	选择可查看查找查询并对查找应用筛选器。
高级	选择可在包含关系查找表的数据库中查看查询、自定义查询或运行查询。
筛选器	输入筛选器以减少集成服务查询的行数。 必须选择**简单**选项才能查看此选项。

属性	说明
使用自定义查询	选择可覆盖查找查询。 必须选择**高级**选项才能查看此选项。
将自定义查询推入数据库	选择可在包含关系查找表的数据库中运行查询。 必须选择**高级**选项才能查看此选项。
SQL 查询	显示用于执行查找的 SQL 查询。 可以自定义 SQL 查询。 必须选择**高级**选项才能查看此选项。

动态映射中的查找转换

可以在动态映射中使用查找转换。 可以将动态端口配置为根据源数据接收和返回不同的端口。 可以参数化查找源和查找条件以根据不同的端口执行查找。

动态映射是指在运行时源、目标及转换逻辑可能发生变化的映射。 您可以设置参数和规则来更改数据的结构。 在动态映射中使用查找转换时，查找转换的输入端口可能会根据源数据而发生更改。 查找源的结构和查找条件中的端口可能会更改。

注意: 如果查找转换包含动态端口或参数化的查找源，则无法保留查找缓存， 也无法配置动态缓存。

可针对查找转换执行以下任务，以在动态映射中使用该转换：

定义动态端口

定义动态端口和生成的端口以适应输入列更改。

参数化查找源

为定义查找源的数据对象分配一个参数。 您可以在不可重用的查找转换中参数化查找源。

定义端口选择器

定义一个指定要在查找条件中使用的端口的端口选择器。 您可以在不可重用的查找转换中参数化端口选择器端口。

参数化查找条件

创建一个表达式参数并定义一个包含整个表达式的默认值。

有关动态映射的详细信息，请参阅*《Informatica Developer 映射指南》*。

定义动态端口

可以在查找转换中定义动态端口。

可以在查找条件的输入列中引用动态端口。 如果动态端口包含多个生成的端口，则可以将端口选择器用于查找条件的查找列元素。 动态输入端口必须包含与查找条件中的端口选择器相同数量的端口。

如果动态端口包含一个值，则可以在查找条件的查找列元素中使用单个端口。

可以在查找条件中引用生成的端口。 但是，如果动态映射中的源发生变化，则生成的端口可能不存在。 映射将失败。

更改查找源

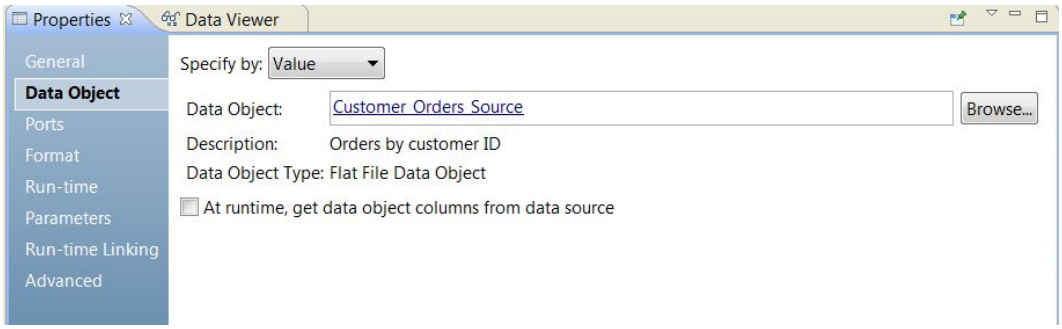
可以更改作为可重用查找转换查找源的数据对象。可以配置一个参数，以确定在运行时要用作查找源的数据对象。

从物理数据对象创建转换时，有关数据对象的信息将显示在转换属性的**数据对象**选项卡上。您可以单击数据对象名称来查看模型存储库中的物理数据对象定义。

您可以通过浏览到模型存储库中的其他物理数据对象来更改转换的数据对象。当您更改数据对象时，转换将使用您选择的数据对象的运行时属性和高级属性。

可以在运行时根据数据源更改来更新数据对象的结构。数据源是数据对象所表示的物理文件或数据库表。启用数据集成服务从数据源获取数据列时，数据集成服务会检查数据源的结构。数据集成服务会根据数据源来更新转换实例中的数据对象端口，但不会更改模型存储库中的物理数据对象定义。

下图显示了**数据对象**选项卡：



数据对象选项卡具有以下字段：

指定依据

选择**值**可输入特定数据对象名称。选择**参数**可参数化数据对象。

数据对象

模型存储库中的数据对象的名称。您可以单击**数据对象**链接来打开存储库中的数据对象定义。还可以浏览模型存储库中的其他数据对象。

说明

存储库中的数据对象的说明。只读。

数据对象类型

描述数据对象的类型，如平面文件数据对象、关系表对象或自定义数据对象。

运行时，从数据源获取数据对象列

数据集成服务从数据对象所引用的数据文件或表中获取元数据和数据定义更改，并在运行时更新转换实例的数据对象结构。

要预览数据集成服务如何在运行时获取元数据和数据定义更改，请查看具有已解析参数的映射。

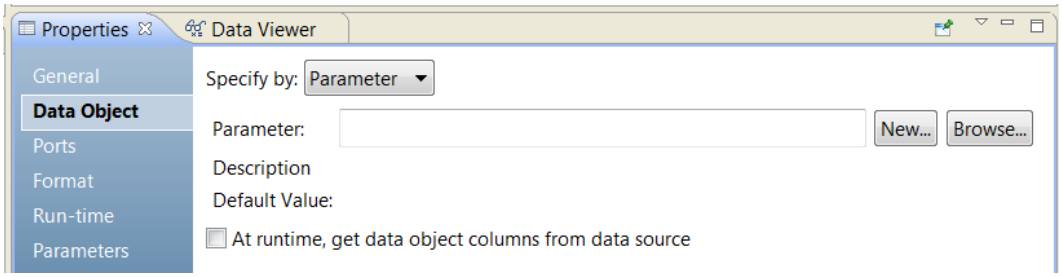
参数化查找源

可以在不可重用的查找转换中为查找源配置参数。

要参数化某个数据对象，请在**数据对象**选项卡中选择**按参数指定**。 **数据对象**选项卡中的属性发生变化。

要参数化该数据对象，请创建一个资源类型参数或浏览查找已创建的资源参数。 参数默认值是模型存储库中物理数据对象的名称。创建默认参数值时，需要从存储库中的数据对象列表中选择物理数据对象。

下图显示了按参数指定数据对象时出现的**数据对象**选项卡。



数据对象选项卡按参数列出以下选项：

参数

您配置为数据对象的资源参数的名称。 只读。

说明

参数的说明。 只读。

新建

创建资源参数。 在模型存储库中浏览参数默认值并选择数据对象。

浏览

浏览资源参数并选择该参数。

默认值

您为数据对象配置的资源参数的默认值。 默认值是物理数据对象名称以及该对象在模型存储库中的路径。 只读。

端口名称与查找端口冲突

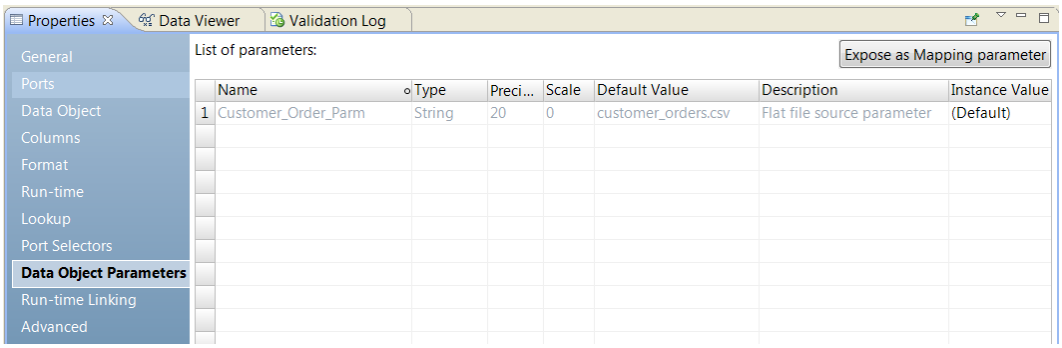
如果将查找源参数化，查找转换输入端口可能与查找源中的端口存在名称冲突。

当查找转换输入端口与查找源中的查找端口存在名称冲突时，Developer tool 不会重命名任一端口。Developer tool 会显示验证错误。必须更改查找转换中的输入端口名称，或者从转换中删除该端口。

包含参数的查找源

可以根据一个包含参数的物理数据对象来创建查找源。 将该物理数据对象添加到映射时，参数会显示在**数据对象参数**选项卡上。

下图显示了查找转换中的**数据对象参数**选项卡：



该图中显示的是 Customer_Order_Parm。 Customer_Order_Parm 参数用于指定平面文件数据对象中的源文件名。 要在映射中替代该源文件名，请将 Customer_Order_Parm 绑定到映射中的某个参数。 单击**公开为映射参数**可在映射中创建重复参数。

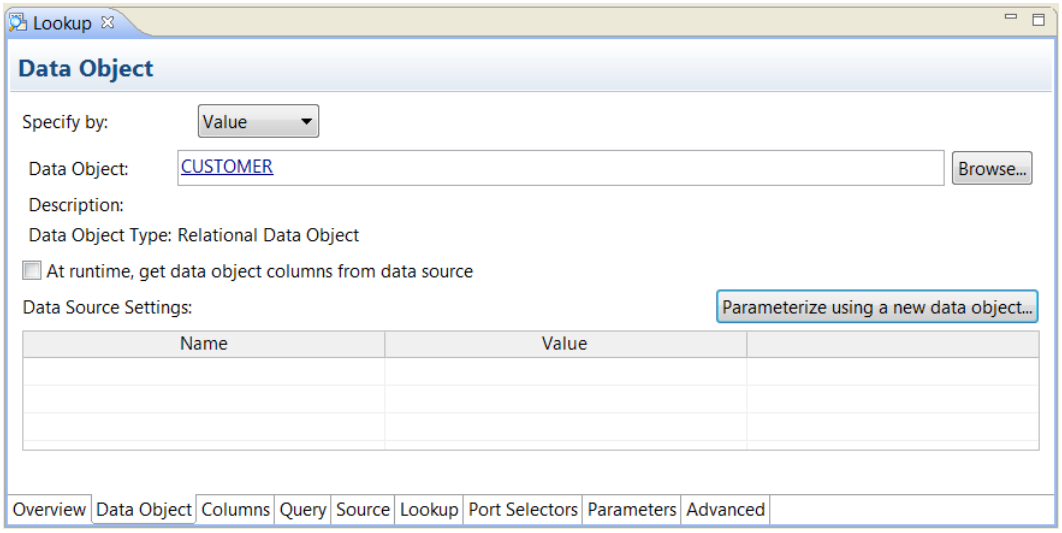
在重复数据对象中配置参数

可以在存储库中创建重复数据对象并参数化此物理数据对象的属性。 为连接、资源名称、表所有者或控制文件名等属性定义默认值。

可以为关系数据对象和平面文件数据对象创建重复数据对象。 在可重用查找转换和不可重用查找转换中均可创建重复数据对象。

在查找转换的**数据对象**选项卡中创建重复数据对象。 如果将数据对象指定为值，则可以创建重复对象。 创建重复数据对象时，使用重复数据对象名称替换查找转换中的数据对象名称。 Developer tool 为数据对象属性创建参数， 并提示您输入参数的默认值。 重复数据对象名称语法为：<原始对象名称>_Param。

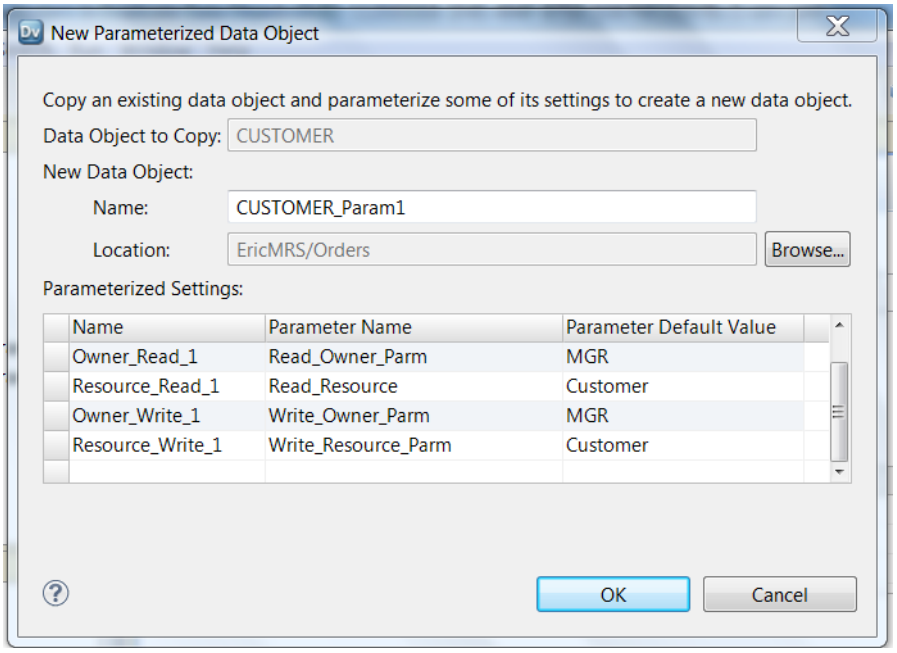
下图显示了关系和平面文件数据对象的**数据对象**选项卡上的**使用新数据对象执行参数化**按钮：



通过创建重复数据对象来参数化数据对象时，Developer tool 将为该数据对象创建一组参数。 Developer tool 根据数据对象是平面文件还是关系数据对象创建不同的参数。

创建重复数据对象时，在**新建参数化数据对象**对话框上配置默认参数值。

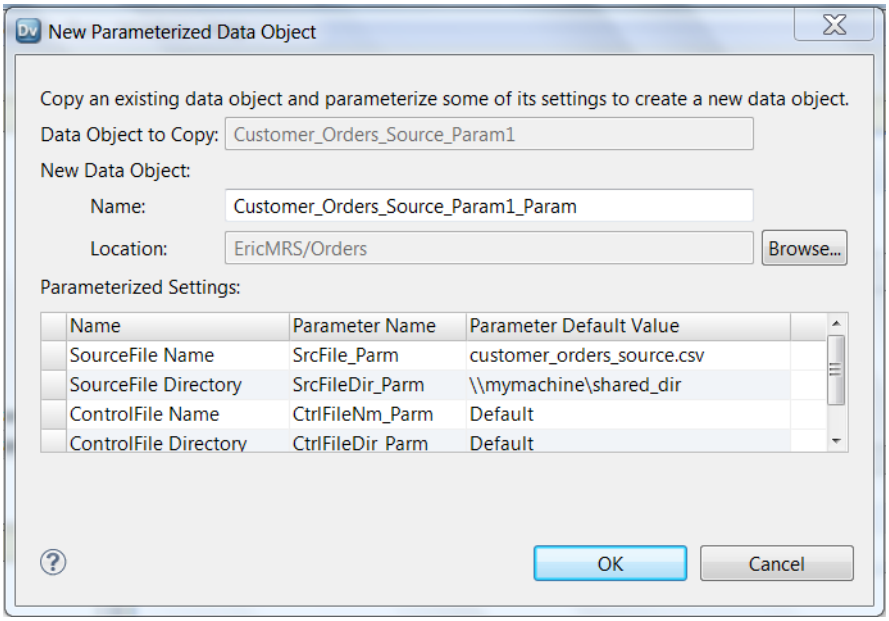
下图显示的是针对关系数据对象的**新建参数化数据对象**对话框：



可以更改数据对象的名称。 为所有者和资源参数输入参数默认值。

如果原始数据对象已参数化，Developer tool 会将原始数据对象中的参数复制到重复数据对象。 如果原始属性未参数化，Developer tool 会在重复数据对象中为其创建一个参数。 Developer tool 使用原始属性值作为重复数据对象中的默认参数值。 如果 Developer tool 无法确定原始属性值，它将根据参数类型使用默认值创建参数。

下图显示的是针对平面文件数据对象的**新建参数化数据对象**对话框：



为源文件及其目录配置默认参数值。 如果平面文件具有控制文件，则配置控制文件名称和目录。

配置默认值后，Developer tool 将创建重复数据对象。 重复数据对象名称显示在查找转换的**数据对象**选项卡上。 重复数据对象显示在**对象浏览器**中。

创建数据对象后，如果要更改其参数值，请在**对象浏览器**中打开物理数据对象。 单击**参数**选项卡。

端口选择器

当查找转换包含生成的端口时，您可以创建查找条件。可以在查找条件中引用动态端口或端口选择器。也可以使用表达式参数来参数化整个查找表达式。

如果动态端口包含多个生成的端口，则可以定义一个端口选择器来筛选查找条件中的生成的端口。查找源在动态映射中可能发生更改。您可以配置一个端口选择器来筛选要用于查找列的端口。查找源端口选择器必须包含与输入列端口选择器相同数量的端口。

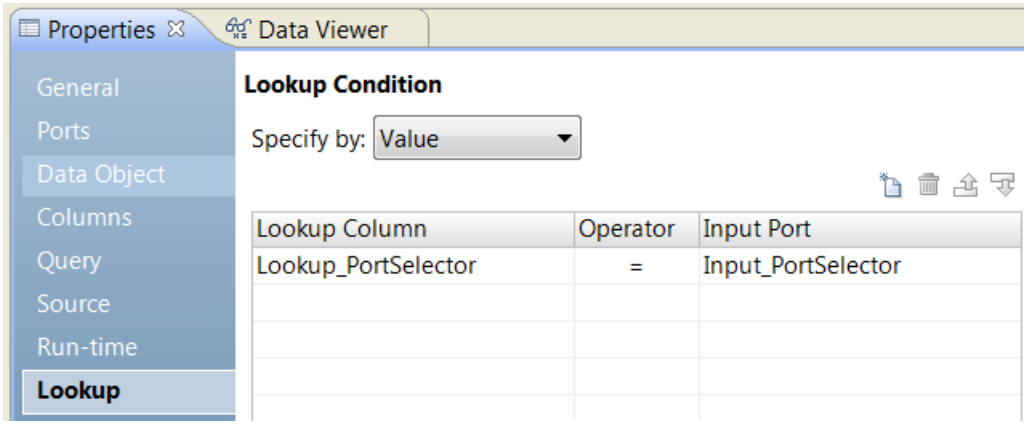
例如，Lookup_PortSelector 包含以下端口：

```
C_CustKey
C_OrderKey
```

Input_PortSelector 包含以下输入端口：

```
CustomerID_IN
OrderID_IN
```

下图显示了包含端口选择器的查找条件：



查找条件扩展为以下表达式：

```
C_CustKey = CustomerID_IN AND C_OrderKey = OrderID_IN
```

如果查找条件包含多个端口，您可以配置一个运算符。例如，可以将运算符更改为大于 (>)。查找条件扩展为以下表达式：

```
C_CustKey > CustomerID_IN AND C_OrderKey > OrderID_IN
```

可以创建一个包含动态端口的查找条件：

```
Lookup_PortSelector = Dynamic_Input_Port
```

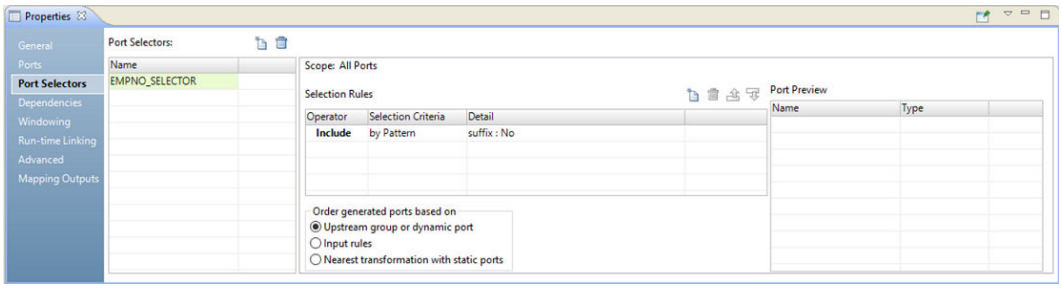
动态端口必须包含与端口选择器相同数量的端口。

端口选择器配置

配置端口选择器时，可以定义选择规则来确定要包括哪些生成的端口。选择规则与您可以为动态端口配置的输入规则类似。

端口选择器可以包括静态端口或生成的端口。在**端口选择器**选项卡上配置端口选择器。

下图显示了**端口选择器**选项卡：



配置端口选择器的以下属性：

名称

标识端口选择器。可以在一个转换中创建多个端口选择器，然后在表达式中引用它们。

范围

标识要应用端口选择器的一组端口。为联接器或查找转换创建端口选择器时，必须选择范围。这些转换包含多个输入组。联接器转换具有“主”或“详细信息”范围。查找转换具有“导入”或“查找”范围。表达式转换包含一个输入组。范围始终是“所有端口”。

选择规则

确定要包括在端口选择器中的端口。创建选择规则时，**端口预览**面板显示当前输入端口中符合条件的端口。这些端口可能会发生变化。配置选择规则以接受来自不同源的端口。

选择规则

与端口选择器相关联的选择规则决定了端口选择器中要包括的端口。

创建选择规则时，**端口预览**面板显示当前输入端口中符合条件的端口。这些端口可能会发生变化。配置选择规则以接受来自不同源的端口。

请根据以下条件创建选择规则：

运算符

包括或排除选择规则返回的端口。默认为包括。必须先包括端口，然后才能排除端口。

选择条件

要创建的选择规则的类型。您可以根据列名称、端口类型、模式或复杂数据类型定义来创建规则。要根据列名称包括端口，请搜索特定名称或搜索名称中的字符模式。

详细信息

要应用到选择条件的值。如果选择条件是按列名称，则配置要搜索的字符串或名称。如果选择条件是按端口类型，则选择要包括的端口类型。

下表介绍了选择条件以及如何指定条件的详细信息：

选择条件	说明	详细信息
全部	包括所有端口。	无需详细信息。
名称	根据端口名称筛选端口。	从值列表中选择端口名称或使用一个类型为“端口”或“端口列表”的参数。

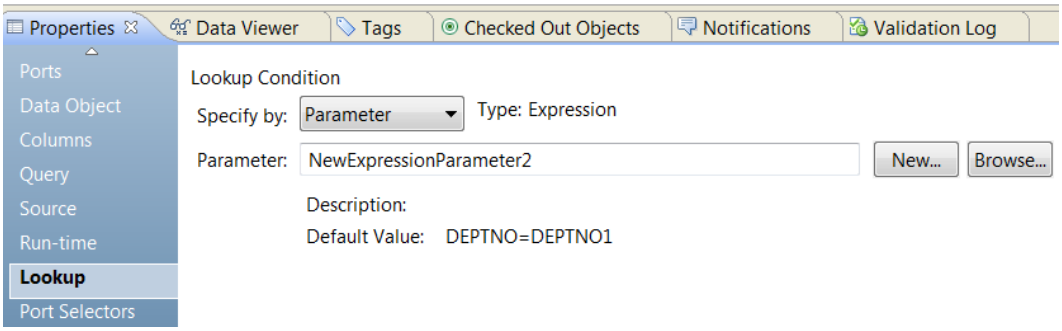
选择条件	说明	详细信息
类型	根据每个端口的数据类型筛选端口。	从列表中选择数据类型。
模式	按照名称中的字符串或按照正则表达式来筛选端口。	选择前缀、后缀或正则表达式作为端口名称的模式类型。然后，输入模式值或使用类型为“字符串”的参数。
复杂数据类型定义	按照复杂数据类型定义来筛选端口。	选择前缀、后缀或正则表达式作为复杂数据类型定义的模式类型。然后，输入模式值或使用类型为“字符串”的参数。

参数化查找条件

您可以对定义查找条件的表达式参数进行配置。 表达式参数包含您在表达式编辑器中创建的完整表达式。 您可以定义一个映射参数在运行时替代表达式参数。

如果使用参数指定查找条件，则可浏览查找表达式参数或者创建参数。

下图显示了在何处为查找条件配置表达式参数：



要创建参数，请单击**新建**。 定义参数名称并编辑默认值。 表达式参数默认值是用于定义查找条件的完全表达式。您可以在表达式中使用生成的端口、动态端口和端口选择器。

注意: 创建表达式时，查找列始终是第一个值，输入列是第二个值。

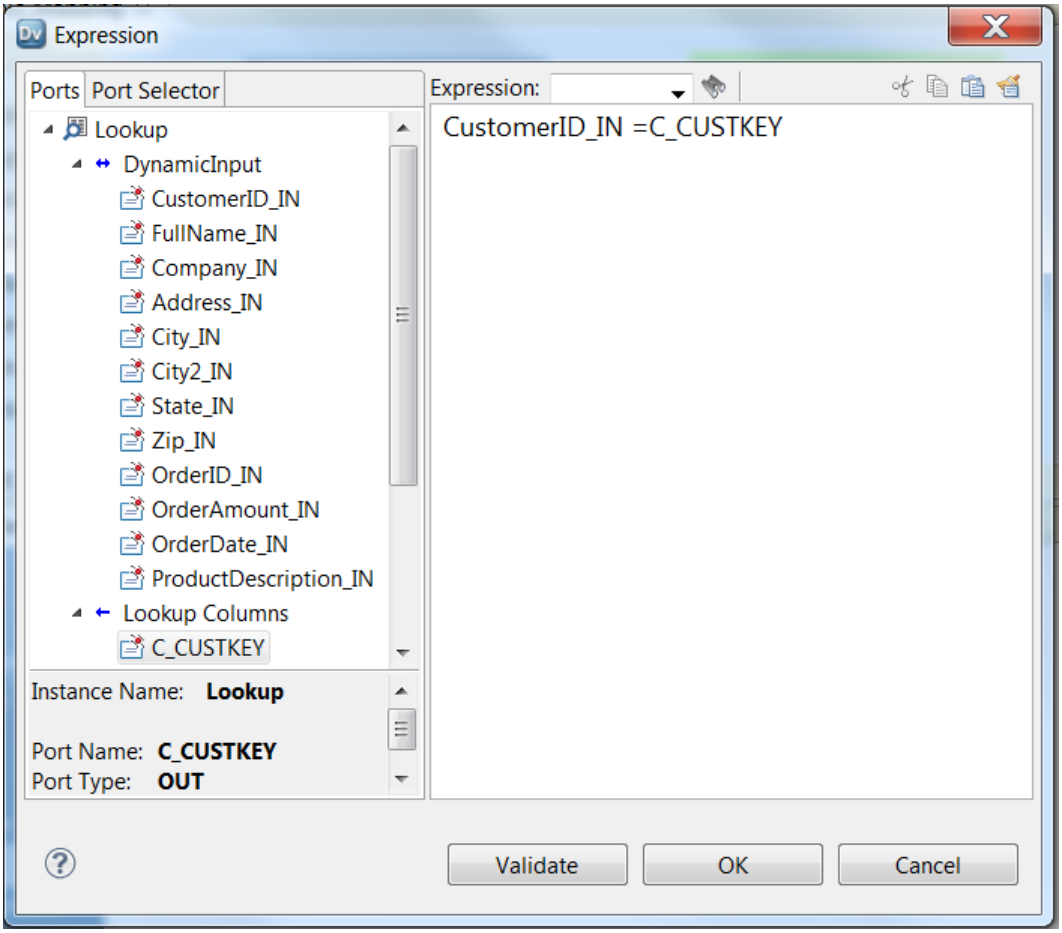
例如，您可以在表达式参数中创建以下查找条件：

```
CustomerID_IN = C_CUSTKEY
```

CustomerID_IN 是查找列。

C_CUSTKEY 是输入列。

下图显示了表达式编辑器中的查找表达式：



创建端口选择器

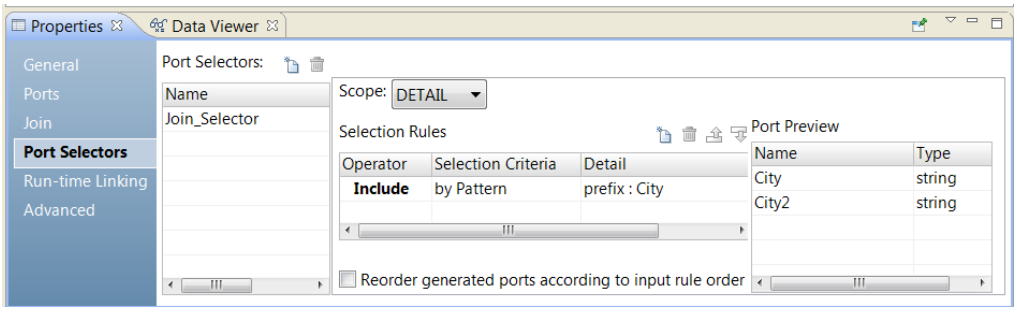
使用端口选择器可确定要在动态表达式、查找条件或联接器条件中使用的端口。

1. 单击**端口选择器**选项卡。
2. 在**端口选择器**区域中，单击**新建**。

   Developer tool 使用包含所有端口的默认选择规则来创建端口选择器。
3. 在**端口选择器**区域内，将端口选择器名称更改为唯一名称。
4. 如果您要使用联接器转换或查找转换，请选择范围。

   可用端口会根据所选的端口组发生相应变化。
5. 在**选择规则**区域内，选择**运算符**。
   - 包含。创建一个包含端口选择器端口的规则。必须先包括端口，然后才能排除端口。
   - 排除。创建一个在端口选择器中排除特定端口的规则。
6. 选择**选择条件**。
   - 按名称。按名称选择特定端口。您可以在范围内从端口列表中选择端口名称。
   - 按类型。按类型选择端口。可以选择一个或多个数据类型。
   - 按模式。按端口名称中的字符模式选择端口。可以使用特定字符进行搜索，也可以创建正则表达式。

下图显示了“端口选择器”选项卡：



7. 单击**详细信息**列。

   输入规则详细信息对话框随即打开。

8. 选择作为端口筛选依据的值。

   - 按名称。选择按值或参数来创建端口列表。单击**选择**在列表中选择端口。
   - 按类型。从列表中选择一个或多个数据类型。**端口预览**区域显示所选类型的端口。
   - 按模式。选择在端口名称的前缀或后缀中搜索特定字符模式。或者，选择创建在搜索时使用的正则表达式。配置参数或者配置在搜索时使用的模式。

   配置规则时，**端口预览**区域会显示端口选择器中的端口。

9. 要重新排序端口选择器中的端口，请选择**根据输入规则顺序重新排序生成的端口**。

运行时属性

设置运行时属性以启用和配置查找缓存。 必须现在映射中添加查找转换，然后才能配置运行时查找属性。

下表介绍了执行平面文件、引用表或关系查找的查找转换的运行时属性：

属性	说明
启用查找缓存	指示集成服务是否缓存查找值。 启用查找缓存时，集成服务将查询一次查找源，对值进行缓存，然后在缓存中查找值。缓存查找值可以提高大型查找表的性能。 如果禁用缓存，则每次将行传入转换时，集成服务都将对查找源发出一条用于查找值的 select 语句。 集成服务始终会对平面文件查找进行缓存。
查找数据缓存大小	数据集成服务在映射运行开始时为转换的数据缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。
查找索引缓存大小	数据集成服务在映射运行开始时为转换的索引缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。
缓存文件名前缀	缓存文件的前缀。您可以为持久性查找缓存指定缓存文件名前缀。

属性	说明
预构建查找缓存	允许集成服务在查找转换接收数据之前构建查找缓存。 集成服务可以同时构建多个查找缓存文件来提高性能。 配置以下选项之一： - 自动。 集成服务确定值。 - “始终允许”。 允许集成服务在查找转换接收数据之前构建查找缓存。 集成服务可以同时构建多个查找缓存文件来提高性能。 - “始终不允许”。 集成服务无法在查找转换接收第一行之前构建查找缓存。
查找缓存目录名称	将查找转换配置为缓存查找源时用于构建查找缓存文件的目录。 默认值为 CacheDir 系统参数。您可以为此属性配置另一个系统参数或用户定义的参数。
从查找源重新缓存	重建查找缓存以将持久性缓存与查找表同步。 当查找转换具有持久性查找缓存，并且查找表经常更改时，可从数据库重新缓存查找。

相关主题：

- “缓存大小” 页面上 67

高级属性

在高级属性中配置持久性查找缓存和到关系数据库的连接。显示的属性基于查找源的类型。

下表介绍了每个类型的查找源的高级属性：

属性	查找源类型	说明
查找缓存持久性	平面文件、引用表、关系查找	指示集成服务是否使用持久性查找缓存（至少由两个缓存文件组成）。如果查找转换是为持久性查找缓存配置的，但不存在持久性查找缓存文件，则集成服务将创建这些文件。
字符串比较(区分大小写)	平面文件	在字符串列上执行查找时，集成服务将使用区分大小写的字符串比较。
空值排序	平面文件	确定集成服务对空值进行排序的方式。您可以选择按高值或低值对空值进行排序。默认情况下，集成服务将按高值对空值进行排序。该操作将替代集成服务配置，以便将比较运算符中的空值视为高值、低值或空值。对于关系查找，空值排序基于数据库默认值。
已排序输入	平面文件	确定源中的数据是否已在查找转换键和数据端口上排序。如果设置为 TRUE，Informatica 在为查找结果构建缓存时会跳过数据排序。默认值为 false。
跟踪级别	平面文件、逻辑数据对象、引用表、关系查找	设置此转换的日志中显示的详细信息量。可以选择精简、普通、详细初始化或详细数据。默认值为“普通”。

属性	查找源类型	说明
多项匹配时的查找策略	平面文件	在查找查询找到多个匹配时确定结果。选择以下方法之一： - 返回第一行 - 返回最后一行 - 报告错误 - 返回任意行 - 返回所有行 默认为“返回任意行”。
忽略与以下项匹配的空值	平面文件	忽略执行查找时匹配的空值。
动态查找缓存	平面文件	用未发现的或新的查找值更新查找缓存。
首选更新，否则插入。	应用表、关系查找	仅应用于动态查找缓存。如果进入查找转换的行类型是更新，集成服务将更新缓存中的行，行存在于索引缓存中，缓存数据不同于现有行。如果行是新行，集成服务会将其插入缓存中。
首选插入，否则更新	应用表、关系查找	仅应用于动态查找缓存。如果进入查找转换的行类型是插入并且为新行，则集成服务会将该行插入缓存中。如果行存在于索引缓存中但数据缓存不同于当前行，则集成服务会更新数据缓存中的行。
更新时输出旧值	应用表、关系查找	集成服务在更新行之前输出缓存中的现有值。否则，集成服务输出其写入缓存中的更新值。
更新动态缓存条件	应用表、关系查找	仅应用于动态查找缓存。指示是否更新动态缓存的表达式。如果条件为 true 并且数据存在于缓存中，集成服务将更新缓存。默认值为 True。
连接	应用表、关系查找	到包含关系查找源的关系数据库的连接。可以将参数集用于该连接。 对于引用表查找，此字段为只读。
已排序输入	平面文件	指示输入数据已按组预先排序。
日期时间格式	平面文件	定义日期时间格式和字段宽度。毫秒、微秒或纳秒格式的字段宽度为 29。如果没有为端口选择日期时间格式，则可以输入任意日期时间格式。 默认为 YYYY-MM-DD HH24:MI:SS。日期时间格式不会更改端口的大小。此字段是只读的。
千位分隔符	平面文件	值为“无”。此字段是只读的。
小数分隔符	平面文件	值为一个句点。此字段是只读的。

创建可重用查找转换

创建查找转换以便在平面文件、逻辑数据对象、引用表或关系数据对象中查找数据。

1. **在对象浏览器**视图中选择一个项目或文件夹。
2. **单击文件 > 新建 > 转换。**
3. **浏览到“查找”向导。**
4. **选择平面文件数据对象查找、逻辑数据对象查找、引用表查找或关系数据对象查找。**

5. 单击**下一步**。

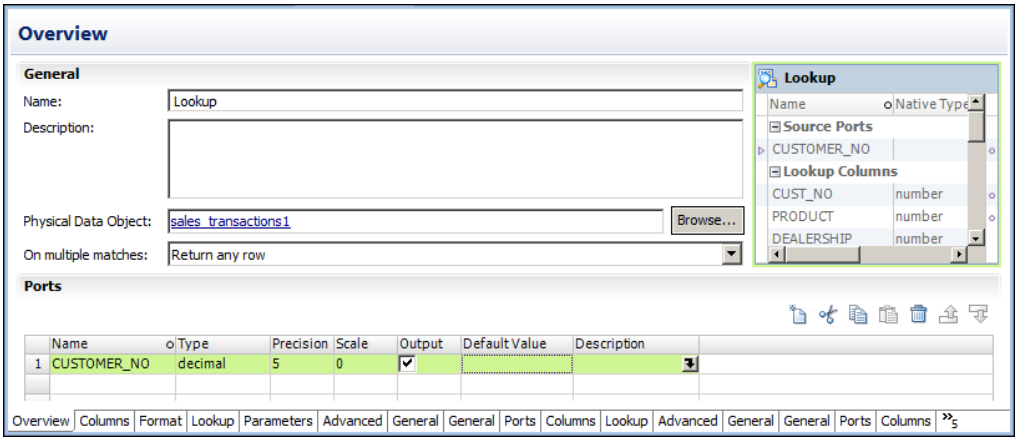
   此时将显示**新建查找转换**对话框。

6. 在 Developer 工具中选择物理数据对象或引用表。
7. 输入转换的名称。
8. 在**多项匹配时**下拉列表中，选择在查找转换找到多个与查找条件匹配的行时要使用的策略。
9. 单击**完成**。

   此时，查找转换将显示在编辑器中。

10. 在**概览**视图的**端口**部分，向转换添加输出端口。

    下图显示了查找转换中的 CUSTOMER_NO 输出端口：



11. 在**属性**视图的**运行时**选项卡上，选择**启用查找缓存**以启用查找缓存。

    注意: 必须现在映射中添加查找转换，然后才能配置运行时查找属性。

12. 在**属性**视图的**查找**选项卡上，添加一个或多个查找条件。
13. 在**属性**视图的**高级**选项卡上，配置跟踪级别、动态查找缓存属性和运行时连接。
14. 保存转换。

创建不可重用查找转换

可在映射或 Mapplet 中创建不可重用查找转换。

1. 在映射或 Mapplet 中，将“转换”选项板中的查找转换拖动到编辑器中。

   此时将显示**新建**对话框。

2. 选择**平面文件数据对象查找**、**逻辑数据对象查找**、**引用表查找**或**关系数据对象查找**。
3. 单击**下一步**。

   此时将显示**新建查找转换**对话框。

4. 在 Developer 工具中选择物理数据对象或引用表。
5. 输入转换的名称。
6. 在**多项匹配时**下拉列表中，选择在查找转换找到多个与查找条件匹配的行时要使用的策略。
7. 单击**完成**。

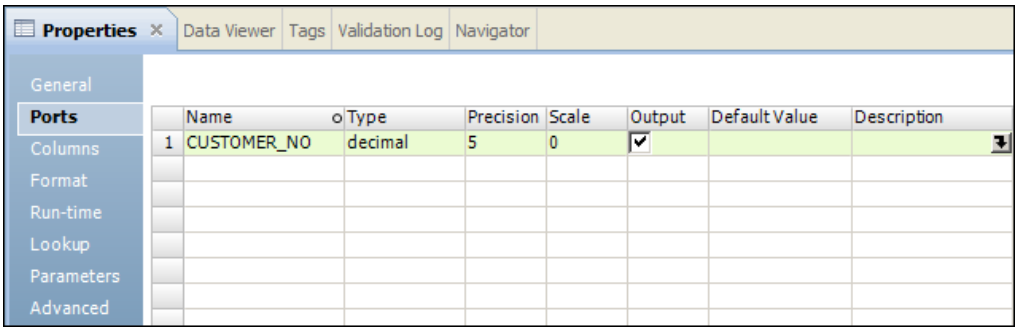
   此时，查找转换将显示在编辑器中。

8. 在编辑器中选择查找转换。

   工具栏将显示在该转换上方。

9. 在**属性**视图的**端口**选项卡上，向转换添加输出端口。

   下图显示了查找转换中的 CUSTOMER_NO 输出端口：



10. 在**属性**视图的**运行时**选项卡上，选择**启用查找缓存**以启用查找缓存。

    注意: 必须现在映射中添加查找转换，然后才能配置运行时查找属性。

11. 在**属性**视图的**查找**选项卡上，添加一个或多个查找条件。
12. 在**属性**视图的**高级**选项卡上，配置跟踪级别、动态查找缓存属性和运行时连接。
13. 保存转换。

创建未连接的查找转换

要通过表达式执行查找时，请创建未连接的查找转换。可以在平面文件、引用表或关系数据对象中创建一个可重用或不可重用的未连接的查找转换。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。
3. 浏览到“查找”向导。
4. 选择**平面文件数据对象查找**, **引用表查找**或**关系数据对象查找**。
5. 单击**下一步**。

   此时将显示**新建查找**对话框。

6. 在 Developer 工具中选择物理数据对象或引用表。
7. 输入转换的名称。
8. 在**多项匹配时**下拉列表中，选择在查找转换找到多个与查找条件匹配的行时要使用的策略。

   注意: 不要为未连接的查找选择**全部返回**。

9. 单击**完成**。

   此时，查找转换将显示在编辑器中。

10. 在**概览**视图的**端口**部分，向转换添加端口。

    为 :LKP 表达式中的每个参数创建一个输入端口。为所创建的每个查找条件创建一个输入端口。可以在多个条件中使用一个输入端口。

11. 在**概览**视图的**端口**部分，将一个端口配置为返回端口。

12. 在**查找**视图中，添加一个或多个查找条件以将转换输入值与查找源或缓存中的值进行比较。

    当条件为 true 时，查找会在返回端口中返回一个值。如果查找条件为 false，查找将返回 NULL。

13. 为允许使用表达式的转换（例如汇总器转换、表达式转换或更新策略转换）中的端口创建 :LKP 表达式。
14. 创建映射时，在编辑器中将未连接的查找转换添加至映射，但不要将端口连接到映射中的其他转换。

未连接的查找示例

加利福尼亚的一个零售店在向该州客户所售商品的每个定价中都添加了州营业税。税额取决于客户所在的县。要检索营业税，请创建一个查找转换以检索县名称并返回对应于该县的营业税额。如果该县不收取营业税，则查找转换将返回空值。在表达式转换中调用查找。

通过完成以下步骤，可配置对营业税的未连接查找（按县）：

1. 导入包含各县营业税额的平面文件物理数据对象。
2. 创建未连接的查找转换。
3. 向查找转换添加输入端口。
4. 定义返回端口。
5. 创建查找条件。
6. 在表达式转换中调用查找。

步骤 1。将营业税查找源导入模型存储库中。

创建查找转换之前，营业税文件必须位于模型存储库中。在此方案中，营业税文件包含两个字段：Sales_County 和 County_SalesTax。Sales_County 是一个包含县名称的字符串。County_SalesTax 是一个包含县税率的小数字段。营业税文件为查找源。

步骤 2。创建未连接的查找转换

使用营业税平面文件数据对象创建可重用的平面文件查找转换。在此方案中，该转换的名称为 Sales_Tax_Lookup。存在多项匹配时，请选择**返回第一行**。

步骤 3。定义查找转换端口

在**属性**视图的**端口**选项卡中定义查找转换端口。

端口类型	名称	类型	长度	小数位数
输入	In_County	String	25	
输出	SalesTax	Decimal	3	3

步骤 4。配置查找转换返回端口

返回端口是平面文件中查找所检索的字段。在**列**选项卡中，County_SalesTax 列为返回端口。

如果查找为 true，则集成服务会在平面文件源中找到县。集成服务将在返回端口中返回营业税值。如果集成服务未找到县，则查找结果为 false，集成服务将在返回端口中返回空值。

步骤 5。定义查找条件

在**查找**视图中定义查找条件，以便将输入值与查找源中的值进行比较。

要添加查找条件，请单击**查找列**。

查找条件使用以下语法：

```
SALES_COUNTY = IN_COUNTY
```

步骤 6。创建表达式转换

创建用于从平面文件检索销售记录的表达式转换。该表达式转换将检索客户编号、销售额以及销售所属的县，并返回客户编号、销售额和营业税。

该表达式转换具有以下端口：

端口类型	名称	类型	长度	精度	默认值
输入	County	String	25	10	
传递	Customer	String	10		
传递	SalesAmt	Decimal	10	2	
输出	SalesTax	Decimal	10	2	0

SalesTax 端口包含一个 :LKP 表达式。该表达式调用 Sales_Tax_Lookup 转换并将县名称作为参数进行传递。Sales_Tax_Lookup 转换将营业税率返回给该表达式。表达式转换将此税率与销售额相乘。

对于 SalesTax 端口，输入以下表达式：

```
(:LKP.Sales_Tax_Lookup(County) * SalesAmt)
```

SalesTax 端口包含表达式结果。如果查找失败，则查找转换会返回空值，因此 SalesTax 端口将包含空值。

您可以添加一个表达式来检查 SalesTax 端口中是否存在空值。如果 SalesTax 为空值，则可以配置 SalesTax 端口以返回零。将以下文本添加到查找表达式中以检查是否存在空值，如果存在，则返回零：

```
IIF(ISNULL(:LKP.Sales_Tax_Lookup(County) * SalesAmt),0, SalesTax)
```

非本地环境中的查找转换

非本地环境中的查找转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的查找转换

Blaze 引擎的某些处理规则与数据集成服务的处理规则不同。

多项匹配

查找转换会根据您在转换中配置的条件查找值。选择如何处理查找源中的多项匹配。

返回第一行；返回最后一行

转换返回第一个匹配结果或最后一个匹配结果。

数据集成服务对结果进行排序，标识出第一行和最后一行。以下规则确定了结果顺序：

- 排序取决于查找条件和查找输出端口中的查找端口。
- 数值排序按升序排列。
- 字符串值排序按字典顺序排列。

- 日期值排序按最早日期在前的顺序排列。
- 如果映射包含非等值连接 (<=,>=,<,>,! =) 比较，并且某些行包含 NULL 值，结果可能会因运行时引擎而异：
  - 在 Spark 引擎上，NULL 结果被视为 TRUE。有关 Spark 如何处理空值的更多信息，请参阅 Apache 文档：https://spark.apache.org/docs/3.0.0-preview/sql-ref-null-semantics.html#comp-operators
  - 在 Blaze 引擎或本地引擎上，NULL 结果被视为 FALSE。

返回任意行

转换返回与查找条件匹配的任意行。该转换会根据键端口而不是所有查找转换端口创建索引。如果您选择此选项，可以提高性能，因为编制行索引的过程更简单。

返回所有行

查找转换返回与查找条件匹配的所有行。

报告错误

查找转换使用静态缓存或没有缓存时，数据集成服务会将行标记为错误。默认情况下，查找转换将该行写入会话日志并将错误计数加一。

如果查找转换具有动态缓存，数据集成服务会在遇到多个匹配项时将会话处理为失败。数据集成服务缓存查找表或查找重复的键值时，会话会失败。

此外，如果将查找转换配置为在更新时输出旧值，查找转换会在遇到多个匹配项时返回错误。该转换会根据键端口而不是所有查找转换端口创建索引。

规则和准则

在下列情况下，映射验证会失败：

- 区分大小写处于禁用状态。
- 查找条件包含二进制数据类型。
- 查找条件使用具有复杂数据类型的字段。
- 缓存被配置为共享、命名、持久、动态或未缓存。缓存必须为静态缓存。

如果在映射中添加使用 Sqoop 作为查找转换的数据对象，数据集成服务则不通过 Sqoop 运行映射，而是通过 JDBC 运行映射。

Spark 引擎上的查找转换

Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

多项匹配

查找转换会根据您在转换中配置的条件查找值。选择如何处理查找源中的多项匹配。

返回第一行；返回最后一行

转换返回第一个匹配结果或最后一个匹配结果。

数据集成服务对结果进行排序，标识出第一行和最后一行。以下规则确定了结果顺序：

- 排序取决于查找条件和查找输出端口中的查找端口。
- 数值排序按升序排列。
- 字符串值排序按字典顺序排列。
- 日期值排序按最早日期在前的顺序排列。

- 如果映射包含非等值连接 (<=,>=,<,>,! =) 比较，并且某些行包含 NULL 值，结果可能会因运行时引擎而异：
  - 在 Spark 引擎上，NULL 结果被视为 TRUE。有关 Spark 如何处理空值的更多信息，请参阅 Apache 文档：https://spark.apache.org/docs/3.0.0-preview/sql-ref-null-semantics.html#comp-operators
  - 在 Blaze 引擎或本地引擎上，NULL 结果被视为 FALSE。

返回任意行

转换返回与查找条件匹配的任意行。该转换会根据键端口而不是所有查找转换端口创建索引。如果您选择此选项，可以提高性能，因为编制行索引的过程更简单。

返回所有行

查找转换返回与查找条件匹配的所有行。

报告错误

查找转换使用静态缓存或没有缓存时，数据集成服务会将行标记为错误。默认情况下，查找转换将该行写入会话日志并将错误计数加一。

如果查找转换具有动态缓存，数据集成服务会在遇到多个匹配项时将会话处理为失败。数据集成服务缓存查找表或查找重复的键值时，会话会失败。

此外，如果将查找转换配置为在更新时输出旧值，查找转换会在遇到多个匹配项时返回错误。该转换会根据键端口而不是所有查找转换端口创建索引。

规则和准则

在下列情况下，映射验证会失败：

- 区分大小写处于禁用状态。
- 查找条件包含二进制数据类型。
- 查找条件使用具有复杂数据类型的字段。
- 缓存被配置为共享、命名、持久、动态或未缓存。缓存必须为静态缓存。

在下列情况下，映射会失败：

- 转换未连接但与联接器或 Java 转换一起使用。

注意: 如果 HBase 查找未找到匹配，则会生成一个所有列均为空值的行。可以在查找转换后添加一个筛选器转换，以筛选出空行。

Databricks Spark 引擎上的查找转换

Databricks Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

多项匹配

当您选择返回多项匹配的第一个、最后一个或任何值时，查找转换会返回任何值。

如果将转换配置为在多项匹配时报告错误，Databricks Spark 引擎则会丢弃重复行并且不将这些行加入日志中。

注意: 如果 HBase 查找未找到匹配，则会生成一个所有列均为空值的行。可以在查找转换后添加一个筛选器转换，以筛选出空行。

映射验证

在下列情况下，映射验证会失败：

- 区分大小写处于禁用状态。
- 查找条件包含二进制数据类型。

- 查找条件使用具有复杂数据类型的字段。
- 缓存被配置为共享、命名、持久、动态或未缓存。缓存必须为静态缓存。
- 查找源不是 Microsoft Azure SQL 数据仓库。

流映射中的查找转换

流映射具有不适用于批处理映射的其他处理规则。

映射验证

在下列情况下，映射验证会失败：

- 查找是数据对象。
- 汇总器转换与配置了不相等查找条件的被动查找转换处于同一流管道中。
- 等级转换与配置了不相等查找条件的被动查找转换处于同一流管道中。
- 管道中包含配置了不相等条件的多个被动查找转换。

在下列情况下，映射会失败：

- 未连接转换。

一般准则

请考虑以下一般准则：

- 使用浮点数据类型查找数据会返回异常结果。
- 使用查找转换在平面文件、HDFS、Hive、关系、JDBC V2 和 HBase 数据中查找数据。
- 为避免 DataFrame 的交叉联接，请将查找转换配置为忽略匹配的空值。

HBase 查找

要在未缓存的 HBase 表上使用查找转换，请执行以下步骤：

1. 创建一个 HBase 数据对象。将 HBase 表添加为 HBase 数据对象的资源时，包括“行 ID”列。
2. 创建 HBase 读取数据操作，并将其导入流映射。
3. 将数据操作导入映射时，选择**查找**选项。
4. 在“查找”选项卡上，配置以下选项：
   - 查找列。在“行 ID”上指定等式条件
   - 运算符。指定 =
5. 验证 HBase 表中日期值的格式是为有效 Java 日期格式。在数据对象读取操作的**高级属性**选项卡的**日期时间格式**属性中指定此格式。

注意: 如果 HBase 查找未找到匹配，则会生成一个所有列均为空值的行。可以在查找转换后添加一个筛选器转换，以筛选出空行。

在下列情况下，映射验证会失败：

- 条件中未包含“行 ID”。
- 转换中包含不相等条件。
- 转换中包含多个条件。
- 输入列为日期类型。

JDBC V2 查找

可以在流映射中添加 JDBC V2 数据对象读取操作作为查找。可以在 Microsoft Azure 云服务的 Azure Databricks 服务中使用 JDBC V2 查找运行流映射。

第 26 章

查找缓存

本章包括以下主题：

查找缓存概览

可以将查找转换配置为缓存关系查找源或平面文件查找源。 为大查找表或文件启用查找缓存可以提高查找性能。

集成服务处理已缓存的查找转换中的第一行数据时，将在内存中构建缓存。 集成服务在源行输入到查找转换时创建缓存。 该服务根据您在转换中配置的内存量为缓存分配内存。 集成服务将条件值存储在索引缓存中，将输出值存储在数据缓存中。 集成服务在缓存中查询输入到转换中的每一行。

如果内存缓存装不下数据，集成服务会将溢出值存储在缓存文件中。 集成服务在缓存目录中创建缓存文件。 默认情况下，集成服务在 CacheDir 系统参数所指定的目录中创建缓存文件。 除非将查找转换配置为使用持久性缓存，否则在映射完成后，集成服务会释放缓存内存，并删除缓存文件。

如果使用平面文件查找，集成服务会对查找源进行缓存。 如果为已排序输入配置平面文件查找，则当条件列未分组时，集成服务将无法缓存该查找。 如果列已分组但未排序，集成服务会像未配置已排序输入一样处理查找。

如果不对查找转换进行配置以实现缓存，集成服务将在查找源中查询每个输入行。 无论是否缓存查找源，查找查询的结果和处理方式都相同。 但如果启用查找缓存，可以提高大查找源的查找性能。

查找缓存类型

可以配置不同类型的查找缓存。例如，如果要在同一个映射的多个查找转换间共享缓存，可以配置共享缓存。

可以配置以下类型的查找缓存：

静态缓存

集成服务处理查找时，静态缓存不发生更改。集成服务在每次处理查找时都重建静态缓存。默认情况下，如果为查找转换启用缓存，集成服务将创建静态缓存。集成服务在处理第一个查找请求时构建缓存。其在缓存中为进入查找转换的每一行查找值。如果查找条件为 true，集成服务将从查找缓存返回一个值。

使用静态缓存有以下原因：

- 映射运行时查找源不更改。
- 查找是未连接的查找。必须对未连接的查找使用静态缓存。
- 希望提高性能。由于集成服务在处理查找转换时不更新缓存，因此集成服务处理使用静态缓存的查找转换的速度比处理使用动态缓存的查找转换的速度更快。
- 您希望在查找条件为 false 时，集成服务为连接的转换返回默认值，或者为未连接的转换返回 NULL。

持久性缓存

集成服务处理每次查找时，持久性缓存都不发生更改。集成服务保存查找缓存文件，并在下次处理配置为使用缓存的查找转换时重新使用这些文件。查找源未发生更改时使用持久性缓存。

必要时可以将查找转换配置为重建持久性查找缓存。

动态缓存

当集成服务处理查找时，动态查找缓存会发生更改。集成服务在处理第一个查找请求时构建动态查找缓存。集成服务在处理每一行时，动态插入或更新查找缓存中的数据，并将数据传递到目标。动态缓存与目标同步。

如果要根据新记录和更改的记录更新目标，可使用动态缓存。在映射需要对目标数据进行查找但与目标之间的连接速度低下时，也可以使用动态缓存。

共享缓存

共享缓存可以由同一个映射中的多个查找转换使用。使用共享缓存可以提高映射性能。集成服务不是为每个查找转换都生成一个单独的查找缓存，而是只生成一个缓存。

未缓存查找

未缓存的查找是指集成服务未缓存查找源时的查找。默认情况下，集成服务不对查找转换使用查找缓存。

集成服务处理未缓存查找的方式与处理缓存查找的方式相同，但其查询查找源，而不是构建和查询查找缓存。

如果查找条件为 true，集成服务将从查找源返回值。当集成服务处理已连接的查找转换时，它将返回由查找/输出端口表示的值。当它处理未连接的查找转换时，集成服务将返回由返回端口表示的值。

如果条件不为 true，集成服务将返回 NULL 或默认值。当集成服务处理已连接的查找转换时，如果不满足条件，它将返回输出端口的默认值。当它处理未连接的查找转换时，如果不满足条件，集成服务将返回空。

静态查找缓存

静态查找缓存指集成服务在处理查找转换时不更新的缓存。默认情况下，集成服务在您为实现缓存而配置查找转换时创建静态查找缓存。

集成服务在处理第一个查找请求时构建缓存。该服务根据传递到转换中的每一行的查找条件查询缓存。

如果查找条件为 true，集成服务将从静态查找缓存返回值。当集成服务处理已连接的查找转换时，它将返回由查找/输出端口表示的值。当集成服务处理未连接的查找转换时，它将返回由返回端口表示的值。

如果条件不为 true，集成服务将返回 NULL 或默认值。当集成服务处理已连接的查找转换时，如果不满足条件，集成服务将返回输出端口的默认值。当集成服务处理未连接的查找转换时，如果不满足条件，它将返回空值。

持久性查找缓存

持久性查找缓存指集成服务为多次运行同一映射而重用的缓存。如果在映射运行期间查找源不更改，则使用持久性查找缓存。

默认情况下，如果在查找转换中启用查找缓存，集成服务将使用非持久性缓存。映射完成时，集成服务将删除缓存文件。下次运行映射时，集成服务从查找源构建内存缓存。

如果将查找转换配置为使用持久性查找缓存，集成服务将保存缓存文件并在多次运行映射时重复使用缓存文件。使用持久性缓存可以消除读取查找表和重建查找缓存所需的时间。

集成服务在首次使用持久性查找缓存运行映射时，会将缓存文件保存到磁盘。下次运行映射时，集成服务将通过这些缓存文件构建内存缓存。

如果原始查找源改变，可以配置集成服务来重建持久性查找缓存。重建缓存时，集成服务创建新的缓存文件，并在集成服务日志中写入一条消息。

重建持久性查找缓存

可以配置集成服务来重建持久性查找缓存。在某些情况下，集成服务将重建持久性查找缓存，即使您未对其进行配置亦不例外。

重建持久性查找缓存时，应考虑以下规则和准则：

- 如果集成服务在最后一次构建缓存后查找源发生了更改，可以重建持久性查找缓存。
- 如果映射包含一个或多个共享一个缓存的查找转换，则可以重建缓存。
- 如果查找表在映射运行之间未更改，则可以将查找转换配置为使用持久性查找缓存。集成服务将保存并重用缓存文件，以消除读取查找表所需的时间。
- 如果将后续查找转换配置为重建查找缓存，则当集成服务处理后续查找转换时，将共享缓存而不会重建缓存。
- 如果映射包含两个持久性查找，并且您将第二个查找转换配置为重建缓存，则集成服务将为这两个查找重建持久性查找缓存。

在以下情况下，集成服务将重建持久性查找缓存：

- 集成服务找不到缓存文件。
- 集成服务无法重用缓存。这种情况下，要么重建查找缓存，要么映射将失败。
- 为集成服务启用或禁用高精度。
- 编辑查找转换或映射。

  注意: 如果编辑转换说明，集成服务将不重建缓存。
- 更改分区数量。
- 您更改了用于访问查找源的数据库连接或文件位置。
- 在 Unicode 模式下更改了排序顺序。
- 更改了集成服务代码页。

动态查找缓存

动态缓存指集成服务在处理每行时更新的缓存。利用动态查找缓存可以使缓存与目标保持同步。

可以将动态缓存与关系查找和平面文件查找共用。集成服务在处理第一个查找请求时构建缓存。其根据传递到查找转换中的每一行的查找条件查询缓存。集成服务在处理每一行时更新查找缓存。

根据查找查询的结果、行类型和查找转换属性，集成服务在缓存中插入或更新行，或者不对缓存进行更改。

共享查找缓存

共享查找缓存指映射中多个查找转换共享的静态查找缓存。使用共享查找缓存可以减少构建缓存所需的时间。

默认情况下，对于映射中缓存结构兼容的查找转换，集成服务会共享缓存。例如，如果在一个映射中包含相同的可重用查找转换的两个实例，并且对这两个实例使用相同的输出端口，则默认情况下查找转换将共享查找缓存。

集成服务在处理第一个查找转换时构建缓存。其使用同一个缓存处理共享该缓存的后续查找转换。集成服务共享查找缓存时，其在集成服务日志中写入一条消息。

集成服务为第一个查找转换分配数据缓存内存和索引缓存内存。其不为共享查找缓存的后续查找转换分配额外内存。

如果转换或缓存结构不允许共享，集成服务将创建新的缓存。

共享查找缓存的规则和准则

共享查找缓存时，应考虑以下规则和准则：

- 可以与动态查找共享一个或多个静态缓存。如果动态查找与同一个映射中的静态查找共享缓存，静态查找将重用动态查找所创建的缓存。
- 无法在动态查找间共享缓存。
- 如果将多个查找转换配置为重建持久性查找缓存，集成服务将为第一个查找转换构建缓存，然后对后续查找转换共享持久性查找缓存。
- 如果不将第一个查找转换配置为重建持久性查找缓存，而将后续查找转换配置为重建缓存，则这些转换无法共享缓存。集成服务在处理每个查找转换时构建缓存。
- 后续查找转换的查找/输出端口必须匹配，或者是集成服务用来构建缓存的查找转换中的端口的子集。端口的顺序无需匹配。
- 共享缓存的查找转换必须具有以下特征：
  - 查找转换必须在查找条件中使用相同的端口。
  - 如果使用了 SQL 替代，则这些查找转换必须使用相同的 SQL 替代。
  - 必须在所有查找转换中启用查找缓存。
  - 查找转换必须使用相同类型的查找源。
  - 所有关系查找转换必须使用相同的数据库连接。
  - 查找转换必须使用相同的查找表名称。
  - 所有查找转换的缓存结构必须兼容。

缓存比较

集成服务根据所配置的查找缓存的类型执行不同的操作。

下表将查找转换与未缓存查找、静态缓存和动态缓存进行了比较：

未缓存	静态缓存	动态缓存
集成服务不在缓存中插入或更新缓存。	集成服务不在缓存中插入或更新缓存。	集成服务可以在向目标传递行时在缓存中插入或更新行。
可以使用关系查找。	可以使用关系查找或平面文件查找。	可以使用关系查找或平面文件查找。
如果条件为 true，集成服务将从查找表或缓存返回一个值。 如果条件不为 true，集成服务将为连接的转换返回默认值，为未连接的转换返回空值。	如果条件为 true，集成服务将从查找表或缓存返回一个值。 如果条件不为 true，集成服务将为连接的转换返回默认值，为未连接的转换返回空值。	如果条件为 true，集成服务将更新缓存中的行，或者保持缓存不变，具体视行类型而定。这表明行在缓存和目标表中。可以将更新的行传递到目标。 如果条件不为 true，集成服务会在缓存中插入行，或者保持缓存不变，具体视行类型而定。这表明行不在缓存或目标中。可以将插入的行传递到目标表。

查找的缓存分区

缓存分区为每个分区创建单独的缓存来处理汇总器、联接器、等级或查找转换。通过缓存分区功能，每个分区并行查询单独的缓存，从而提高了映射应能。

集成服务创建映射的分区时，可对已分区的查找转换使用缓存分区。

当以下条件为真时，集成服务将对已连接的查找转换使用缓存分区：

- 查找条件仅包含等式运算符。
- 已连接的查找转换查找关系表中的数据时，数据库将配置为区分大小写的比较。

集成服务不对未连接的查找转换使用缓存分区。

如果集成服务不对查找转换使用缓存分区，则该查找转换的所有分区将共享同一个缓存。每个分区将连续查询同一缓存。

第 27 章

动态查找缓存

本章包括以下主题：

动态查找缓存概览

利用动态查找缓存可以使缓存与目标保持同步。可以将动态缓存与关系查找或平面文件查找共用。

集成服务在处理第一个查找请求时构建动态查找缓存。该服务根据传递到转换中的每一行的查找条件查询缓存。集成服务在处理每一行时更新查找缓存。

集成服务根据查找查询的结果、行类型以及查找转换属性，在从源读取行时对动态查找缓存执行以下操作之一：

在缓存中插入行

如果行不在缓存中，并且已将查找转换配置为在缓存中插入行，则集成服务会插入行。可以将转换配置为根据输入端口或生成的序列 ID 在缓存中插入行。集成服务将行标记为插入。

更新缓存中的行

如果行存在于缓存中，并且已将查找转换配置为更新缓存中的行，则集成服务会更新行。集成服务根据输入端口更新缓存中的行。集成服务将行标记为更新行。

不对缓存进行更改

在以下情况下，集成服务不会对缓存进行更改：如果行存在于缓存中，并且已将查找转换配置为仅插入新行。或者，行不存在于缓存中并且指定了仅更新现有行。或者，行存在于缓存中，但根据查找条件，未进行任何更改。集成服务将行标记为未更改。

根据 NewLookupRow 的值，还可以配置路由器或筛选器转换的动态查找转换，以将插入行或更新行路由到目标表。可以将未更改的行路由到其他目标表或平面文件，也可以删除这些行。

动态查找缓存的使用

可以使用动态查找缓存配置查找转换，以根据查找源中的更改更新该缓存。

您可能会因以下原因使用动态查找缓存：

将新的和更新后的客户信息更新到主客户表中。

例如，您可以使用查找转换对客户表执行查找，以确定目标中是否存在客户。缓存代表客户表。将行传递到目标时，查找转换会在缓存中插入并更新行。

使用导出的平面文件作为查找源来代替关系表。

如果到数据库的连接非常慢，您可以将关系表内容导出到平面文件中，并使用该文件作为查找源。例如，如果到数据库的 ODBC 连接非常慢，您可能需要使用此方法。可以在映射中配置数据库表作为关系目标，然后将查找缓存更改传递回数据库表。

动态查找缓存属性

配置动态查找属性以启用动态查找缓存，并配置更新缓存的方式。 例如，可以配置要在动态缓存中插入和更新的值。

启用动态查找缓存后，配置以下属性：

多项匹配时

设置为报表错误。

动态查找缓存

启用动态查找缓存。

该选择在启用查找缓存后可用。

更新 Else 插入

适用于输入查找转换的行类型为“更新”的行。 如果启用该项，集成服务将更新缓存中的现有行，如果是新行则插入行。 如果禁用该项，集成服务不插入新行。

该选择在启用动态缓存后可用。

插入 Else 更新

适用于输入查找转换的行类型为“插入”的行。 如果启用该项，集成服务在缓存中插入行并更新现有行。 如果禁用该项，集成服务不更新现有行。

该选择在启用动态缓存后可用。

更新时输出旧值

查找转换可以从缓存输出现有值或新值。 如果启用该项，集成服务将在更新缓存中的值之前从查找/输出端口输出现有值。 集成服务更新缓存中的行时，其在基于输入数据更新行之前输出查找缓存中的值。 当集成服务在缓存中插入行时，将输出空值。

禁用该属性以使集成服务从查找/输出端口和输入/输出端口传递相同的值。 默认情况下该属性处于启用状态。

该选择在启用动态缓存后可用。

更新动态缓存条件

如果启用该项，集成服务将使用条件表达式确定是否更新动态缓存。 如果条件为 true 并且数据存在于缓存中，集成服务将更新缓存。

使用查找端口或输入端口创建表达式。 表达式可以包含输入值或查找缓存中的值。 默认值为 true。

该选择在启用动态缓存后可用。

NewLookupRow

开发程序工具将此端口添加到配置了动态缓存的查找转换中。

NewLookupRow 属性可以包含以下值之一：

- 0 = 不对缓存进行更新。
- 1 = 在缓存中插入行。
- 2 = 更新缓存中的行。

为使查找缓存与目标表保持同步，可在 NewLookupRow 值等于 1 或 2 时将行传递到目标。

关联端口

集成服务在更新缓存中的数据时使用关联端口的值。 集成服务关联在查找条件中指定的输入端口和查找源端口。 必须为动态查找中的其余查找源端口配置关联端口。 如果不为动态查找中的所有查找源端口配置关联端口，映射验证将失败。

可以将查找源端口与以下对象关联：

对象	说明
输入端口	根据输入端口的值更新缓存。
关联表达式	选择以输入表达式。集成服务根据表达式的结果更新缓存。
序列 ID	为查找缓存中插入的行生成主键。只能将序列 ID 与长整型和整型列关联。

忽略更新的空值输入

将查找转换配置为使用动态缓存时，开发程序工具会为查找/输出端口激活此端口属性。 如果不希望集成服务将缓存中的列更新为空输入值，则选择该属性。

在比较中忽略

将查找转换配置为使用动态缓存时，开发程序工具会为查找条件中未使用的查找/输出端口激活此端口属性。默认情况下，集成服务将所有查找端口中的值与其关联端口中的值相比较。 如果希望集成服务在更新行之前比较值时忽略端口，则选择该属性。 使用该属性可提高比较的性能。

动态查找缓存和输出值

如果启用动态查找缓存，输出端口值将因动态查找缓存的配置方式而异。 查找/输出端口的输出值取决于在集成服务更新行时选择输出旧值还是输出新值。

可以配置**更新时输出旧值**属性，为查找/输出端口指定以下类型的输出值之一：

- 更新时输出旧值。 集成服务在更新行之前输出缓存中的现有值。
- 更新时输出新值。 集成服务输出写入到缓存中的已更新的值。 查找/输出端口值与输出端口值匹配。

查找转换值

查找转换包含输入端口、查找和输出端口的值。 如果启用动态查找缓存，输出端口值将因动态查找缓存的配置方式而异。

查找转换包含以下类型的值：

输入值

集成服务传递到查找转换的值。

查找值

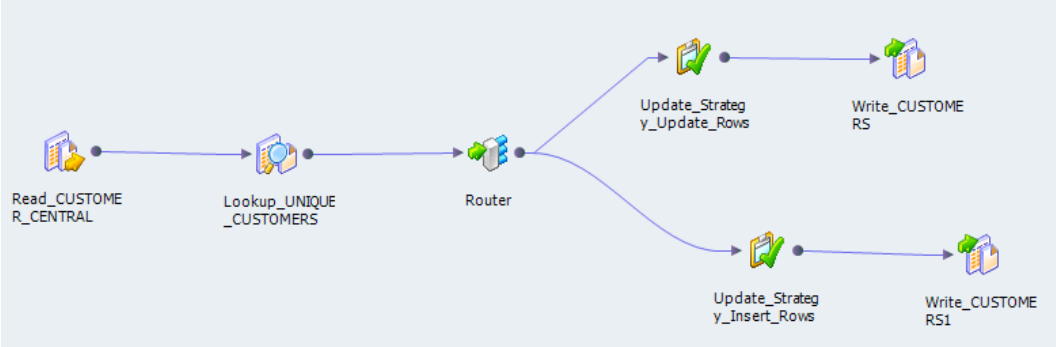
集成服务插入到缓存中的值。

输出值

集成服务从查找转换的输出端口传递的值。 查找/输出端口的输出值取决于在集成服务更新行时选择输出旧值还是输出新值。

查找转换值示例

例如，创建一个含有以下对象的映射：



对于查找转换，可启用动态查找缓存并定义以下查找条件：

```
IN_CUST_ID = CUST_ID
```

默认情况下，所有输入到查找转换中的行的类型均为“插入”。 要在缓冲和目标表中同时执行插入和更新，可以在查找转换中选择**插入 Else 更新**属性。

初始缓存值

运行映射时，集成服务将从目标表构建查找缓存。

下表显示了查找缓存的初始值：

PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
100001	80001	Marion James	100 Main St.
100002	80002	Laura Jones	510 Broadway Ave.
100003	80003	Shelley Lau	220 Burnside Ave.

输入值

源包含目标表中存在的行和不存在的行。 集成服务将源行传递到查找转换。

下表显示了源行：

SQ_CUST_ID	SQ_CUST_NAME	SQ_ADDRESS
80001	Marion Atkins	100 Main St.
80002	Laura Gomez	510 Broadway Ave.
99001	Jon Freeman	555 6th Ave.

查找值

集成服务根据查找条件在缓存中查找值。 该服务更新缓存中 ID 为 80001 和 80002 的现有客户的行。 对于客户 ID 99001，该服务在缓存中插入一个行。 集成服务为新行生成新键 (PK_PRIMARYKEY)。

下表显示了从查找返回的行和值：

PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
100001	80001	Marion Atkins	100 Main St.
100002	80002	Laura Gomez	510 Broadway Ave.
100004	99001	Jon Freeman	555 6th Ave.

输出值

集成服务根据其对动态缓存执行的插入和更新来标记查找转换中的行。 最后，集成服务将行传递到路由器转换，该转换为插入行创建一个分支，为更新行创建另一个分支。 每个分支均包含一个更新策略转换。 更新策略转换根据 NewLookupRow 端口的值标记要插入或更新的行。

查找/输出端口和输入/输出端口的输出值取决于在集成服务更新行时选择输出旧值还是输出新值。 但是，新行和已更新行的 NewLookupRow 端口及使用序列 ID 的任何查找/输出端口的输出值相同。

如果选择输出新值，查找/输出端口将输出以下值：

NewLookupRow	PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
2	100001	80001	Marion Atkins	100 Main St.
2	100002	80002	Laura Gomez	510 Broadway Ave.
1	100004	99001	Jon Freeman	555 6th Ave.

如果选择输出旧值，查找/输出端口将输出以下值：

NewLookupRow	PK_PRIMARYKEY	CUST_ID	CUST_NAME	ADDRESS
2	100001	80001	Marion James	100 Main St.
2	100002	80002	Laura Jones	510 Broadway Ave.
1	100004	99001	Jon Freeman	555 6th Ave.

集成服务更新查找缓存中的行时，对缓存和目标表中的行使用主键 (PK_PRIMARYKEY) 值。

对于未在缓存中找到的客户，集成服务使用序列 ID 生成主键。 集成服务将主键值插入到查找缓存中，并将该值返回到查找/输出端口。

集成服务从输入/输出端口输出与输入值匹配的值。

注意: 如果输入值为 NULL，并对关联的输入端口选择“忽略空值”属性，则输入值不等于查找值或来自输入/输出端口的值。 选择“忽略空值”属性时，如果将空值传递给目标，则查找缓存和目标表可能不同步。 必须确认未将空值传递给目标。

SQL 替代和动态查找缓存

可以在查找查询中添加 WHERE 子句，以筛选用来构建缓存的记录，并在数据库表中对未缓存的查找执行一次查找。 但是，集成服务在动态缓存中插入行时不使用 WHERE 子句。

使用动态缓存在查找转换中添加 WHERE 子句时，在查找转换之前连接筛选器转换以筛选您不想插入到缓存或目标表中的行。 如果您不包括筛选器转换，可能会在缓存和目标表之间获得不一致的结果。

例如，配置查找转换以按 EMP_ID 在员工表、EMP、匹配行上执行动态查找。 定义以下查找 SQL 替代：

```
SELECT EMP_ID, EMP_STATUS FROM EMP ORDER BY EMP_ID, EMP_STATUS WHERE EMP_STATUS = 4
```

首次运行映射时，集成服务基于查找 SQL 替代从目标表构建查找缓存。 缓存中的所有行均匹配 WHERE 子句中的条件，`EMP_STATUS = 4`。

例如，集成服务读取符合您指定的查找条件的源行，但是 EMP_STATUS 的值为 2。 虽然目标可能包含 EMP_STATUS 为 2 的行，但是集成服务因 SQL 替代仍未在缓存中找到该行。 集成服务将行插入缓存并将行传递到目标表。 当集成服务将此行插入目标表时，如果该行已存在则可能会获得不一致的结果。 此外，并不是缓存中的所有行都匹配 SQL 替代中 WHERE 子句中的条件。

要验证您是否仅将行插入匹配 WHERE 子句的缓存，请在查找转换之前添加筛选器转换并将筛选器条件定义为查找 SQL 替代中 WHERE 子句中的条件。

对于上述示例，在筛选器转换中输入以下筛选器条件，在 SQL 替代中输入 WHERE 子句。

```
EMP_STATUS = 4
```

动态查找缓存的映射配置

如果将查找与动态缓存结合使用，必须将映射配置为更新动态查找缓存并将更改后的行写入目标中。

完成以下步骤为映射配置动态查找缓存：

将查找转换的输入行标记为插入或更新。

默认情况下，所有输入行的行类型均为“插入”。 在查找转换之前添加一个更新策略转换，以便为输入行指定不同的行类型。

指定集成服务处理动态缓存的输入行的方式。

选择**插入 Else 更新**或**更新 Else 插入**选项，以处理标记为插入或更新的行。

为要插入到目标并在目标中更新的行创建单独的映射管道。

在查找转换的后面添加一个筛选器转换或路由器转换，以将插入行和更新行路由到单独的映射分支中。 使用 NewLookupRow 的值确定每一行的相应分支。

配置查找转换的输出行的行类型。

添加更新策略转换以将行标记为插入或更新。

插入 Else 更新

使用**插入 Else 更新**属性以在插入行类型时更新动态查找缓存中的现有行。

该属性仅适用于输入查找转换且行类型为“插入”的行。 当任何其他类型的行（如“更新”）输入查找转换时，**插入 Else 更新**属性对集成服务处理行的方式没有影响。

如果选择**插入 Else 更新**并且输入查找转换的行类型为插入，则集成服务会在行为新行时将其插入到缓存中。 如果行存在于索引缓存中但数据缓存不同于当前行，则集成服务会更新数据缓存中的行。

如果不选择**插入 Else 更新**并且输入查找转换的行类型为插入，则集成服务会在行为新行时将其插入到缓存中，并且如果该行已存在则不对缓存进行更改。

下表介绍了当输入到查找转换的行的类型为插入时，集成服务如何更改查找缓存：

插入 Else 更新选项	在缓存中发现行	数据缓存不同	查找缓存结果	NewLookupRow 值
已清除 - 仅插入	是	-	无更改	0
已清除 - 仅插入	否	-	插入	1
已选定	是	是	更新	2[1]
已选定	是	否	无更改	0
已选定	否	-	插入	1

1. 如果对查找条件中不存在的所有查找端口选择“忽略空值”，并且如果所有这些端口中都包含空值，集成服务将不更改缓存，并且 NewLookupRow 值等于 0。

更新 Else 插入

更新行类型时，使用**更新 Else 插入**属性可在动态查找缓存中插入新行。

可以在查找转换中选择**更新 Else 插入**属性。 此属性仅适用于进入要更新行类型的查找转换的行。 当任何其他行类型的行（例如插入）进入查找转换时，此属性对集成服务处理该行的方式不产生任何影响。

选择此属性并更新进入查找转换的行类型时，如果索引缓存中存在该行，但缓存数据与现有行不同，集成服务将更新缓存中的相应行。 如果行是新行，集成服务会将其插入缓存中。

如果在未选择此属性时更新进入查找转换的行类型，则只要该行存在，集成服务就会更新缓存中的相应行，如果该行是新行，则不对缓存进行更改。

如果对查找条件中不存在的所有查找端口选择**忽略空值**，并且如果所有这些端口中都包含空值，集成服务将不更改缓存，并且 NewLookupRow 值等于 0。

下表介绍了更新进入查找转换的行的行类型时，集成服务如何更改查找缓存：

更新 Else 插入选项	在缓存中发现行	数据缓存不同	查找缓存结果	NewLookupRow 值
已清除（仅限更新）	是	是	更新	2
已清除（仅限更新）	是	否	无更改	0
已清除（仅限更新）	否	-	无更改	0
已选定	是	是	更新	2

更新 Else 插入选项	在缓存中发现行	数据缓存不同	查找缓存结果	NewLookupRow 值
已选定	是	否	无更改	0
已选定	否	-	插入	1

动态查找缓存和目标同步

配置下游转换，以确保动态查找缓存与目标同步。

使用动态查找缓存时，集成服务在写入目标表之前先写入查找缓存。 如果集成服务不将数据写入目标，查找缓存和目标表可能会不同步。 例如，目标数据库可能会拒绝数据。

请考虑以下指导原则以保持查找缓存与查找表同步：

- NewLookupRow 值等于一或二时，请对缓存的目标使用路由器转换。
- 如果 NewLookupRow 值等于零，请使用路由器转换以删除行。 或者，将行输出至其他目标。
- 在查找转换之后使用更新策略转换以标记要插入或更新到目标的行。
- 验证查找转换输出到目标的值与集成服务写入查找缓存的值是否相同。 选择在更新上输出新值时，仅将查找/输出端口连接至目标表（而非输出端口）。 如果选择在更新时输出旧值，请在查找转换之后路由器转换之前添加一个表达式转换。 请在表达式转换中为目标表中的每个端口添加输出端口并创建表达式，以确保不会将空输入值输出到目标中。
- 定义更新策略目标表选项时选择“插入”和“更新”。 这将确保集成服务更新标记为更新的行，插入标记为插入的行。

条件动态查找缓存更新

可以根据布尔表达式的结果更新动态查找缓存。 该表达式为 true 时，集成服务更新缓存。

例如，可能在目标表中有产品编号、现存数量以及时间戳列。 需要使用最新源值更新现存数量。 当源数据的时间戳大于动态缓存中的时间戳时，可以更新现存数量。 在查找转换中创建一个类似以下表达式的表达式：

```
lookup_timestamp < input_timestamp
```

该表达式可以包含查找和输入端口。 您可以访问内置、映射和参数变量。 您可以包含用户定义的函数并引用未连接的转换。

表达式返回 true、false 或 NULL。 如果表达式的结果是 NULL，则表达式为 false。 集成服务不会更新缓存。 如果需要将表达式结果更改为 true，则可以为表达式中的 NULL 值添加复选框。 默认表达式值为 true。

条件动态查找缓存处理

可以创建用于确定集成服务是否更新动态查找缓存的条件。 如果条件为 false 或 NULL，集成服务将不更新动态查找缓存。

如果条件为 false 或 NULL，则无论查找转换属性如何，NewLookupRow 值均为 0，并且集成服务不用行插入或行更新来更新动态查找缓存。

如果 NewLookupRow 值为 2，即该行标记为更新，并且您启用“首选更新，否则插入”，则集成服务将执行以下操作之一：

- 如果缓存中存在该行，则集成服务将更新缓存中的现有行。
- 如果缓存中不存在该行，则集成服务会在缓存中插入新行。

如果 NewLookupRow 值为 1，即该行标记为插入，并且您启用“首选插入，否则更新”，则集成服务将执行以下操作之一：

- 如果缓存中不存在该行，则集成服务会在缓存中插入新行。
- 如果缓存中存在该行，则集成服务将更新缓存中的现有行。

配置条件动态查找缓存

可以配置用于确定集成服务是否更新动态查找缓存的表达式。

1. 创建查找转换。
2. 在**属性**视图的**运行时**选项卡上，选择**启用查找缓存**。
3. 在**属性**视图的**高级**选项卡上，选择**动态查找缓存**。
4. 要输入条件，请单击**更新动态缓存条件**属性的向下箭头。

   此时将显示表达式编辑器。
5. 定义表达式条件。

   可以为表达式选择输入端口、查找端口和函数。
6. 单击**验证**以验证表达式是否有效。
7. 单击**确定**。
8. 如果适用，配置其他用于动态查找缓存的高级属性。

表达式结果的动态缓存更新

查找转换可以将动态查找缓存值更新为表达式的结果。

例如，产品表目标具有一个包含顺序计数的数值列。 每次查找转换收到产品的顺序，都会通过以下表达式的结果更新动态缓存 order_count：

```
order_count = order_count + 1
```

查找转换返回 order_count。

可以配置集成服务如何处理表达式计算结果为 NULL 的情况。

空表达式值

如果表达式中的一个值为空，则表达式将返回 NULL。 但可以将表达式配置为返回非空值。

如果表达式引用查找端口，但源数据为新数据，则查找端口将包含一个默认值。 默认值可能为 NULL。 可以配置 IsNull 表达式以检查空值。

例如，以下表达式将检查 lookup_column 是否为 NULL：

```
iif (isnull(lookup_column), input_port, user_expression)
```

如果列为空，则返回 input_port 值。 否则返回表达式的值。

表达式处理

集成服务可以根据表达式结果插入和更新动态查找缓存中的行。 表达式结果可能会因查找端口值是否为 NULL 以及是否包含在表达式中而有所不同。

启用“插入 Else 更新”后，如果数据不在缓存中，集成服务会插入含有表达式结果的行。 如果数据不存在于缓存中，查找端口值为 NULL。 如果表达式引用查找端口值，集成服务将置换表达式中的默认端口值。 如果启用“插入 Else 更新”，并且数据存在于缓存中，则集成服务会用表达式结果更新缓存。

启用“更新 Else 插入”后，如果数据存在于缓存中，集成服务会用表达式结果更新缓存。 如果数据不存在于缓存中，集成服务会插入一个包含表达式结果的行。 如果表达式引用查找端口值，集成服务将置换表达式中的默认端口值。

配置动态缓存更新的表达式

可以配置动态查找缓存更新的表达式。

必须先启用查找转换以执行动态查找，然后才能创建条件表达式。

1. 创建查找转换。
2. 在**属性**视图的**运行时**选项卡上，选择**启用查找缓存**。
3. 在**属性**视图的**高级**选项卡上，选择**动态查找缓存**。
4. 如果适用，配置其他用于动态查找缓存的高级属性。
5. 要创建表达式，在**属性**视图中选择**列**选项卡。
6. 在**关联端口**列中，单击下拉箭头以找到要更新的查找端口。
7. 从下拉列表中选择**关联表达式**，然后单击**输入**。

   此时将显示表达式编辑器。
8. 定义表达式。

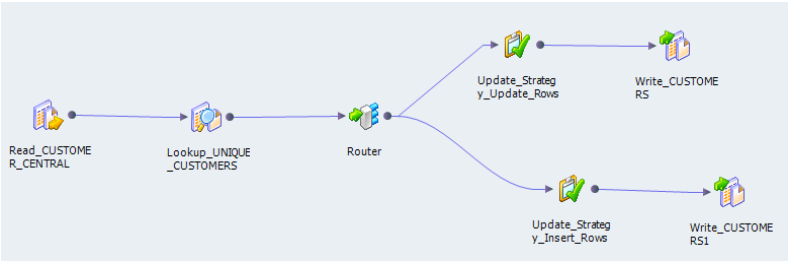
   可以为表达式选择输入端口、查找端口和函数。 表达式返回值必须与查找端口的数据类型相匹配。
9. 单击**验证**以验证表达式是否有效。
10. 单击**确定**。

动态查找缓存示例

可以使用动态查找缓存在目标中插入和更新行。 使用动态查找缓存时，可以在缓存中插入和更新在目标中插入和更新的行。

例如，需要更新包含客户数据的表。 源数据包含要在目标中插入或更新的客户数据行。 创建代表目标的动态缓存。 配置查找转换以在动态缓存中查找客户。

下图显示的映射中包含使用动态查找缓存的查找转换：



路由器转换拆分为两个分支。 路由器转换将插入行传递到一个分支中，将更新行传递到另一个分支中。 每个分支均包含将行写入目标的更新策略转换。 两个分支包含同一个目标。

映射开始时，集成服务从客户目标表构建查找缓存。 集成服务读取不在查找缓存中的行时，会在缓存中插入该行。

查找转换将每个行都返回到路由器转换。 根据行标记为插入还是更新，路由器转换会将行定向到多个更新策略转换中的一个转换。 路由器转换根据 NewLookupRow 属性来确定行标记为插入还是更新。更新策略转换将每个行标记为插入或更新，然后再将其传递到目标。

映射运行时，客户目标表会发生更改。 集成服务在查找缓存中插入新行并更新现有行。 集成服务使查找缓存和客户目标表保持同步。

要生成目标的键，可在关联端口中使用序列 ID。 集成服务将序列 ID 用作插入到目标表中的每个新行的主键。

由于从数据库构架一次缓存，因此使用动态查找缓存可以提高会话性能。

动态查找缓存的规则和准则

使用动态查找缓存时，请考虑以下准则：

- 使用动态查找缓存时，必须设置**多项匹配时**属性以报告错误。 要重置该属性，请将动态查找更改为静态查找、更改该属性，然后将静态查找更改为动态查找。
- 无法在同一目标加载顺序组中的动态查找转换与静态查找转换之间共享缓存。
- 可以为关系或平面文件查找启用动态查找缓存。
- 查找转换必须是已连接的转换。
- 可以使用持久或非持久缓存。
- 如果动态缓存不持久，则即使不启用**从查找源重新缓存**，集成服务也始终从数据库重建缓存。
- 可以仅创建一个等式查找条件。 不能在动态缓存中查找数据的范围。
- 必须将不在查找条件中的每个查找端口与输入端口、序列 ID 或关联的表达式相关联。
- NewLookupRow 值等于一或二时，请对缓存的目标使用路由器转换。
- 如果 NewLookupRow 值等于零，请使用路由器转换以删除行。 或者，可以将行输出到不同的目标。
- 验证集成服务是否将相同的值输出到写入查找缓存的目标。 如果选择在更新时输出新值，则仅将查找/输出端口（而不是输入/输出端口）连接到目标表。 如果选择在更新时输出旧值，请在查找转换之后路由器转换之前添加一个表达式转换。 请在表达式转换中为目标表中的每个端口添加输出端口并创建表达式，以确保不会将空输入值输出到目标中。
- 使用查找 SQL 替代时，请将正确的列映射到恰当的查找目标。

- 将 WHERE 子句添加到查找 SQL 替代时，请先应用筛选器转换，然后再应用查找转换。 这样可确保集成服务在与 WHERE 子句匹配的动态缓存和目标表中插入行。
- 配置可重用查找转换以使用动态缓存时，无法在映射中编辑条件或禁用**动态查找缓存**属性。
- 请先应用查找转换，然后再应用更新策略转换，以便为目标标记要插入或更新的行。
- 如果要在查找转换中使用“更新 Else 插入”属性，请先应用更新策略转换，然后再应用查找转换，以将部分或全部行定义为更新。

第 28 章

宏转换

本章包括以下主题：

宏转换概述

宏转换是一种被动转换，可为转换逻辑启用动态功能。

在动态映射中使用宏转换，可以在运行时适应对源和目标的更改。宏转换将更改传播到宏转换引用的 Mapplet 中，使 Mapplet 能够充当动态 Mapplet。宏转换始终与一个称为宏指令的 Mapplet 相关联。宏指令 Mapplet 可以包含不支持动态功能的转换。

宏转换不能配置为可重用转换。

宏指令

宏指令是宏转换引用的 Mapplet。每个宏转换都必须有一个宏指令。

在创建宏转换时，Developer tool 提示您选择一个 Mapplet 作为宏指令。在创建转换之后，无法选择其他 Mapplet 作为宏指令。

请考虑以下关于宏指令 Mapplet 的规则和准则：

- 此 Mapplet 必须是中游 Mapplet。
- 此 Mapplet 必须仅有一个输入转换。
- 此 Mapplet 不能使用主动转换。

宏转换参数

在向映射添加宏转换时，可以使用宏指令 Mapplet 中的参数作为映射参数，就像向映射添加 Mapplet 时一样。映射参数表示可以在映射运行之间更改的常量值。

添加带有 Mapplet 参数的 Mapplet 作为宏指令时，可以设置参数的实例值。Mapplet 参数的实例值是指特定映射的参数值。从宏转换的**属性**视图中的**参数**选项卡设置实例值。

有关详细信息，请参阅*《Informatica® Developer 映射指南》*中的“映射中的 Mapplet 参数”。

动态映射中的宏转换

可以向动态映射添加宏转换。转换包含可链接到动态源和目标的动态端口。

当您创建宏转换并将宏指令 Mapplet 分配给转换时，将根据宏指令的输入和输出组创建动态输入和输出端口。

默认情况下，输出端口与输入端口具有相同的类型、精度、小数位数和说明。您可以通过编辑**端口**属性中与每个输出端口关联的基本端口来配置输出端口。

宏转换中由动态端口创建生成的端口数量必须等于转换中其他动态端口生成的端口数量（不包括传递端口）。宏转换中的传递端口可以有任意数量的生成端口。

例如，宏转换有三个动态端口 transformed_input1、transformed_input2 和 passthrough_input。如果 transformed_input1 生成三个端口，则 transformed_input2 必须生成一个或三个端口。passthrough_input 可以生成任意数量的端口。

宏转换示例

您是一家医疗保险公司的数据管理者，需要对保单持有人的个人信息进行加密。数据源的架构可能不同，因此需要使用动态映射。

您需要一个动态映射，从源读取数据，使用数据屏蔽转换加密所有字符串数据，并将修改后的数据写入目标。由于数据屏蔽转换不支持动态功能，因此需要使用宏转换来为包含数据屏蔽转换的 Mapplet 启用动态功能。宏转换允许 Mapplet 转换逻辑充当动态映射逻辑，方法是接受任意数量的字符串端口的输入并通过数据屏蔽 Mapplet 的副本传递每个字符串输入端口。

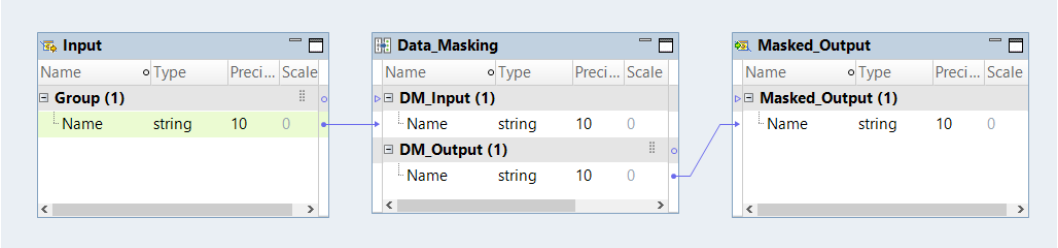
要使用宏转换，请执行以下步骤：

1. 配置 Mapplet。
2. 使用宏转换配置动态映射。
3. 运行映射。

配置 Mapplet

在配置宏转换之前，需要创建一个执行数据加密的 Mapplet。Mapplet 包括输入转换、数据屏蔽转换和输出转换。输入转换有一个字符串类型的端口，该端口连接到数据屏蔽转换的输入组。在数据屏蔽转换中，为端口设置屏蔽类型，然后将输出端口连接到输出转换。

下图显示了 Mapplet：



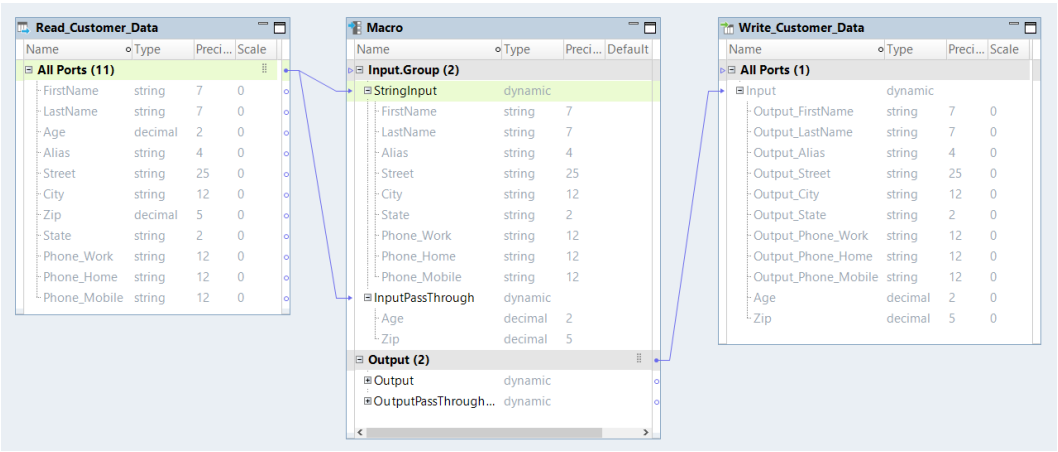
配置映射

配置 Mapplet 后，需要配置动态映射。您将创建一个从参数化源读取的映射，并配置该映射以在运行时获取列名。然后向动态映射添加宏转换，并选择数据屏蔽 Mapplet 作为宏指令。

将读取转换端口组连接到输入端口 *StringInput* 和输入传递端口 *InputPassThrough*。将 *StringInput* 的输入规则配置为仅包括字符串类型的端口。

最后，将写入转换添加到映射，并将其配置为通过映射流定义端口。您可以将宏转换输出组连接到写入转换动态输入端口。

下图显示了动态映射：



运行映射

当您运行映射时，映射将从数据源确定端口。宏转换为每个字符串输入端口创建一个 Mapplet 的副本，并通过 Mapplet 的副本传递每个字符串端口。数据屏蔽转换对字符串数据进行加密，并通过宏转换将输出传回映射。映射将加密的字符串数据和未更改的十进制数据写入目标。

非本地环境中的宏转换

非本地环境中的宏转换处理取决于运行转换的引擎。

请注意是否支持以下非本地运行时引擎：

- Blaze 引擎。无限支持。
- Spark 引擎。对批处理和流无限支持。
- Databricks Spark 引擎。无限支持。

第 29 章

匹配转换

本章包括以下主题：

匹配转换概览

匹配转换属于主动转换，用于分析不同记录之间的相似度水平。 使用匹配转换可以查找在一个数据集中或者在两个数据集之间包含重复信息的记录。

匹配转换分析输入端口上的值并生成一组数值型得分，用于表示不同值之间的相似度。 可以选择多个端口来确定两个输入记录之间的总体相似度水平。 指定一个最低得分作为阈值来标识可能包含重复信息的记录。

匹配转换可以用于以下数据项目：

- 客户关系管理。例如，一家店铺想要设计一个电子邮件营销活动，因此需要检查客户数据库中的重复客户记录。
- 合并和收购。 例如，一家银行收购了同一地区的另一家银行，并且两家银行拥有共同的客户。
- 合规性监管。 例如，一家企业所要遵守的政府或行业法规要求所有数据系统都不包含重复的记录。
- 财务风险管理。 例如，某家银行可能想要搜索帐户所有者之间的关系。
- 主数据管理。 例如，一家零售连锁店有一个主客户记录数据库，旗下的每家零售店都定期向主数据库记录提交。
- 必须标识数据集中的重复记录的任何项目。

匹配分析

可以在匹配转换中定义不同类型的重复项分析。 所定义的重复项分析操作依赖于映射中的数据源数量以及源中包含的信息类型。

配置匹配转换时，请考虑以下因素：

- 可以选择数据集中的一个列，也可以选择多个列。
- 可以分析一个数据源中的多个列，也可以分析两个数据源。
- 可以将匹配转换配置为分析输入端口字段中的原始数据，也可以将该转换配置为分析数据中的标识信息。
- 可以将匹配转换配置为写入不同类型的输出。 所选输出类型决定转换写入的记录数量和记录顺序。
- 要提高性能，请在执行匹配分析之前将输入记录分成不同的组。

列分析

配置匹配转换时，请选择要用于进行分析的一个或多个列。

匹配转换将成对分析这些列。选择单个列进行分析时，转换将创建该列的临时副本，并将源列与临时列进行比较。选择两个列进行分析时，转换将比较选定的两个列中的值。转换会将其中一个列中的每个值与另一个列中的所有值相比较。转换会返回所分析的每个值对的匹配得分。

在匹配转换中配置策略时，您可以选择要分析的列。该策略指定了要分析的列以及应用到这些列的算法。该算法将计算每个值对的相似度水平。转换中的不同算法使用不同的条件来计算值与值之间的相似度水平。您可以在转换中定义多个策略，还可以向每个策略分配不同的列。

列分析示例

您希望比较一列姓氏数据中的值。 创建一个包含数据源和匹配转换的映射。 将*姓氏*端口连接到匹配转换。当映射运行时，转换将在*姓氏*端口上创建数据的临时副本。

下图显示了 surname 数据的一个片段：

	A	B
1	**Surname**	**Surname_1**
2	Annan	Annan
3	Baker	Baker
4	Barker	Barker
5	Edwards	Edwards
6	Parker	Parker
7	Smith	Smith
8	Smith	Smith
9	Zhang	Zhang

映射将生成一组匹配得分，以指示以下值可能重复：

- Baker, Barker
- Barker, Parker
- Smith, Smith

检查这些数据时，您判定 *Baker*、*Barker* 和 *Parker* 不属于重复值。您还判定 *Smith* 和 *Smith* 属于重复值。

单源分析和双源分析

可以将匹配转换配置为分析一个或两个数据源中的数据。 在转换中定义策略时，请从每个数据源选择端口。

将转换配置为执行单源分析时，请选择一个数据集中的一个或多个端口。 将转换配置为执行双源分析时，请选择每个数据集中的一个或多个端口。 请成对选择端口。 对于选定的每对端口，转换会将一个端口中的每个值与其他端口中的每个值进行比较。 如果对单个列中的数据执行单源分析，转换将创建所选端口的临时副本。

注意: 执行标识匹配分析时，您可以将数据源与在先前的映射中创建的标识数据的永久性索引相比较。 通过**匹配类型**选项指定使用永久性索引的标识分析。

字段匹配分析和标识匹配分析

可以配置一个匹配转换以执行字段匹配分析或标识匹配分析。

在字段匹配分析中，匹配转换分析进入转换的源数据。 可以针对任何数据类型执行字段匹配分析。在标识匹配分析过程中，匹配转换将从输入数据生成备选数据值的索引，并对索引数据进行分析。如果输入端口包含标识数据，请将匹配转换配置用于标识匹配分析。标识是用于标识人员或组织的一组数据值。

数据集可以用不同的方式表示单个标识。例如，以下数据值表示姓名 John Smith：

- John Smith
- Smith, John
- jsmith@email.com
- SMITHJMR

匹配转换读取记录中的标识数据，并计算可能的标识备选版本。 转换将创建一个索引，其中包含标识的当前版本和备选版本。 匹配转换会分析索引值，但不会分析输入记录中的值。

标识填充文件

标识匹配操作读取名为 populations 的引用数据文件。社群文件定义标识数据中的潜在变体。这些文件不会随 Informatica 应用程序一同安装。您需要从 Informatica 购买和下载社群数据文件。

请将这些文件安装到内容管理服务能够访问的位置。使用 Informatica Administrator 可设置内容管理服务上的位置。

匹配分析中的组

如果转换必须执行的数据比较次数较多，则匹配分析映射可能需要运行很长时间。比较的次数与所选端口的数据值个数相关。

下表显示了某个映射对一个端口中不同数量的数据值执行的计算次数：

数据值数量	比较次数
10,000	5000 万
100,000	50 亿
100 万	5000 亿

要缩短映射的运行时间，请将输入数据记录分配到组。组是指一组记录，这些记录在指定的端口上包含相同值。对分组数据执行匹配分析时，匹配转换将分析每个组中的记录。转换不会将一个组中的记录与其他组中的记录相比较。组可减少转换必须执行的总的比较次数，而不会降低映射分析的准确性。

将数据组织成组时，请考虑以下规则和准则：

- 用于对数据进行分组的端口是组键端口。组键端口必须包含一个重复值范围，例如地址数据集中的城市名称或州名。如果映射数据不包含可用的组键端口，请使用键生成器从当前映射数据创建该端口。从键生成器转换将组键输出端口连接到匹配转换。

  您也可以使用键生成器转换将序列标识符添加到映射数据。
- 字段匹配操作必须指定组键端口。如果将匹配转换配置用于标识分析，请勿选择组键端口。标识分析将为标识索引数据生成组键。
- 请勿指定您计划在匹配分析中使用组建端口。
- 创建组时，必须验证组的大小是否有效。如果组太小，匹配分析可能无法找到数据集中的所有重复数据。如果组太大，匹配分析可能会返回假的重复项。选择创建的平均组大小为 10000 条记录的组键。
- 组不会对记录在映射数据集中的位置进行重新排序。

匹配对和群集

匹配转换能够读取和写入不同数量的输入行和输出行，并且能够更改输出行的序列。 您决定匹配分析结果的输出格式。

转换可以按以下格式写入行：

匹配对

转换为与满足匹配阈值的得分相匹配的每一对记录写入一行。 转换将每一对记录写入到一个行中。

因为一个记录可能与多个其他记录相匹配，所以一个记录可以出现在多个输出行上。

最佳匹配

转换为数据集中的每条记录写入一行，然后将另一个数据集中最相似的记录添加到同一行。

群集

转换根据不同记录之间的相似度水平将输出记录分配到群集。 群集是指一组记录，其中每个记录至少与一个得分满足匹配阈值的其他记录相匹配。 转换将每个记录写入到一个行中。

群集中的每个记录必须至少与群集中的另一个记录相匹配。 因此，群集可以包含彼此不匹配的记录对。 如果某个记录与其他所有记录均不匹配，则可以使用一个仅包含该记录的群集。

注意: 字段分析中的“群集”选项与标识分析中的“群集 - 全部匹配”选项相对应。 标识分析中的“群集 - 最佳匹配”选项将群集计算与匹配对计算组合使用。

在转换的**匹配输出**视图中配置输出选项。

相关主题：

- “群集输出选项” 页面上 397

匹配得分计算

匹配得分是表示两个列值之间相似度的数值。 通过某种算法计算匹配得分，并以一个介于 0 到 1 之间的小数值表示。 如果两列值完全相同，算法将分配得分 1。

如果选择多个列对进行分析，转换将根据选定列中的得分计算平均得分。 默认情况下，转换会为每个列对的得分分配相等的权重。 转换不会推理数据集中列数据的相对重要性。

您可以编辑转换用于计算匹配得分的权重值。 如果要为数据集中的列分配较高或较低的优先级，请编辑权重值。

还可以设置当转换在列中发现空值时要应用的得分。 默认情况下，转换会将空值视为数据错误，并为包含空值的任何值对分配较低的匹配得分。

注意: 选定的算法将确定两个值之间的匹配得分。 算法将为两个值生成一个得分。 匹配得分并不取决于您选择的匹配输出类型或计分方法类型。

加权得分

为匹配分析选择多个列时，转换会根据这些列中的得分计算每个记录的平均得分。 平均得分包括应用到每列的比较算法的任何权重值。

默认情况下，所有算法使用的权重值均为 0.5。如果选定列很有可能包含重复信息，则可以增大此值。 如果选定列中的重复值不太能指示记录间的真正重复信息，则可以减小权重值。 匹配转换使用平均得分作为每对记录的单一匹配得分。

空值匹配得分

当一个或两个值为空值时，匹配算法将应用预定义的匹配得分。可以编辑字段匹配算法应用于空值的匹值配得分。

空值匹配得分和字段匹配算法

配置字段匹配算法时，请验证算法应用于空数据的匹值配得分值。当字段匹配算法比较了两个值且其中一个值为空值或两个值均为空值时，将应用 0.5 默认得分。得分为 0.5 指示数据值之间的相似度较低。

验证空值匹配得分时，请考虑以下规则和准则：

- 当算法分析的列包含主键或其他重要数据时，请勿编辑默认得分。在此情况下，空值代表数据错误，默认得分适用于数据。
- 当算法分析的列可选择性地包含数据时，将空值匹配得分值更新至与匹配阈值相同的值。当将空值匹配得分设置为匹配阈值时，将取消空值对匹配分析的影响。

空值匹配得分和标识匹配算法

当标识匹配算法比较了两个值且其中一个值为空值或两个值均为空值时，应用的匹配得分为 0。标识匹配分析将向唯一记录群集分配具有空匹配得分的记录，并将群集大小值记录为 1。无法编辑标识匹配算法应用于空数据的得分。

群集输出选项

如果希望将相似或相同记录组织在输出数据中，可选择群集输出选项。

选择群集输出选项时，转换会将一个集群 ID 值添加到每个输出记录。可以按照群集 ID 值对记录进行排序。转换为每个记录输出一行。如果某个记录与具有满足匹配阈值的得分的另一个记录不匹配，转换会将唯一群集 ID 分配给该记录。使用**匹配输出**视图可选择或更新群集输出选项。

可以选择以下群集输出选项：

群集

选择将群集 ID 值分配给输出记录的选项。

群集 - 最佳匹配

选择将具有最高匹配得分的记录对添加到群集的选项。因为某个记录可能表示与多个其他记录的最佳匹配项，所以多个记录对可以共享一个群集 ID 值。

群集 - 全部匹配

群集 - 全部匹配选项与**群集**选项的作用方式相同。

在标识匹配分析中，转换使用**群集 - 全部匹配**和**群集 - 最佳匹配**作为选项名称。

注意: 如果一项数据集成服务同时运行多个匹配转换，则该数据集成服务会为每个转换的输出生成唯一的群集 ID 值。因此，每个转换生成的记录的群集 ID 值可以不是连续的。

“群集”选项和“群集 - 全部匹配”选项

在字段匹配分析中，选择“群集”选项。在标识匹配分析中，选择“群集 - 全部匹配”选项。

匹配转换使用以下规则创建群集：

- 如果两个记录的匹配得分满足匹配阈值，匹配转换会将这两个记录添加到群集。
- 如果数据集中的某个记录与群集中的任何记录匹配，该转换会将这个记录添加到群集中。
- 如果一个群集中的某个记录与另一个群集中的记录匹配，进程会合并两个群集。
- 该转换对匹配结果执行连续清理，直至所有记录均属于群集。
- 如果某个记录与数据集中的任何其他记录都不匹配，转换会将唯一群集 ID 值分配给该记录。

“群集 - 最佳匹配”选项

在标识匹配分析中，选择“群集 - 最佳匹配”选项。

转换使用以下规则创建群集：

- 转换识别与当前记录的匹配得分最高的记录。如果匹配得分符合阈值，转换会将记录对添加到群集中。
- 如果有一个匹配记录位于群集中，转换会在当前群集中添加其他记录。
- 该转换对匹配得分结果执行连续清理，直至所有记录均属于群集。
- 如果某个记录与数据中的其他所有记录均不匹配，则可以使用一个仅包含该记录的群集。

注意: 可以使用**匹配输出**视图上的**匹配**属性指定转换如何将单个数据源与永久性数据存储进行比较。**匹配**属性确定转换是在源数据还是永久性数据存储中查找重复项。

相关主题：

- “匹配对和群集” 页面上 396

群集分析中的驱动程序得分和链接得分

如果在匹配转换中选择了群集输出选项，则可以在输出中添加链接得分和驱动程序得分数据。

链接得分是指可指示两个记录是同一群集中成员的两个记录之间的得分。 记录之间的链接将确定群集的组成。 任何记录都可以链接到同一群集中的其他任何记录。

驱动程序得分是指群集中具有最高序列 ID 值的记录与同一群集中另一个记录之间的得分。 驱动程序得分提供了访问群集中所有记录（而非单个记录）的方法。 如果在匹配输出中添加驱动程序得分，映射运行速度将减慢，因为匹配转换必须在所有群集完成后才能计算驱动程序得分。

注意: 匹配分析将针对您定义的每个策略生成一组得分。 驱动程序得分和链接得分指示每个群集中不同记录对的匹配得分。 驱动程序得分和链接得分可能会依赖于记录进入转换的顺序。 驱动程序得分可能低于匹配阈值。

群集分析示例

配置字段匹配策略以分析姓氏数据列。 在策略中设置 0.825 的匹配阈值。 选择一种群集输出格式，运行转换上的数据查看器。

下表显示了数据查看器所显示的数据：

姓氏	序列 ID	群集 ID	群集大小	驱动程序 ID	驱动程序得分	链接 ID	链接得分
SMITH	1	1	2	1 - 6	1	1 - 1	1
SMYTH	2	2	2	1 - 3	0.83333	1 - 2	1
SMYTHE	3	2	2	1 - 3	1	1 - 2	0.83333
SMITT	4	3	1	1 - 4	1	1 - 4	1
SMITS	5	4	1	1 - 5	1	1 - 5	1
SMITH	6	1	2	1 - 6	1	1 - 1	1

数据查看器包含以下关于姓氏数据的信息：

- SMITT 和 SMITS 不匹配任何得分符合匹配阈值的记录。 匹配转换将确定记录在数据集中是唯一的。
  SMITT 和 SMITS 的群集大小为 1。 要找到群集输出中的唯一记录，请搜索包含单个记录的群集。
- SMITH 和 SMITH 的链接得分是 1。 匹配转换将确定记录是完全相同的。 转换将记录添加到单个群集。
- SMYTH 和 SMYTHE 的链接得分是 0.83333。得分超出匹配阈值。 因此，转换将记录添加到单个群集。

主数据分析

在匹配转换中分析两个数据源时，必须将其中一个源标识为主数据集。 转换会将所指定的数据集中每条记录的数据值与第二个数据集的每个记录中的相应值进行比较。

在许多组织中，主数据集构成永久的高质量数据存储。 向主数据集添加记录之前，请使用匹配转换确认记录不会向主数据中添加重复信息。

主数据示例

一家银行需要维护客户账户记录的主数据集。 这家银行每天都会将用于标识新客户账户的记录更新到主数据集中。 银行使用重复分析映射来确认新记录与主数据集中的客户信息不重复。 主数据集和新账户表具有通用结构，但这些表使用类型相同的数据库。 因此，每次需要更新主数据集时，银行都可以重用重复分析映射。

主数据集分析的方向性

匹配转换在单个方向上比较两个数据集中的记录。 该转换会将主数据集中的每个记录与第二个数据集中的所有记录相比较， 但不将第二个数据集中的每个记录与主数据集中的所有记录相比较。 因此，主数据集的选择会影响匹配分析的结果。

下表显示可在标识匹配分析中比较的两个数据集：

数据集 1	数据集 2
Alex Bell	Alexander Bell
Alexander Graham Bell	Thomas Edison

数据集 1	数据集 2
Alva Edison	Nicola Tesla
Marie Curie	Irene Joliot Curie
Dorothy Crowfoot	Dorothy Hodgkin

如果选择数据集 1 作为主数据集且选择**最佳匹配**输出选项，输出将包括以下记录：

- Alex Bell、Alexander Bell
- Alexander Graham Bell、Alexander Bell

如果选择数据集 2 作为主数据集且选择**最佳匹配**输出选项，输出将包括以下记录：

- Alexander Bell、Alex Bell

数据集 2 为主数据集时，该转换无法匹配 Alexander Bell 和 Alexander Graham Bell，因为 Alexander Bell 已匹配输出数据中的 Alex Bell。

映射重用

如果您定期向主数据集添加数据，请配置能够重用的重复分析映射。 如果与主数据集进行比较的数据源上的端口配置不会更改，则您可以重复使用该映射。

运行映射时，请验证转换是否指定了主数据和较新的数据。可以运行映射但不更新任何其他配置。

标识匹配分析和持久性索引数据

运行映射以分析标识信息时，匹配转换将生成一个索引，以存储数据集中标识的任何备选版本。默认情况下，匹配转换将索引数据写入临时文件。可以将该转换配置为将索引数据保存到数据库表。

生成的索引表代表可以在后续映射中重复使用的数据存储。您可以将索引表与数据源相比较，也可以用数据源中的索引数据更新索引表（可选）。由于转换不会重新生成索引表，因此后续映射的运行时间将缩短。此外，索引表还可以代表标识数据的受信任数据存储。

相关主题：

- “标识分析中的匹配性能” 页面上 403

永久性索引数据的规则和准则

将匹配转换配置为分析标识信息的主数据集时，请考虑以下规则和准则：

- 要生成主数据集的可重用索引，请将转换配置为将索引数据写入数据库表。数据库表构成了索引数据的永久性存储。
- 要比较另一数据集与索引数据存储中的标识，请将该数据集配置为映射数据源。配置匹配转换，以读取数据源和索引数据存储。从您指定的默认数据库连接架构中选择索引表。
- 匹配转换会将输入记录中的序列标识符值添加到与该记录对应的索引数据行。*SequenceID* 输入端口包含序列标识符。转换会使用序列标识符来跟踪匹配分析过程中各个步骤的索引数据。请勿断开序列 ID 端口的连接。

- 将匹配转换连接到索引存储时，转换会重用创建了该存储的转换中的填充、键级别、键类型和键字段属性值。转换还会重复使用创建了该存储的转换中的端口配置。

  如果转换属性不匹配，则标识分析将无法正确比较映射源数据和索引数据。
- 匹配转换会使用选定为键字段的输入端口上的数据来生成标识索引。该转换还可以将来自其他端口的数据写入到索引。如果您断开非键字段数据端口与转换的连接，则在运行映射时会清除对应索引列中的所有数据。要保留索引表中的输入端口数据，请勿断开输入数据端口的连接。
- 为数据集生成索引表数据时，您可以在匹配转换中禁用匹配分析。例如，为数据集创建索引存储时，您可以禁用匹配分析。如果禁用匹配分析，映射的运行速度会更快。

  如果禁用匹配分析，则匹配转换可以生成并显示持久性状态代码和持久性状态说明。转换不会生成或显示匹配得分或与匹配分析结果相关联的其他数据。例如，如果您将转换配置为向群集分配记录，并且禁用匹配分析，则该转换不会生成或显示群集 ID 值。
- 您应确定匹配转换是否用映射源中的数据更新索引存储。匹配转换使用序列标识符来确定索引存储中的行和映射数据中的行是否代表相同记录。

匹配映射性能

运行包含匹配转换的映射之前，可以预览决定转换性能的数据因素。可以验证系统是否具有运行映射所需的资源。您还可以验证是否正确配置了转换，以测量输入数据中的相似度水平。

使用**匹配性能分析**选项可以验证系统是否具有所需的资源。使用**匹配群集分析**选项可验证映射能否准确测量输入数据中的相似度水平。

对读取单一数据源的任何匹配转换运行匹配性能分析和匹配群集分析。在执行双源字段匹配分析的任何匹配转换中运行匹配性能分析。不要对连接到索引表的标识匹配策略运行匹配性能分析或匹配群集分析。

对匹配性能分析进行向下钻取

您可以对匹配分析数据进行向下钻取，以查看满足或超出匹配阈值的记录对。在**详细信息**视图中双击记录，并使用数据查看器查看与所选记录匹配的记录。数据查看器会将每个记录对的数据显示在单个行上。行包含该对中每个记录的行标识符。

对匹配群集分析进行向下钻取

您可以对群集分析数据进行向下钻取，以查看每个群集中的记录。在**详细信息**视图中双击群集，并在数据查看器中读取数据。数据查看器一次只能显示一个群集。群集数据包括所选得分选项，例如，驱动程序得分、链接得分、驱动程序标识符或链接标识符。

匹配转换日志记录

运行使用匹配转换的映射时，Developer 工具日志将跟踪该映射所执行的比较计算次数。要查看日志数据，请在数据查看器中选择**显示日志**选项。

映射每执行 100,000 次计算就会更新一次日志。

查看匹配群集分析数据

您可以查看转换能够创建的群集的统计数据。 群集统计信息根据当前映射配置汇总数据集中的记录重复级别。

要查看数据，请右键单击“匹配映射画布中的转换”，然后选择**匹配群集分析**。

开始运行分析之前，请验证映射是否包含转换。

匹配群集分析显示以下属性的数据：

属性	说明
源	输入数据行的数量。
上次运行	分析的日期和时间。
已发现群集的总数	映射运行时匹配分析生成的群集数量。
最小群集大小	包含最少记录的群集中的记录数。 如果最小群集大小为 1，则数据集将至少包含一个唯一的记录。
最大群集大小	包含最多记录的群集中的记录数。 如果此值大大超过平均群集大小，最大的群集中可能包含假重复项。
唯一记录的数量	数据集中与匹配程度满足匹配阈值的其他记录不匹配的记录数。
重复记录的数量	数据集中与匹配程度满足匹配阈值的其他记录匹配的记录数。
总比较次数	映射执行的比较操作的次数。
平均群集大小	群集中记录的平均数。

查看匹配性能分析数据

您可以查看记录组的统计数据，映射将这些数据读取为输入数据。

要查看数据，请右键单击“匹配映射画布中的转换”，然后选择匹配性能分析。

开始运行分析之前，请验证映射是否包含转换。

匹配性能分析显示以下属性的数据：

属性	说明
源	输入数据行的数量。
上次运行	分析的日期和时间。
已发现组的总数	为数据集定义的组数量，具体取决于选定的组键值。
吞吐量(每分钟的记录数)	用于估计匹配分析速度的变量值。 该值由您负责设置。 使用该值可估计运行匹配分析需要的时间。
匹配记录的估计时间	分析数据集中的所有记录需要的时间，具体取决于匹配转换配置。
生成对的总数	转换必须执行的比较数量，具体取决于输入数据行的数量和组数量。
最小组大小	用于指示组可以包含的最小记录数量的变量值。 该值由您负责设置。 使用该值可验证映射是否会创建大小不可用的组。 **注意:** 最小组大小值不决定映射运行时创建的组的大小。
低于最小阈值的组数	包含的记录数小于最小组大小值的组数。 如果许多组的大小都小于最小组大小，您可能需要编辑转换并选择其他组键。

属性	说明
最大组大小	用于指示组可以包含的最大记录数量的变量值。 您负责为性能分析设置该值。 使用该值可验证映射是否会创建大小不可用的组。 **注意:** 该值不决定映射运行时创建的组的大小。
高于最大阈值的组数	包含的记录数大于最大组大小值的组数。 如果许多组的大小都大于最大组大小，您可能需要编辑转换并选择其他组键。

标识分析中的匹配性能

要提高对两个数据集执行标识分析时的映射性能，请将匹配转换配置为从数据库表读取标识索引数据。运行映射可为主数据集创建索引表。再次运行该映射可将索引数据与其他数据源相比较。

使用**匹配类型**视图中的选项可标识用于存储索引数据的数据库表。配置转换以将索引数据与其他源中的数据相比较时，请使用相同的选项来选择索引表。

要将索引数据写入数据库表，请执行以下任务：

1. 创建读取标识信息的数据源的映射。
2. 将映射中的匹配转换配置为向数据库写入索引数据。
3. 运行映射以生成索引数据。索引数据表示能够重用的数据存储。

要从数据库表中读取索引数据，请执行以下任务：

1. 创建一个读取其他标识数据源的映射。
2. 将该映射中的匹配转换配置为从您先前指定的数据库中读取索引数据。

   映射数据源与索引数据共享一个通用结构时，您可以重用生成索引数据的映射。
3. 运行该映射以将数据源与索引数据进行比较。

   该映射将生成数据源的索引数据。映射不需要生成较大数据集的索引数据。因此，与生成两个数据集的索引数据的双源映射相比，该映射运行的速度更快。

相关主题：

- “标识匹配分析和持久性索引数据” 页面上 400

为标识索引数据创建数据存储

配置读取包含标识信息的数据源的映射。使用匹配转换将索引数据写入数据库。

1. 创建一个映射，然后将该数据源添加到映射画布中。
2. 将匹配转换添加到映射画布中。
3. 在数据源中，选择包含标识信息的端口。
   - 将标识信息端口连接到匹配转换。
   - 将包含序列标识符值的端口连接到匹配转换。
4. 在匹配转换中，选择**匹配类型**视图。
5. 将匹配类型设置为**使用永久性记录 ID 进行标识匹配**。

6. 配置以下选项以创建索引数据存储。
   - 将持久性方法设置为**使用新的 ID 更新数据库**。
   - 验证匹配进程的值。默认值为 **Enable**。如果禁用匹配进程，则匹配会创建标识索引表，但不会对数据执行匹配分析。
   - 在“数据库连接”菜单中，选择用于索引表的数据库。
   - 在“永久性存储”菜单中，选择**新建**。在**创建存储表**对话框中输入索引的名称。
7. 执行匹配转换要求执行的所有其他配置步骤。例如，配置转换策略。
8. 向映射添加目标数据对象。
9. 将匹配转换输出端口连接到目标数据对象。
10. 运行映射。

映射将数据源的索引数据写入到所指定的数据库表中。

在单源分析中使用索引数据存储

配置读取标识信息的数据源的映射。使用匹配转换比较数据源和主数据集的索引数据存储。

开始配置映射之前，请验证数据源是否包含具有序列标识符值的列。

可复制或重用创建了索引数据存储的映射，以节约时间。

1. 打开生成了索引数据存储的映射。

   或者，打开映射的副本。
2. 验证映射中的数据源。

   如果需要，可将数据源替换为包含当前数据的源。

   注意: 如果删除数据源，匹配转换上的端口连接也会一起删除。
3. 标识包含标识信息的数据源端口。
   - 将标识信息端口连接到匹配转换。
   - 将包含序列标识符值的端口连接到匹配转换。
     输入端口和输入端口顺序必须与创建了索引表的转换上的输入端口一致。
4. 在匹配转换中，选择**匹配类型**视图。
5. 将匹配类型设置为**使用永久性记录 ID 进行标识匹配**。
6. 验证填充、键级别、键类型和键字段值。
7. 配置用于标识索引数据存储的选项：
   - 设置持久性方法。例如，选择**不更新数据库**可将当前数据保留在索引表中。
   - 将匹配进程设置为 **Enable**。
   - 在数据库连接菜单中，选择包含索引表的数据库。
   - 在“永久性存储”菜单中，浏览到包含索引数据的表。
8. 在“匹配输出”视图中配置以下属性：
   - 匹配。当转换从数据库表读取索引数据时标识要分析的记录。
   - 输出。筛选转换作为输出写入的记录。
9. 执行匹配转换要求执行的所有其他配置步骤。

   例如，配置转换策略。
10. 验证映射中的数据目标。

如果需要，可将数据目标替换为其他目标对象。

11. 将匹配转换输出端口连接到目标数据对象。
12. 运行映射。

映射将数据源记录与索引数据存储进行比较。转换将数据源的索引数据写入到数据存储中。

匹配转换视图

匹配转换对能够在一系列视图中配置的选项进行组织。 在可重用转换中，这些视图在 Developer tool 画布中显示为选项卡。 要打开某个视图，请单击一个选项卡。 在不可重用转换中，这些视图在画布中显示为列表。 要打开某个视图，请单击列表中的某个项目。

配置匹配分析操作时，可以配置以下视图：

常规

使用“常规”视图可更新不可重用匹配转换的名称和说明。说明是可选的。

端口

使用“端口”视图可验证不可重用匹配转换的输入端口和输出端口。

概览

使用“概览”可更新不可重用转换的名称和说明。说明是可选的。还可以使用“概览”创建可重用转换的输入端口和输出端口。

匹配类型

使用“匹配类型”视图可选择转换执行的重复项分析的类型。 可以选择字段匹配分析或标识匹配分析。 可以指定单数据源或双数据源。

策略

使用“策略”视图可定义用于分析输入数据的一个或多个策略。 请在每个策略中选择两个数据列，然后将匹配分析算法分配给这些列。

匹配输出

使用“匹配输出”视图可指定输出数据的结构和格式。

参数

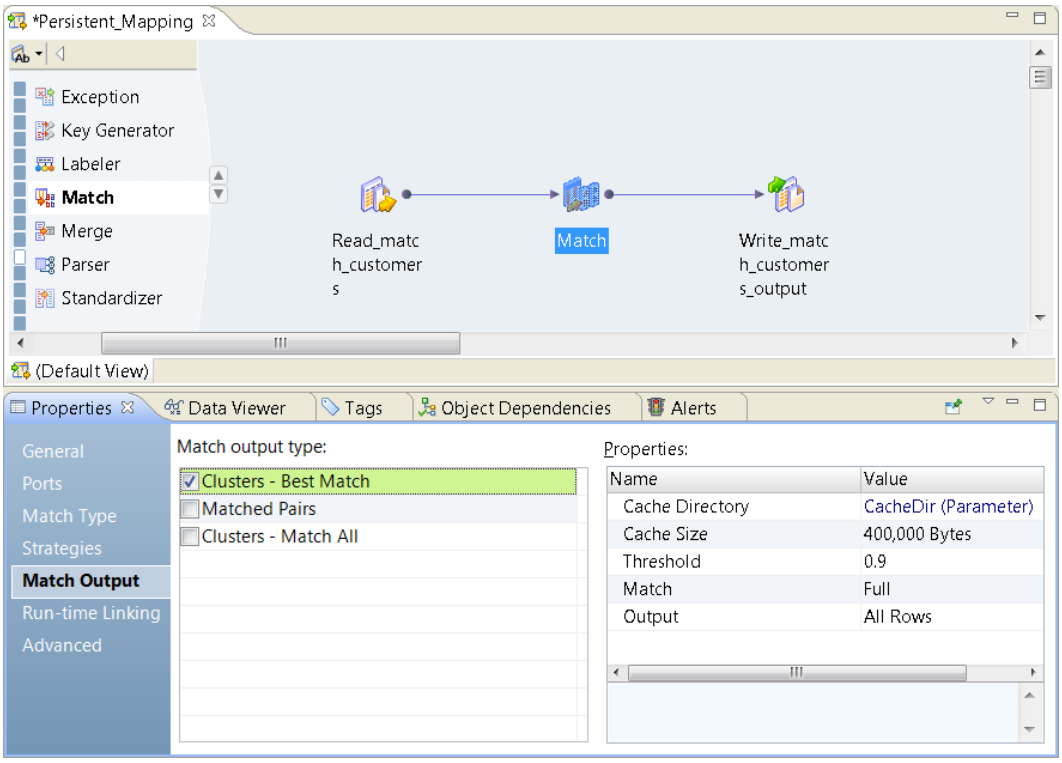
使用“参数”视图可定义运行包含转换的映射时数据集成服务能够对该转换应用的任何参数。

高级

使用“高级”视图可指定以下属性：

- 任何日志消息中映射为转换写入的详细信息级别。
- 运行包含转换的映射时标识匹配操作使用的进程数量。
- 转换是否将相同记录中的数据直接传递到输出端口。 配置转换以写入群集输出时，可以筛选相同的记录。

下图显示了一个您配置使用永久性索引存储进行标识分析的匹配转换的视图：



匹配转换端口

匹配转换包括一组预定义的输入端口和输出端口，这些端口中包含所定义的匹配分析操作的元数据。 当您配置匹配类型和匹配输出选项时，转换将选择或清除这些端口。

配置转换时，请检查元数据端口。 将转换添加到映射时，请验证您是否将元数据端口连接到上游和下游映射对象的正确端口。

匹配转换输入端口

预定义的输入端口包含转换执行匹配分析所需的元数据。

创建匹配转换后，可以配置以下输入端口：

SequenceId

映射源数据集中每条记录的唯一标识符。 输入数据集中的每条记录都必须包括一个唯一的序列标识符。 如果数据集中包含重复的序列标识符，匹配转换将无法正确标识重复记录。 如果数据中没有标识符，则可以使用键生成器转换创建唯一的标识符。

为标识数据创建索引数据存储时，匹配转换会将每条记录的序列标识符都添加到数据存储中。 配置转换以将数据源与索引数据存储进行比较时，转换可能会查找这两个数据集中的通用序列标识符。 如果序列标识符在各个数据集中是唯一的，转换将能够对这些标识符进行分析。

GroupKey

用于标识记录所属的组的键值。

注意: 要提高映射性能，请使用相同的精度值配置连接到映射的 GroupKey 输入端口和输出端口。

匹配转换输出端口

预定义的输出端口包含与转换执行的分析有关的元数据。

创建匹配转换后，可以配置以下输出端口：

GroupKey

用于标识记录所属的组的键值。

下游转换（例如关联转换）可以读取组键值。

ClusterId

记录所属的群集的标识符。 在群集输出中使用。

ClusterSize

记录所属的群集中的记录数。 群集包含唯一的记录时，其大小为 1。 在群集输出中使用。

RowId 和 RowId1

记录的唯一行标识符。 匹配分析操作过程中，匹配转换使用行标识符标识行。 标识符可能与输入数据中的行数不匹配。

DriverId

群集中驱动程序记录的行标识符。 在群集输出中使用。驱动程序记录是群集中具有最高 SequenceID 输入端口值的记录。

DriverScore

转换将分配匹配对输出和群集输出中的驱动器得分。 在匹配对中，驱动程序得分是记录对之间的匹配得分。在群集中，驱动程序得分是群集中的当前记录与驱动程序记录之间的匹配得分。

LinkId

与当前记录匹配并链接到群集的记录的行标识符。 在群集输出中使用。

LinkScore

导致创建群集或在群集中增加记录的两个记录之间的匹配得分。 LinkID 端口标识当前记录与其共享链接得分的记录。 在群集输出中使用。

PersistenceStatus

表示输入记录的匹配分析结果的八字符代码。 转换将数据源与索引数据存储进行比较时，在单源标识分析中使用。

转换将在代码中填充三分之一的字符。 转换可以在每个位置返回不同的字符。 对于位置四到八，转换将返回 0。

将转换配置为生成匹配对输出时，转换将创建 PersistenceStatus 端口和 PersistenceStatus1 端口。

PersistenceStatusDesc

持久性状态代码值的文本说明。转换将数据源与索引数据存储进行比较时，在单源标识分析中使用。

将转换配置为生成匹配对输出时，转换将创建 PersistenceStatusDesc 端口和 PersistenceStatusDesc1 端口。

持久性状态代码和持久性状态说明

持久性状态代码和持久性状态说明描述了匹配转换所分析的各种索引数据类型之间的关系。当您配置转换以读取永久性标识数据存储时，转换将生成状态代码和状态说明。

转换会将持久性状态代码写入到 PersistenceStatus 端口。该代码包含八个字符。转换会用代码值填充字符串的前三位。对于位置四到八，转换将返回 0。

转换会将持久性状态说明写入到 PersistenceStatusDesc 端口。该说明包含三个逗号分隔的文本字符串，而这些字符串描述了持久性状态代码中前三位的值。

转换将使用源数据记录中的序列标识符值来比较两个数据集的索引数据。

下表介绍了转换写入到状态说明和状态代码中各个位置的信息的类型：

位置	说明
1	标识包含该记录的数据集。
2	指示记录的重复状态。转换将查找转换输入数据和索引数据存储之间共同的序列标识符。
3	描述转换对数据执行的任何操作。
4-8	状态代码的这几个位置包含 0。 状态说明的这几个位置不包含文本。

状态代码值和状态说明值

持久性状态代码和持久性状态说明描述了转换输入记录和数据存储所代表的记录之间的关系。转换将使用序列标识符值来标识记录，以及确定数据集中各个记录之间的关系。

持久性状态代码和持久性状态说明具有通用结构。状态代码和状态说明的输出数据字符串中每个位置都包含相同信息。

数据集状态

状态代码和状态说明的第一个值标识了包含记录的数据集。

下表介绍了转换可能在第一位返回的状态代码和状态说明：

状态代码	状态说明
S	存储。 当前记录源自索引数据存储。
I	输入。 当前记录源自转换输入数据。

重复记录状态

状态代码和状态说明的第二个值描述了转换索引数据和永久性数据存储之间的关系。

下表介绍了转换可能在第二位返回的状态代码和状态说明：

状态代码	状态说明
A	不存在。 索引数据存储不包含当前记录的数据。
E	存在。 当前记录存在于索引数据存储和转换输入数据中。
I	无效。 转换无法分析当前记录。例如，转换无法生成记录的索引数据，因为“匹配类型”选项卡上的键字段与记录数据不兼容。
N	新的。 数据源中存在记录。
0	[短划线] 记录存在于索引数据存储中。

数据存储状态

状态代码和状态说明的第三个值描述了转换对索引数据表执行的任何操作。

下表介绍了转换可能在第三位返回的状态代码和状态说明：

状态代码	状态说明
A	已添加。 转换会将当前输入记录的索引数据添加到永久性数据存储。 转换输入数据和永久性索引数据具有不同的序列标识符。
I	已忽略。 转换不会将当前输入记录的任何索引数据添加到永久性数据存储。
N	转换将返回以下说明之一： - 无更改。 当前记录源自永久性数据存储，并且转换不会执行任何操作。 - 未添加。 受已定义的匹配策略所限制，转换不会用当前输入记录的任何数据更新永久性数据存储。
R	已删除。 转换将从索引数据存储中删除该记录的索引数据。
U	已更新。 转换将用转换输入记录中的索引数据更新永久性数据存储中的行。 转换输入数据和永久性索引数据具有共同的序列标识符。

持久性状态说明示例

持久性状态代码 INA00000 具有以下持久性状态说明：

输入、新建、已添加

状态代码和状态说明包含以下有关记录的信息：

- 记录源自转换输入数据。
- 永久性数据存储不包含记录的副本。
- 转换会将记录的索引数据添加到永久性数据存储。

输出端口和匹配输出选择

您选择的匹配输出选项决定转换上的输出端口。例如，选择群集输出类型时，转换将创建 ClusterId 端口和 ClusterSize 端口。

选择所需的转换输出类型，并检查转换上的端口。

如果更新了匹配输出类型，请在更新后验证转换的输出端口配置。 如果您在映射中使用了转换，可能需要将输出端口重新连接到映射中的下游对象。

匹配 Mapplet

匹配 Mapplet 是一种可以在匹配转换中创建和嵌入的 Mapplet。

通过将匹配转换的配置另存为匹配 Mapplet，可以创建匹配 Mapplet。 创建匹配 Mapplet 时，可以将匹配转换设置转换到比较转换和加权平均值转换中。

创建匹配 Mapplet 后，可以添加转换以自定义匹配分析。 例如，可以添加表达式转换以评估两个策略的链接得分，然后选择最高得分。

与匹配转换不同，匹配 Mapplet 是被动的，这意味着您可以在 Analyst 工具中将其用作规则。 在数据剖析过程中，在 Analyst 工具中使用匹配 Mapplet 以匹配记录。

匹配转换只能读取在匹配转换内创建的匹配 Mapplet。

创建匹配 Mapplet

创建匹配 Mapplet 可定义使用多个转换的匹配分析操作。

1. 在编辑器中打开匹配转换，然后选择**策略**视图。
2. 选择**使用匹配规则**。
3. 在**名称**字段中，选择**新建**。

   此时将打开**新建 Mapplet** 窗口。
4. 在**新建 Mapplet** 窗口中，为 Mapplet 输入名称并选择 Mapplet 的保存位置。
5. （可选）选择**重用匹配转换中的策略**将当前匹配转换的输入、策略和权重复制到匹配 Mapplet。

   注意: Informatica 建议使用此设置快速创建匹配 Mapplet 以复制当前在匹配转换中定义的匹配功能。
6. 单击**完成**。

   此时将在编辑器中打开匹配 Mapplet。
7. （可选）通过在匹配 Mapplet 中添加和配置比较转换和加权平均值转换来创建匹配操作。
8. 单击**文件** > **保存**以保存 Mapplet。
9. 关闭 Mapplet，然后选择包含该匹配转换的编辑器。验证您创建的 Mapplet 是否显示在**名称**字段中。
10. （可选）通过单击**匹配字段**按钮配置 Mapplet 中的匹配字段。

此时将打开**配置匹配规则**窗口。

11. 双击**输入字段**和**可用输入**列中的字段，为匹配输入分配输入端口。
12. 单击**文件** > **保存**以保存转换。

使用匹配 Mapplet

您可以选择和配置以前在匹配转换中定义的匹配 Mapplet。

1. 在编辑器中打开匹配转换，然后选择**策略**视图。
2. 选择**使用匹配规则**。
3. 在**名称**字段中，选择**使用现有的**。

   此时将打开**配置匹配规则**窗口。
4. 单击**浏览**以在存储库中查找匹配 Mapplet。

   重要说明: 只能选择匹配转换所创建的 Mapplet。

   此时将打开**选择匹配 Mapplet** 窗口。
5. 选择匹配 Mapplet，然后单击**确定**。
6. 双击**输入字段**和**可用输入**列中的字段，为匹配输入分配输入端口。
7. 单击**确定**。

   此时将关闭**配置匹配规则**窗口。
8. 单击**文件** > **保存**以保存匹配转换。

配置匹配分析操作

要配置匹配操作，请将源数据连接到匹配转换，然后在转换视图中编辑相应属性。

1. 创建匹配转换，然后将源数据连接到该转换。
2. 选择**匹配类型**视图，然后选择一种匹配类型。
3. 配置所选匹配操作类型的属性。

   如果选择了双源匹配类型，请配置**主数据集**属性。
4. 选择**策略**视图，然后选择**定义匹配策略**。
5. 单击**新建**。

   此时将打开**新建匹配策略**向导。
6. 选择匹配策略，然后单击**下一步**。
7. （可选）编辑权重和空值匹配设置。单击**下一步**。
8. 双击“可用”列中的单元格以选择要分析的输入端口。

   单击**下一步**以配置其他策略，或者单击**完成**以退出向导。

   注意: 要编辑策略配置，请在**策略**视图中单击单元格中该策略所对应的箭头。
9. 选择**匹配输出**视图。

   选择匹配输出类型，并配置属性。

注意: 还可以通过在**策略**视图中选择或创建匹配 Mapplet 来配置匹配策略。 匹配 Mapplet 是一种可以嵌入到匹配转换中的 Mapplet。

非本地环境中的匹配转换

非本地环境中的匹配转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的匹配转换

支持匹配转换，但存在以下限制：

- 匹配转换在本地和非本地环境中生成群集 ID 值的方式不同。在非本地环境中，转换将组 ID 值附加到群集 ID。
- 当将匹配转换配置为向数据库表写入标识索引数据时，映射验证会失败。

Spark 引擎上的匹配转换

支持匹配转换，但存在以下限制：

- 匹配转换在本地和非本地环境中生成群集 ID 值的方式不同。在非本地环境中，转换将组 ID 值附加到群集 ID。
- 当将匹配转换配置为向数据库表写入标识索引数据时，映射验证会失败。

Databricks Spark 引擎上的匹配转换

支持匹配转换，但存在以下限制：

- 匹配转换在本地和非本地环境中生成群集 ID 值的方式不同。在非本地环境中，转换将组 ID 值附加到群集 ID。
- 当将匹配转换配置为向数据库表写入标识索引数据时，映射验证会失败。

第 30 章

字段分析中的匹配转换

本章包括以下主题：

字段匹配分析

执行字段匹配分析可以在一个数据集中或者两个数据集之间找到相似或重复记录。

配置匹配转换以进行字段匹配分析时，在以下视图上设置选项：

- 匹配类型
- 策略
- 匹配输出

或者，在“参数”和“高级”视图上设置选项。

字段匹配分析的流程

下面的流程概括了配置匹配转换以进行字段匹配分析所要执行的步骤。 可以定义一个单独使用匹配转换或者使用匹配转换和其他转换的流程。

注意: 向字段匹配分析中的映射添加匹配转换时，应在映射中添加一个上游键生成器转换。

要准备用于匹配转换的数据，请执行以下步骤：

1. 将源数据记录整理成组。

   使用键生成器转换为每个记录分配一个组键值。 组分配可减少匹配转换必须执行的计算量。
2. 验证数据源记录是否包含唯一的序列标识符值。 可以使用键生成器转换创建这些值。

在匹配转换中执行以下步骤：

1. 指定字段分析作为匹配类型，并指定数据源的数量。

   如果将转换配置为分析两个数据集，请选择一个主数据集。

   使用**匹配类型**视图设置数据源的类型和数量。

2. 定义匹配分析策略。 选择一种算法，并为该算法分配一对列。

   使用**策略**视图定义策略。

3. 指定转换生成匹配分析结果所用的方法。

4. 设置匹配阈值。 匹配阈值是可以将两个记录标识为彼此重复的最低得分。

   使用**匹配输出**视图选择输出方法和匹配阈值。

   注意: 可以在匹配转换或加权平均值转换中设置匹配阈值。 如果创建了匹配 Mapplet，请使用加权平均值转换。

字段匹配类型选项

“匹配类型”视图包含一个适用于双源匹配分析的选项。 该选项可标识主数据集。 “匹配类型”视图不包含用于单源匹配分析的选项。

分析两个数据集时，必须选择其中的一个作为主数据集。 如果项目或组织中的任何数据集都不代表主数据集，请选择更大的数据集作为主数据集。

使用**主数据集**选项指定主数据集。

字段匹配策略

“策略”视图列出为输入数据定义的策略。

策略决定转换度量数据源记录之间的相似度和差异的方式。

字段匹配算法

匹配转换包括用于比较两个列的数据值的算法。 每个算法都用不同的方式计算数据值之间的差异程度。

选择一种可以度量需要在所选列中找到的数据差异类型的算法。

二元语法

使用“二元语法”算法可比较长文本字符串，例如，在一个字段中输入的邮政地址。

“二元语法”算法将根据两个数据字符串中连续字符的出现情况计算两个字符串的匹配得分。 该算法查找这两个字符串共有的连续字符对。 将在这两个字符串中均匹配的字符对的数量除以字符对的总数。

“二元语法”示例

请考虑以下字符串：

- larder
- lerder

这些字符串将产生以下“二元语法”组：

l a, a r, r d, d e, e r

l e, e r, r d, d e, e r

请注意，字符串“lerder”中第二次出现的字符串“e r”不匹配，因为字符串“larder”中没有第二次出现的对应“e r”。

要计算“二元语法”匹配得分，转换会将匹配对数 (6) 除以两个字符串中的总对数 (10)。 在本示例中，字符串的相似度为 60%，匹配得分为 0.60。

汉明距离

如果数据字符的位置非常关键，可使用“汉明距离”算法，例如，在电话号码、邮政编码或产品代码等数值字段或代码字段中。

“汉明距离”算法通过计算两个数据字符串中字符不同的位置数来计算这两个数据字符串的匹配得分。 对于长度不同的字符串，最长字符串中每个额外字符都将视为两个字符串之间的一个差异。

“汉明距离”示例

请考虑以下字符串：

- Morlow
- M**a**rlow**es**

突出显示的字符指示汉明算法确定为差异的位置。

要计算汉明匹配得分，转换会将匹配字符数 (5) 除以最长字符串的长度 (8)。 在本示例中，字符串的相似度为 62.5%，匹配得分为 0.625。

编辑距离

使用“编辑距离”算法可比较单词或短文本字符串，例如名称。

“编辑距离”算法通过插入、删除或替换字符来计算将一个字符串转换为另一个字符串的最低“成本”。

“编辑距离”示例

请考虑以下字符串：

- Levenston
- Levens**h**t**e**in

突出显示的字符指示将一个字符串转换为另一个字符串所需的操作。

“编辑距离”算法会将未更改的字符数 (8) 除以最长字符串的长度 (11)。 在本示例中，字符串的相似度为 72.7%，匹配得分为 0.727。

Jaro 距离

如果字符串中开头字符的相似性需要优先考虑，可使用“Jaro 距离”算法比较两个字符串。

Jaro 距离匹配得分反映这两个字符串中的前四个字符之间的相似度以及已标识字符转换的数量。 转换通过使用在罚值属性中输入的值来权衡前四个字符之间的匹配重要性。

“Jaro 距离”属性

配置“Jaro 距离”算法时，可以配置以下属性：

罚值

在两个比较字符串中的前四个字符不同时确定匹配得分罚值。 如果首字符不匹配，则转换将减去整个罚值。 转换将根据其他不匹配字符的位置减去相应罚值分数。 默认罚值为 0.20。

区分大小写

确定 Jaro 距离算法在比较字符时是否考虑字符的大小写。

“Jaro 距离”示例

请考虑以下字符串：

- 391859
- 813995

如果使用默认罚值 0.20 来分析这些字符串，“Jaro 距离”算法将返回匹配得分 0.513。 该匹配得分指示字符串相似度为 51.3%。

反向汉明距离

使用反向汉明距离算法可计算两个字符串之间字符位置不同的百分比（从右往左读取）。

“汉明距离”算法通过计算两个数据字符串中字符不同的位置数来计算这两个数据字符串的匹配得分。 对于不同长度的字符串，该算法会将最长字符串中的每个附加字符计算为字符串之间的一个差异。

反向汉明距离示例

请考虑使用以下字符串，采用了从右往左的对齐，以模仿反向海明算法：

- 1-999-9999
- **011-0**1-999-999**1**

突出显示的字符指示反向汉明距离算法识别为有差异的位置。

要计算反向海明匹配得分，该转换会将匹配字符数 (9) 除以最长字符串的长度 (15)。 在本示例中，匹配得分为 0.6，指示字符串的相似度为 60%。

字段匹配策略属性

在**策略**视图上打开**策略**向导，并配置每个字段匹配策略的属性。

配置字段匹配策略时，可以配置以下属性：

名称

按名称标识策略。

权重

确定计算记录的总得分时为匹配得分分配的相对优先级。 默认值为 0.5。

单个字段为空

定义当一对数据值中的一个值为空时算法将应用于该数据值对的匹配得分。 默认值为 0.5。

两个字段都为空

定义当一对数据值中的两个值都为空时算法将应用于该数据值对的匹配得分。 默认值为 0.5。

注意: 当匹配的列值中的一个值为空或两个值都为空时，匹配算法不会计算匹配得分。 算法将应用在空匹配属性中定义的得分。 无法清除空匹配属性。

字段匹配输出选项

配置**匹配输出**选项以定义字段匹配分析的输出格式。

可以在**匹配输出类型**区域和**属性**区域配置这些选项。

匹配输出类型

“匹配输出”视图包括用于指定输出数据格式的选项。 可以将转换配置为将记录写入群集或匹配对中。

请选择以下匹配输出类型之一：

最佳匹配

写入主数据集中的每个记录以及表示第二个数据集中的最佳匹配的记录。匹配操作在第二个数据集中为主记录选择匹配得分最高的记录。如果两个或多个记录返回最高得分，则匹配操作将在第二个数据集中选择第一个得分最高的记录。最佳匹配将每一对记录写入到一个行中。

为双源分析配置转换时，可以选择**最佳匹配**。

群集

包含记录集的写入群集，其中的记录通过满足匹配阈值的匹配得分彼此相链接。每个记录必须至少与群集中的一个其他记录匹配（得分满足阈值）。

在配置转换进行单源分析和双源分析时，可以选择**群集**。

匹配对

写入与满足匹配阈值的得分相互匹配的所有记录对。 转换将每对写入一行，然后将每对的匹配得分添加到每行。 如果记录与多个其他记录相匹配，转换会为每个记录对写入一行。

在配置转换进行单源分析和双源分析时，可以选择**匹配对**。

匹配输出属性

“匹配输出”视图包含的属性可指定高速缓存行为、匹配得分阈值以及显示在转换输出中的匹配得分。

还可以使用匹配输出属性指定转换在输出记录中添加匹配得分值的方式。

选择匹配输出类型后，配置以下属性：

缓存目录

指定数据集成服务在字段匹配分析过程中将临时数据写入到的目录。当匹配分析生成的数据超出可用系统内存量时，数据集成服务会将临时文件写入此目录。数据集成服务会在运行映射后删除这些临时文件。

您可以在该属性上输入目录路径，也可以使用参数标识目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。默认值为 CacheDir 系统参数。

缓存大小

确定数据集成服务分配给字段匹配分析的系统内存量。默认值为 400,000 字节。

在对数据排序之前，数据集成服务会分配您指定的内存量。如果匹配分析生成的数据较多，数据集成服务会将超出的数据写入缓存目录。如果匹配分析所需的内存超出系统内存和文件存储可以提供的内存，映射会失败。

注意: 如果输入 65536 或更高的值，转换将以字节为单位读取值。如果输入低一些的值，转换将以 MB 为单位读取值。

阈值

设置将两个记录标识为可能彼此重复的最低匹配得分。

可以将参数分配给阈值。 可将小数值设置为介于 0 到 1 之间。

计分方法

确定在转换输出中显示的匹配得分值。 选择群集输出的计分方法。

下表介绍了计分方法选项：

计分方法选项	说明
两者	在群集的每个记录中添加链接得分和驱动程序得分。
链接得分	在群集的每个记录中添加链接得分。默认选项。
驱动程序得分	在群集的每个记录中添加驱动程序得分。
无	不在群集的任何记录中添加匹配得分。

注意: 如果在记录中添加驱动程序得分，会增加映射运行时间。 映射将等到所有群集完成后，才向记录添加驱动程序得分值。

字段匹配高级属性

转换包含确定执行实例数、转换如何分析相同行以及日志数据跟踪级别的高级属性。

您可以配置以下高级属性：

执行实例

确定转换在运行时使用的线程数。

匹配转换在字段匹配分析中使用一个执行实例。 配置转换以进行标识匹配分析时，可以编辑执行实例的数量。

筛选完全匹配

确定转换是否会对输入数据中的相同记录对应用匹配策略中的比较算法。

转换发现相同记录对时，该算法不需要分析记录之间的相似度。转换可直接将记录传递至输出阶段，无需开展深入分析。要将转换配置为直接将相同记录传递至输出阶段，请选择“筛选完全匹配”。当输入数据包含多个相同行时，比较算法执行的计算会更少，映射运行速度会更快。

如果输入数据包含多个相同行，请选择此选项。如果输入数据包含的相同行不多，请勿选择此选项，因为转换运行的速度可能会减慢。

注意: 选择或取消选择此选项时，转换输出包含相同的记录数据。选择或取消选择此选项时，转换可能会向输出记录分配不同的链接得分和驱动程序得分。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

字段匹配分析示例

您是一家零售银行的数据管理员。 您接收一组账户记录，这些记录属于在过去七天开立了银行账户的客户。 您希望确认这组数据不包含重复记录。 您要设计一个用于在记录中搜索重复数据的映射。

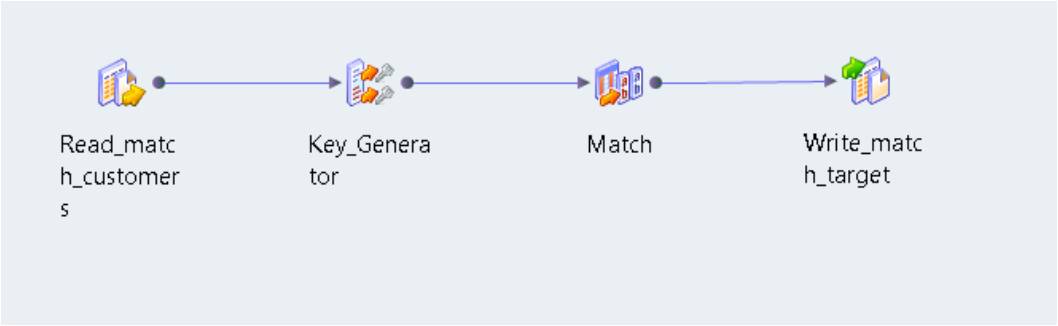
创建映射

创建用于在多个字段中查找重复数据的映射。

该映射执行以下任务：

- 读取数据源。
- 在源记录中添加组键值和序列标识符值。
- 分析记录中的字段数据。
- 将结果写入数据目标。

下图显示了 Developer 工具中的映射：



创建的映射包含以下对象：

对象名称	说明
Read_Match_Customers	数据源。 包含账户所有者详细信息，包括姓名、地址和账号。
Key_Generator	键生成器转换。 在源数据中添加组键值和序列标识符值。
匹配	匹配转换。 分析账户数据中的重复程度。
Write_match_target	数据目标。 包含字段分析的结果。

输入数据示例

数据集包含每位客户的账号、姓名、地址和雇主。 通过模型存储库中的数据集创建数据源。 将数据源添加到映射中。

以下数据段显示了客户账户数据示例：

CustomerID	Lastname	城市	省/自治区/直辖市	邮政编码
15954467	JONES	SCARSDALE	NY	10583
10110907	JONES	MINNEAPOLIS	MN	55437
19131127	JONES	INDIANAPOLIS	IN	46240
10112097	JONES	HOUSTON	TX	77036
19133807	JONES	PLANTATION	FL	33324
10112447	JONES	SCARSDALE	NY	10583
15952487	JONES	HOUSTON	TX	77002
10112027	JONES	OAKLAND	CA	94623

键生成器转换配置

配置键生成器转换时，连接要分析的数据源端口。 指定包含組键数据的端口。 如果记录不包括唯一标识符，请使用序列 ID 端口在记录中添加唯一标识符。

指定组键端口时，请考虑以下准则：

- 选择包含端口数据中定期重复的值的端口。 最佳。 选择可创建大小相似的组的端口。
- 选择与重复分析无关的端口。

在当前示例中，选择城市端口作为组键。 如果账户名称在一个城市中出现多次，账户可能包含重复数据。 如果账户名称在不同城市中出现多次，这些账户不一定重复。

提示：在选择组键端口前，对数据源运行列配置文件。 配置文件结果可以指出每个值在一个端口上出现的次数。

匹配转换配置

在映射中添加不可重用匹配转换，以执行字段分析。

完成以下任务以配置匹配转换：

1. 选择要执行的匹配分析的类型。
2. 将输入端口连接到转换。
3. 配置用以比较记录数据的策略。
4. 选择转换所创建的匹配输出数据的类型。
5. 将输出端口连接到数据目标。

选择匹配操作的类型

使用**匹配类型**视图上的选项选择匹配操作。 要比较账户数据，请配置转换以执行单源字段分析。

连接输入端口

将客户账户数据端口连接到匹配转换。

匹配转换使用预设的输入端口确定处理记录的顺序。 该转换使用序列标识符跟踪从输入端口到写入为输出的匹配对或群集的记录。 该转换使用组键对处理的记录进行排序。

将匹配转换上的以下预设端口连接到键生成器转换上的预设输出端口：

- SequenceID
- GroupKey

配置用于字段分析的策略

使用**策略**视图上的选项配置策略。 这些策略确定转换对记录数据执行的分析的类型。

创建以下策略：

- 创建使用编辑距离算法分析客户身份证号的策略。 选择 CustomerID_1 和 CustomerID_2 端口。
- 创建使用 Jaro 距离算法分析姓氏数据的策略。 选择 Lastname_1 和 Lastname_2 端口。

  注意: Jaro 距离算法对以不同字符开头的相似字符串应用附加罚值。 因此，Jaro 距离算法可以对 PATTON 和 PATTEN 应用高匹配得分，但对 BAYLOR 和 TAYLOR 则应用低一些的匹配得分。
- 创建使用反向汉明距离算法分析邮政编码数据的策略。 选择 Zip_1 和 Zip_2 端口。

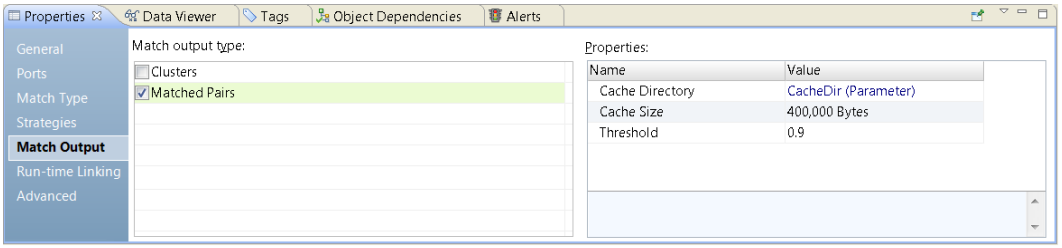
相关主题：

- “字段匹配算法” 页面上 414

选择匹配输出类型

使用**匹配输出**视图上的选项定义匹配分析结果的输出格式。

下图显示了单源字段分析的“匹配输出”视图：



可以配置转换以将输出记录整理到匹配对中。 由于转换成对返回记录，因此分析结果中不包含单独的记录。

连接输出端口

将匹配转换输出端口连接到映射中的数据目标。 选择包含要写入到数据目标中的记录数据的端口。

转换包含可识别每个匹配对中的记录的预设端口。 预设端口名称为 **RowId** 和 **RowId1**。 每个行 ID 值都标识输出数据中的一个记录。

行 ID 值对应于在匹配策略中选择的端口。 配置策略时，选择后缀为 _1 或 _2 的端口名称。 RowId 值标识包含后缀为 _1 的端口的记录。 RowId1 值标识包含后缀为 _2 的端口的记录。

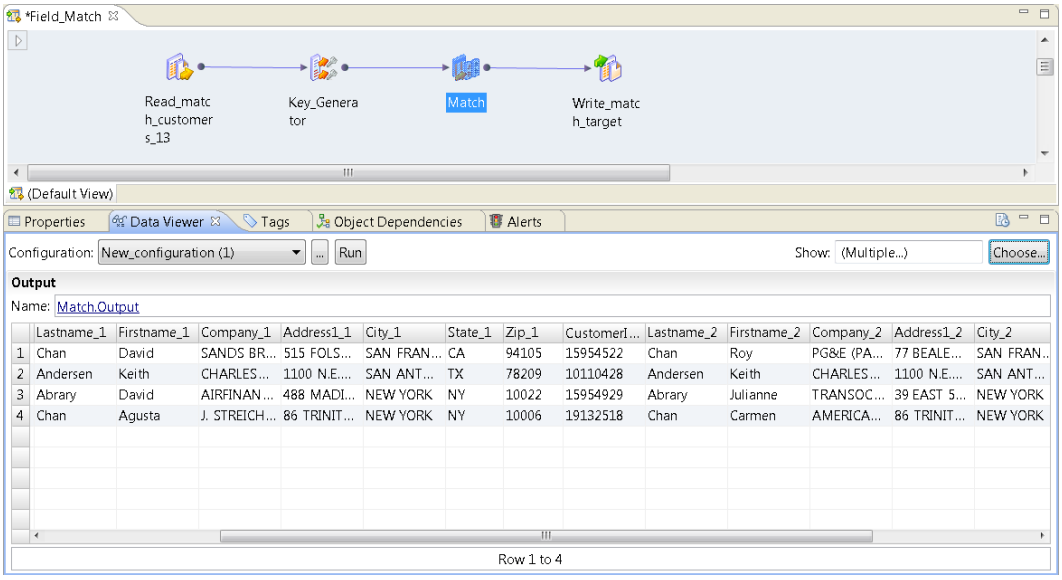
可以使用其他输出端口查看记录间的关系。 链接端口值和驱动程序端口值指出每个群集中的记录间的相似度范围。

在当前示例中，将所有端口连接到数据目标。 要查看端口上的输出数据，请运行数据查看器。

运行数据查看器

运行数据查看器可查看匹配分析的结果。 默认情况下，数据查看器显示匹配转换上的所有输出端口。 运行映射时，将用来自输出端口的数据更新数据目标。

下图显示了数据查看器中的输出数据：



数据查看器验证客户账户数据是否包含一个或多个重复记录。

考虑数据查看器中的以下数据：

- 该转换确定 Augusta Chan 和 Carmen Chan 的记录可能包含相同的信息，因为他们包含相同的姓氏和地址数据。 查看记录时，应判定记录在数据集中是否唯一。 但您会注意到这些记录共享一个共用的客户 ID 值。 由于客户 ID 列是数据集中的主键，因此您联系纽约办事处。 纽约办事处解决该错误。
- 该转换确定 Keith Anderson 的记录可能包含相同的信息。 查看记录时，验证两个记录是否代表同一个账户。 但您注意到记录的客户 ID 值并不相同。 由于客户账户必须有一个单独的 ID 值，因此您联系圣安东尼奥办事处。 圣安东尼奥办事处解决该错误。

结论

匹配分析的结果表明客户账户数据集至少包含一对重复记录。 您联系当地分公司以了解需要验证的账户。 您确认数据集中的其他记录唯一标识客户账户。

第 31 章

标识分析中的匹配转换

本章包括以下主题：

标识匹配分析

执行标识匹配分析可以在一个数据集中或者两个数据集之间找到相似或重复的标识。

两个或多个记录中有相似的标识可能说明记录重复。 或者，相似标识可能表明记录之间的关联性，例如共享的家庭标识或共享的雇主标识。

配置匹配转换以进行标识匹配分析时，在以下视图上设置选项：

- 匹配类型
- 策略
- 匹配输出

或者，在“参数”和“高级”视图上设置选项。

为标识匹配分析配置匹配转换时，将输入端口连接到标识策略中的所有首要必需字段。 大多数标识策略都包含首要字段。 某些策略还包含次要必需字段。 将输入端口连接到至少一个次要必需字段。

标识匹配分析的流程

下面的流程概括了配置匹配转换以进行标识匹配分析所要采取的步骤。 可以定义一个单独使用匹配转换或者使用匹配转换和其他转换的流程。

将匹配转换连接到上游数据对象之前，验证记录是否包含唯一的序列标识符值。 可以使用键生成器转换创建这些值。 执行标识匹配分析时，可以选择性地将输入数据整理成组。

在匹配转换中执行以下步骤：

1. 指定标识分析作为匹配类型，并指定数据源的数量。

   如果将转换配置为分析两个数据集，请选择一个主数据集。

   使用**匹配类型**视图设置数据源的类型和数量。

2. 标识用来存储索引数据的位置。 该转换可将索引数据写入到临时文件中，或者将索引数据保存到数据库表中。

   使用**匹配类型**视图指定索引数据存储。

3. 定义匹配分析策略。 选择填充和比较算法，并为算法分配一对列。

   填充指出要选择的列对。

   使用**策略**视图定义策略。

4. 指定转换生成匹配分析结果所用的方法。

5. 设置匹配阈值。 匹配阈值是可以将两个记录标识为彼此重复的最低得分。

   使用**匹配输出**视图选择输出方法和匹配阈值。

   注意: 可以在匹配转换或加权平均值转换中设置匹配阈值。 如果创建了匹配 Mapplet，请使用加权平均值转换。

标识匹配类型属性

您可以使用“匹配类型”视图指定匹配转换执行的分析类型，以及设置用于定义分析的属性。您可以指定单源分析或双源分析，还可为标识索引数据指定永久性数据存储。

应配置的属性取决于所选分析类型。其中许多选项是所有分析类型共用的。

共同属性

以下属性是所有分析类型共用的：

填充

标识转换使用的填充文件。填充文件包含生成索引键的键构建算法。

键级别

确定标识算法生成的键的数量。默认设置为标准。受限设置将导致键数量减少，准确性提高，但处理时间延长。扩展设置将导致键数量增加，准确性降低，但处理时间会缩短。

键类型

描述键字段包含的信息类型。标识分析可以针对人员姓名、组织和地址生成键。选择最能反映您在**键字段**属性上所指定列的键类型。

搜索级别

指示转换应用于匹配分析的搜索深度和搜索速度之间的平衡。搜索深度与返回的匹配项数量成反比。例如，**完全**选项返回的匹配数较少。

键字段

指定匹配转换用于生成索引键数据的列。 验证所选列是否包含在**键类型**属性上指定的信息类型。

索引目录

标识数据集成服务向其中写入当前转换的索引键数据的目录。默认情况下，该属性为空。如果未指定索引目录，数据集成服务将使用您在内容管理服务上设置的位置。

您可以输入目录的路径，也可以使用参数识别该目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。

缓存目录

指出数据集成服务在标识匹配分析的索引创建阶段向其中写入临时数据的目录。更新该属性为当前转换的数据指定位置。默认情况下，该属性为空。如果未指定缓存目录，数据集成服务将使用您在内容管理服务上设置的位置。

您可以输入目录的路径，也可以使用参数识别该目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。

缓存大小

确定数据集成服务分配给标识索引创建的系统内存量。默认值为 400,000 字节。

如果索引创建操作生成的数据较多，数据集成服务会将超出的数据写入缓存目录。如果操作所需的内存超出系统内存和文件存储可以提供的内存，映射将失败。

注意: 如果输入 65536 或更高的值，转换将以字节为单位读取值。如果输入低一些的值，转换将以 MB 为单位读取值。

双源属性

将转换配置用于双源分析时，您应设置共同属性和以下属性：

主数据集

标识包含主数据的数据源。指定双源分析中的主数据集。

永久性数据存储属性

将转换配置为使用永久性索引数据存储时，您应设置共同属性和以下属性：

持久性方法

指定转换是否使用来自映射数据源的索引数据更新当前的索引表。选择以下选项之一：

- 使用新的 ID 更新数据库。
  转换会将所有不会导致索引数据中出现重复序列标识符的行添加到索引数据中。转换不会更新索引中的当前行。

  默认情况下，如果选择此选项，则转换将执行匹配分析。您可以使用“匹配进程”选项来启用或禁用匹配分析。
- 不更新数据库。
  转换不会使用映射数据源的索引数据更新索引表。

  如果选择此选项，则转换将执行匹配分析。
- 从数据库中删除 ID。
  如果索引表中的行与映射源数据中的行共享序列标识符，则转换会删除索引表中的这些行。

  如果选择此选项，则转换不会执行匹配分析。

- 更新数据库中的当前 ID。
  如果索引表中的行与映射源数据中的行共享序列标识符，则转换会将索引表中的行替换为映射源数据中的行。转换不会向索引添加行。

  默认情况下，如果选择此选项，则转换将执行匹配分析。您可以使用“匹配进程”选项来启用或禁用匹配分析。

默认的持久性方法为**使用新的 ID 更新数据库**。

匹配进程

确定当前转换是否执行标识分析。

在“持久性方法”属性上选择的选项决定了“匹配进程”属性上的选项。

数据库连接

标识包含索引表的数据库。

永久性存储

标识指定数据库中的索引表。

相关主题：

- “永久性索引案例研究” 页面上 432
- “持久性方法参数” 页面上 427

索引目录和缓存目录属性

索引目录和缓存目录存储匹配转换在标识分析过程中生成的临时数据。

如果没有将转换配置为将索引数据写入数据库表，数据集成服务会将数据写入索引目录。如果标识分析所需的内存量超过缓存大小指定的量，数据集成服务会将数据写入缓存目录。默认情况下，“匹配类型”视图上的属性为空。如果未指定索引目录或缓存目录，数据集成服务会从内容管理服务读取目录路径。

在“匹配类型”视图上指定索引目录或缓存目录时，请输入数据集成服务计算机上的本地路径。可以输入完全限定的路径或相对路径。如果输入相对路径，请以句点开头。此路径相对于数据集成服务计算机上的 tomcat/bin 目录。

下表展示了关于缓存目录或索引目录属性的相对路径，并指出了该属性代表的完全限定路径。

相对路径	完全限定的路径
./ch	[Informatica_installation_directory]/tomcat/bin/ch

可以将索引目录属性和缓存目录属性配置为标识相同的目录。如果在每个属性上指定相同的目录，数据集成服务会在指定的目录中创建目录。数据集成服务会为索引数据创建 index 目录，为缓存数据创建 cache 目录。如果在每个属性上指定不同的目录，数据集成服务会将数据写入指定的目录。

数据集成服务会在运行映射后从目录中删除索引文件和缓存文件。数据集成服务不会删除目录。如果匹配转换可识别这些目录，数据集成服务会在其他映射中重复使用这些目录。

内容管理服务属性

内容管理服务为索引目录和缓存目录指定以下默认位置：

- ./identityIndex.标识索引数据的默认目录。
- ./identityCache.标识缓存数据的默认目录。

如果未针对匹配转换设置这些属性，数据集成服务会在运行标识匹配映射时创建默认目录。数据集成服务会在 tomcat/bin 目录中创建目录。

持久性方法参数

在标识匹配分析中选择持久性数据索引时，可以使用参数来标识持久性方法。 使用字符串来定义参数值。

下表列出了可以在**匹配类型**选项卡上选择的持久性方法以及可以为这些方法设置的参数值：

持久性方法	参数
使用新的 ID 更新数据库	ignore
不更新数据库	addNone
从数据库中删除 ID	remove
更新数据库中的当前 ID	update

参数值区分大小写。

相关主题：

- “标识匹配类型属性” 页面上 424

标识匹配策略

“策略”视图列出您为标识数据定义的策略。策略决定转换度量标识索引中的数据之间的相似度和差异的方式。

标识匹配算法

匹配转换包括用于比较标识索引中的数据值的预定义标识算法。 选择最能代表数据集中的标识数据类型的算法。

下表介绍了这些算法并标识为每个算法选择的输入：

标识算法	说明
地址	标识共享某个地址的记录。 该算法需要以下首要输入： - 地址 该算法不需要次要输入。
联系人	标识单个组织位置中共享某个联系人的记录。 该算法需要以下首要输入： - Person_Name - Organization_Name - Address_Part1 该算法不需要次要输入。

标识算法	说明
法人实体	标识共享企业标识数据的记录。 或者，可以选择该算法来分析地址和电话数据。 该算法需要以下首要输入： - Organization_Name 该算法不需要次要输入。
除法	标识共享组织内某个办公地点的记录。 该算法需要以下首要输入： - Organization_Name - Address_Part1 该算法不需要次要输入。
家庭	标识属于一个家庭的个人。 分析姓名、地址和电话号码数据。 该算法需要以下首要输入： - Person_Name 该算法需要以下次要输入之一： - Address_Part1 - Telephone_Number
字段	标识共享选定端口上的数据的记录。 该算法不指定任何必需输入。 选择可能包含重复标识数据的一个或多个端口。
住户	标识属于一个住户的个人。 分析姓名数据和地址数据。 该算法需要以下首要输入： - Person_Name - Address_Part1
个人	标识重复的个人。 分析姓名、生日和个人标识数据，如社会保障号、账号和机动车牌号。 该算法需要以下首要输入： - Person_Name 该算法需要以下次要输入之一： - 日期 - ID
组织	标识共享组织数据的记录。 该算法需要以下首要输入： - Organization_Name 该算法不需要次要输入。
人名	标识共享个人相关信息的记录。 该算法需要以下首要输入： - Person_Name 该算法不需要次要输入。
居民	标识位于某个地址的重复个人。 或者，可以将此策略配置为分析个人标识数据。 该算法需要以下首要输入： - Person_Name - Address_Part1 该算法不需要次要输入。

标识算法	说明
广义联系人	标识共享组织内某个联系人的记录。 该算法需要以下首要输入： - Person_Name - Organization_Name 该算法不需要次要输入。
广义住户	标识属于同一住户的个人。 该算法需要以下首要输入： - Address_Part1 该算法不需要次要输入。

标识匹配策略属性

配置每个标识策略的属性。

配置标识策略时，可以配置以下策略属性：

填充

确定要应用于标识分析的填充。 填充包含针对特定区域设置和语言的键构建算法。

匹配级别

确定搜索质量和搜索速度之间的平衡。 搜索速度与返回的匹配数成反比。 使用松散设置的搜索返回的匹配数较少，使用保守设置的搜索返回的匹配数较多。

标识匹配输出选项

“匹配输出”视图包括用于指定输出数据格式的选项。 可以将转换配置为将记录写入群集或匹配对中。 还可以将转换配置为在对永久性索引数据存储执行标识分析时，包括或排除不同类别的标识。

可以在**匹配输出类型**区域和**属性**区域配置这些选项。

匹配输出类型

“匹配输出”视图包括用于指定输出数据格式的选项。 可以将转换配置为将记录写入群集或匹配对中。

请选择以下匹配输出类型之一：

最佳匹配

写入主数据集中的每个记录以及表示第二个数据集中的最佳匹配的记录。匹配操作在第二个数据集中为主记录选择匹配得分最高的记录。如果两个或多个记录返回最高得分，则匹配操作将在第二个数据集中选择第一个得分最高的记录。最佳匹配将每一对记录写入到一个行中。

为双源分析配置转换时，可以选择**最佳匹配**。

群集 - 最佳匹配

写入代表同一个数据集中一个记录与另一个记录之间或者两个数据集之间最佳匹配的群集。 两个记录之间的匹配得分必须满足匹配阈值。 如果某个记录代表其他多个记录的最佳匹配项，最佳匹配群集可以包含超过两个记录。

可以在任何类型的标识分析中选择**群集 - 最佳匹配**。

注意: 在**群集 - 最佳匹配**模式下，您选择的索引数据存储方法可影响群集输出的内容。连接到索引表的转换可针对临时文件中的相同记录创建与存储索引数据的转换不同的群集。 索引数据存储方法不会影响转换为记录对生成的匹配得分。

群集 - 全部匹配

写入与满足匹配阈值的得分相匹配的记录群集。 每个记录必须至少与群集中另一个记录相匹配。

可以在任何类型的标识分析中选择**群集 - 全部匹配**。

匹配对

写入与满足匹配阈值的得分相互匹配的所有记录对。 转换将每对写入一行，然后将每对的匹配得分添加到每行。 如果记录与多个其他记录相匹配，转换会为每个记录对写入一行。

可以在任何类型的标识分析中选择**匹配对**。

匹配输出属性

“匹配输出”视图包含指定高速缓存行为和匹配得分阈值的属性。您还可以使用这些属性确定转换如何选择数据存储记录进行分析，并将数据存储记录作为输出写入。

选择匹配输出类型后，配置以下属性：

缓存目录

指定数据集成服务在标识匹配分析过程中将临时数据写入到的目录。当匹配分析生成的数据超出可用系统内存量时，数据集成服务会将临时文件写入此目录。数据集成服务会在运行映射后删除这些临时文件。

您可以在属性上输入目录路径，也可以使用系统参数标识目录。指定数据集成服务计算机上的本地路径。数据集成服务必须能够写入此目录。默认值为 CacheDir 系统参数。

缓存大小

确定数据集成服务分配给标识匹配分析的系统内存量。默认值为 400,000 字节。

如果匹配分析生成的数据较多，数据集成服务会将超出的数据写入缓存目录。如果匹配分析所需的内存超出系统内存和文件存储可以提供的内存，映射会失败。

注意: 如果输入 65536 或更高的值，转换将以字节为单位读取值。如果输入低一些的值，转换将以 MB 为单位读取值。

匹配

当转换从数据库表读取索引数据时标识要分析的记录。 使用**匹配类型**视图上的选项标识索引表。

默认情况下，转换会分析数据源和索引数据库表中的所有记录。配置匹配属性，以指定用于重复项分析的记录子集。

输出

配置转换以读取索引数据库表时，筛选转换作为输出写入的记录。 使用**匹配类型**视图上的选项标识索引表。

默认情况下，匹配转换将数据源和索引数据库表中的所有记录作为输出写入。 如果不需要查看输入数据中的所有记录，请配置**输出**属性。

阈值

设置将两个记录标识为可能彼此重复的最低匹配得分。

可以将参数分配给阈值。 可将小数值设置为介于 0 到 1 之间。

匹配属性配置

使用**匹配输出**视图上的**匹配**属性指定转换如何选择输入数据以进行分析。 配置匹配转换以读取索引数据的永久性存储时，应配置此属性。 “匹配”属性可细化在**匹配类型**视图上设置的选项。

可以配置匹配属性以执行以下类型的分析：

将数据源记录与索引数据记录进行比较。

要在数据源与索引数据表之间查找重复记录，可选择**独占**。

如果选择“独占”选项，则匹配转换会将数据源记录与索引数据存储进行比较。 转换不分析数据源或数据存储中的记录。

如果知道索引数据存储不包含重复记录，并且知道数据源不包含重复记录，则选择**独占**。

将数据源记录与索引数据记录进行比较，并将数据源记录相互进行比较。

要在数据源中查找重复项以及在数据源与索引表之间查找重复项，请选择**部分**。

转换将数据源记录与索引数据存储进行比较。 转换还将数据源内的记录相互进行比较。

如果知道索引数据存储不包含重复记录，但未对数据源执行任何重复分析，则选择**部分**。

将数据源和索引表中的所有记录作为一个数据集进行比较。

要查找数据源与索引表之间的重复项，以及在数据源内部和索引表内部查找重复项，请选择**完整**。 默认选项为“完整”。

转换将数据源和数据存储作为一个数据集进行分析，并比较该数据集内的所有记录数据。

如果无法验证数据集是否没有重复记录，则选择**完整**。

输出属性配置

使用**匹配输出**视图上的**输出**属性筛选转换作为输出写入的记录。如果指定索引数据包并选择群集化输出格式，则配置该属性。筛选记录，以将输出限制为包含一个或多个来自数据源的记录的群集。

可以采用以下方式筛选输出数据：

写入包括来自数据源或索引表的记录的每个群集。

选择**所有行**。转换将写入包含至少一个来自数据源或索引数据存储的记录的每个群集。默认值为“所有行”。

由于一个群集可以包含一个记录，因此输出将包含所有记录。

写入包括来自数据源的记录的每个群集

选择**新行和关联行**。转换将写入包含至少一个来自数据源的记录的每个群集。

由于一个群集可以包含一个记录，因此输出将包含数据源中的所有记录。群集还可以包含来自索引表的记录。

写入来自数据源的每个群集

选择“仅新行”。转换将写入包含来自数据源的记录的群集。输出不包含来自索引表的任何记录。

标识匹配高级属性

转换包含确定执行实例数、转换是否分析相同行以及日志数据跟踪级别的高级属性。

您可以配置以下高级属性：

执行实例

指定数据集成服务在运行时试图为当前转换创建的线程数。 如果您替代包含转换的映射上的“最大并行数”运行时属性，则数据集成服务则会考虑“执行实例”值。 默认“执行实例”值为“自动”。

数据集成服务会考虑多个因素来确定分配给转换的线程数。 首要因素为“执行实例”值以及映射和域中关联应用程序服务上的值。

运行映射的数据集成服务将读取以下值来确定用于转换的线程数：

- 数据集成服务上的*最大并行数*值。 默认值为 1。
- 您在映射级别设置的任何*最大并行数*值。 默认为“自动”。
- 转换上的*执行实例*值。 默认为“自动”。

如果您替代映射级别的“最大并行数”值，则数据集成服务会尝试使用属性间的最低值来确定线程数。

如果您使用映射级别的默认“最大并行数”值，则数据集成服务会忽略“执行实例”值。

设置执行实例数量时，请注意以下规则和准则：

- 多个用户可能会在数据集成服务上运行并发映射。要计算正确的线程数，请使用数据集成服务可以访问的中央处理单元数量除以并发的映射数量。
- 如果使用默认的“执行实例”值和默认的“最大并行数”值，则转换操作不可分区。

筛选完全匹配

确定转换是否会对输入数据中的相同记录对应用匹配策略中的比较算法。

转换发现相同记录对时，该算法不需要分析记录之间的相似度。转换可直接将记录传递至输出阶段，无需开展深入分析。要将转换配置为直接将相同记录传递至输出阶段，请选择“筛选完全匹配”。当输入数据包含多个相同行时，比较算法执行的计算会更少，映射运行速度会更快。

如果输入数据包含多个相同行，请选择此选项。如果输入数据包含的相同行不多，请勿选择此选项，因为转换运行的速度可能会减慢。

注意: 选择或取消选择此选项时，转换输出包含相同的记录数据。选择或取消选择此选项时，转换可能会向输出记录分配不同的链接得分和驱动程序得分。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

永久性索引案例研究

假设您在一家开设了多个分行的零售银行担任数据管理者一职，负责管理所有分行的客户帐户记录的主集，则您可以使用一组索引数据库表来验证客户帐户数据库是否不包含冗余记录或重复记录。

要创建并管理索引数据存储，请执行以下操作：

- 创建数据存储。
- 用各个分行提供的最新数据更新数据存储。

  您可以将帐户数据添加到数据存储，或者也可以更新数据存储中的当前数据。
- 从数据存储中删除过时的记录。

您明白每个操作都可能在数据存储中创建重复记录，因此决定开发一个策略，以便在将分行数据添加到主数据存储数据之前对数据进行分析。您可以使用标识匹配分析来分析分行数据，以及验证数据是否未在数据存储中创建重复标识。您可以对匹配转换配置永久性索引选项，以分析分行数据和数据存储。

开发用于永久性索引数据管理的策略

作为数据管理者，您可以定义指明客户帐户数据存储不能包含重复标识的业务规则。您可以设计一个标识匹配映射，以便在将分行数据添加到数据存储之前对暂存数据库中的数据进行分析。

在以下情况下，将分行数据添加到数据存储的操作会创建重复标识：

- 分行数据包含重复标识。
- 分行数据包含同时也存在于索引中的标识。
- 分行数据包含数据存储中某个标识的较新版本，而该较新版本与索引中的其他标识匹配。

将暂存数据库与数据存储相比较时，请选择可反映分行数据的重复记录状态的永久性索引选项。更新数据存储之前，您可以决定是否要将分行数据与索引数据相比较。

注意: 您可以对部分选项启用和禁用匹配分析。启用匹配分析时，您可以分析映射数据或将索引数据存储与映射数据相比较。如果不需要比较数据，则禁用匹配分析。您还可以使用“匹配输出”选项卡上的“匹配”属性，以便在匹配分析时包含或排除特定的数据。

将映射数据源与索引数据存储相比较

要将映射输入数据与索引数据存储相比较而不对数据存储进行任何更改，请选择以下选项：

- 不更新数据库

映射会将输入数据与索引数据存储相比较，但不会在索引数据存储中添加、删除或更新任何数据。

如果选择此选项，则您不能禁用标识匹配分析。

由于未更新索引数据，因此您不会在存储中创建重复行。从“匹配输出”选项卡上的“匹配”属性中，选择满足数据项目的当前需求的选项。例如，选择**完整**选项。**完整**选项可验证映射数据是否不包含重复项，并验证映射数据是否未将重复项添加到数据存储。

注意: 更新数据存储之前，您可以使用此选项将映射数据与数据存储相比较。如果映射输出指示映射数据未将重复项添加到数据存储，则再次运行该映射。再次运行该映射时，请选择此选项以更新数据库。

创建数据存储以及向数据存储添加行

要创建数据存储或向数据存储添加映射数据中的行，请选择以下选项：

- 使用新的 ID 更新数据库

如果某个行不与数据存储中的行共享序列标识符，则映射会将该行添加到数据存储。映射不会覆盖索引表中的任何行。如果指定空的数据库表，则映射会将所有映射索引数据写入到表中。

如果选择此选项，则可以启用或禁用标识匹配分析。此选项默认启用匹配分析。

由于您未更新索引行，因此请从“匹配输出”选项卡上的“匹配”属性中选择**独占**选项或**部分**选项。如果已在先前的进程中验证了映射数据行的唯一性，请使用**独占**选项。

更新数据存储中的行

要用映射数据更新数据存储中的当前行，请选择以下选项：

- 更新数据库中的当前 ID

如果数据存储中的当前记录与映射数据中的记录共享序列标识符，则映射会更新该记录。映射不会将任何行添加到索引表。

如果选择此选项，则可以启用或禁用标识匹配分析。此选项默认禁用匹配分析。

由于您未将索引行添加到索引表，因此请从“匹配输出”选项卡上的“匹配”属性中选择**完整**选项。

注意: 更新数据存储中的行时，映射源数据和数据存储之间预计会出现重复项。选择**完整**选项可验证您添加到存储中的标识数据是否不与存储中的当前的数据匹配。

从数据存储中删除行

要从数据存储中删除行，请选择以下选项：

- 从数据库中删除 ID

如果数据存储中的行与映射数据中的记录共享序列标识符，则映射会删除该行。

如果选择此选项，则可以启用或禁用标识匹配分析。此选项默认禁用匹配分析。

注意: 从数据存储中删除数据后，存储中行与行之间的关系随即更改。如果存储包含重复标识，则您可以删除群集中某个驱动程序记录或链接记录的数据。或者，您也可以删除匹配对中的最佳匹配项的数据。再次运行映射时，映射可能会生成不同的群集或重复对。如果从不包含重复记录的数据存储中删除行，则您无法更改记录的重复状态。如果您在删除行之后运行映射，映射将为保留在数据集中的标识生成相同的匹配得分。

标识匹配分析示例

假设您在一家在多个城市都设有开发中心的软件企业担任人力资源主管一职，企业职员的人事记录存储在总部的数据库中，各个开发中心会定期雇用职员，并将所雇用职员的人事数据发送给您。

您可以将人事记录添加到电子表格文件，并使用文件数据更新员工数据库。您担心当前文件可能包含重复标识。

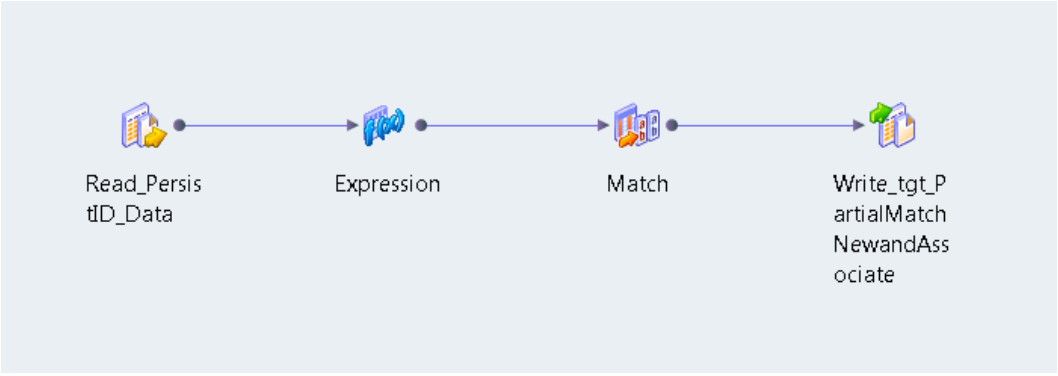
您要设计一个映射，用于对员工记录执行标识分析。 您可以配置匹配转换，以在电子表格文件中搜索重复标识。您还必须验证文件数据是否未导致主数据库中出现任何重复的员工数据。您可以配置匹配转换，以将文件数据与您存储的组织员工主数据相比较。

由于数据库是主数据集，因此您应将职员记录的索引数据存储在永久性数据存储中。

创建映射

创建用于查找重复标识的映射。 该映射读取数据源、在源记录中添加序列标识符、执行标识分析并将结果写入数据目标。

下图显示了 Developer 工具中的映射：



创建的映射包含以下对象：

对象名称	说明
Read_PersistID_Data	数据源。 包含员工姓名和详细信息。
表达式	表达式转换。 在源数据中添加序列标识符值。
匹配	匹配转换。 分析源数据标识中的重复程度。
Write_tgt_PartialMatchNewandAssociate	数据目标。 包含标识分析的结果。

注意: 该映射不使用键生成器转换。 在标识匹配分析中，键生成器转换是可选项。

输入数据示例

职员文件包含员工姓名、员工工作城市和员工的指定职责。 通过模型存储库中的职员文件创建数据源。 将数据源添加到映射中。

下面的数据段显示了职员文件中的员工数据示例：

名称	城市	指定
Chaithra	Bangalore	SE
Ramanan	Chennai	SSE
Ramesh	Chennai	SSE
Ramesh	Chennai	Lead
Sunil	Bangalore	Principal
Venu	Hyderabad	Principal
Harish	Bangalore	SE
Sachin	Bangalore	SSE

表达式转换配置

配置表达式转换时，连接要在映射输出中包括的所有数据源端口。 将端口作为传递端口连接。 创建表达式，用以将序列标识值添加到端口中。

下面的表达式将创建变量 *Init3*，并将每个序列标识符加上整型值 1267：

Init3+1267

下表介绍了表达式转换上读取员工数据源的端口：

名称	端口类型	端口组
SEQID	bigint	仅输出
名称	string	传递
城市	string	传递
指定	string	传递
Init3	Integer	变量

匹配转换配置

在映射中添加不可重用匹配转换，以执行标识分析。

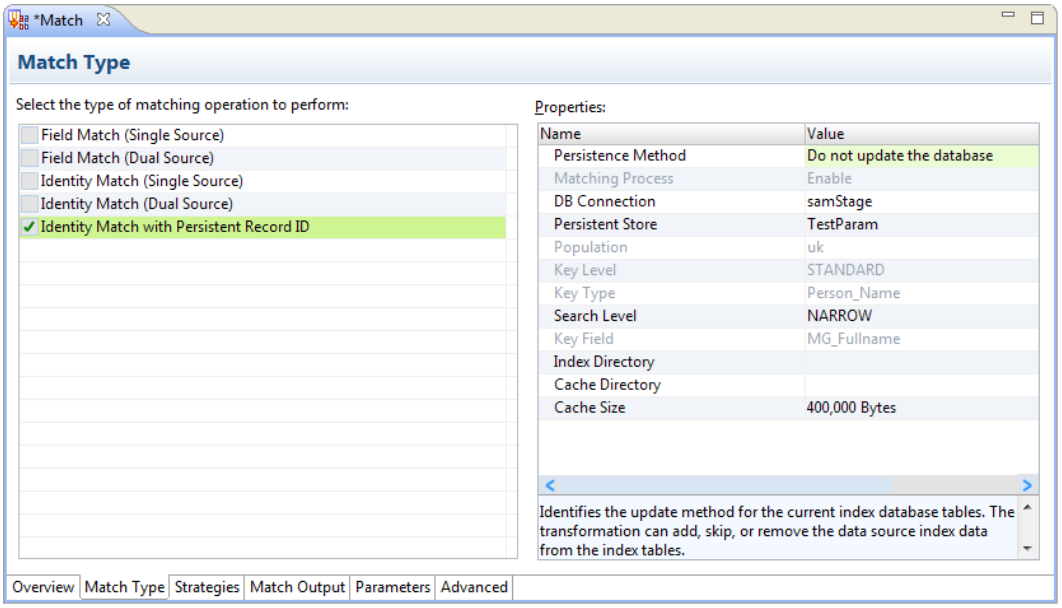
执行以下任务以配置转换：

1. 选择要执行的匹配分析的类型。
2. 将输入端口连接到转换。
3. 配置用以比较记录数据的策略。
4. 选择转换所创建的匹配输出数据的类型。
5. 将输出端口连接到数据目标。

选择匹配操作的类型

使用“匹配类型”视图上的选项选择匹配操作。

下图显示了“匹配类型”视图：



要将数据源中的索引数据与主数据集中的索引数据相比较，请选择**使用永久性记录 ID 进行标识匹配**。更新持久性方法，以便匹配分析不会将数据添加到索引表中。您可以在审阅映射结果后决定是否要更新索引表。

使用**数据库连接**选项标识包含索引数据的数据库。使用**永久性存储**选项选择索引表。

注意: 匹配转换将从索引数据库表中的元数据读取标识填充、键级别、键类型和键字段属性的值。 这些值与创建了索引数据存储的转换中的相应属性匹配。

连接输入端口

将输入数据端口连接到转换。确保输入端口名称、端口顺序、数据类型和精度必须与创建了数据存储的转换的端口配置匹配。

匹配转换使用预设的输入端口确定处理记录的顺序。该转换使用序列标识符跟踪从输入端口到写入为输出的匹配对或群集的记录。该转换使用组键对处理的记录进行排序。

将预设的端口连接到表达式转换上的以下端口：

- SequenceID。从表达式转换连接到 SEQID 端口。
- GroupKey。从表达式转换连接到城市端口。

配置标识分析的策略

使用“策略”视图上的选项配置策略。策略决定转换对记录数据执行的分析的类型。

选择 Person_Name 算法用于记录数据。为分析选择“名称”输入端口。由于转换将创建端口数据的副本，因此选择 Name_1 port 和 Name_2 端口。

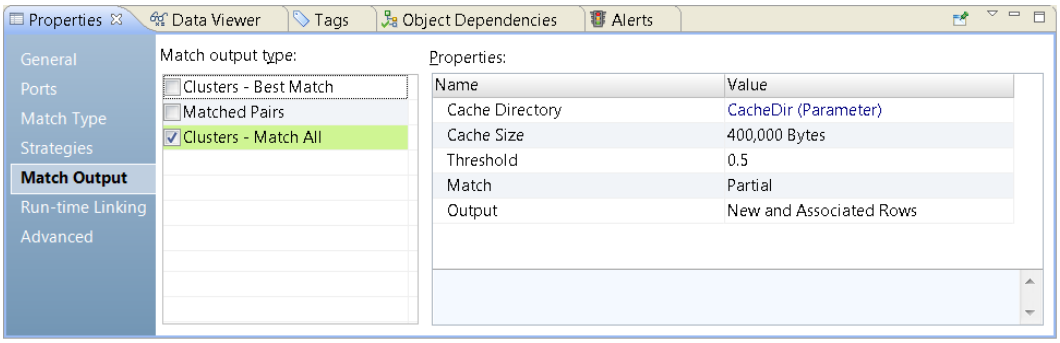
相关主题：

- “标识匹配算法” 页面上 427

选择匹配输出类型

使用“匹配输出”视图上的选项定义匹配分析结果的输出格式。

下图显示了单源标识分析的“匹配输出”视图：



可以配置转换以整理群集中的输出记录。 每个群集包含根据您指定的匹配属性与至少一个其他记录相匹配的所有记录。 匹配属性决定转换如何将数据源记录与索引记录进行比较。

下表介绍了为分析职员记录数据而指定的匹配属性选项：

匹配属性	选项	选项说明
匹配	部分	转换将数据源记录与索引数据存储进行比较。 转换还将数据源内的记录相互进行比较。
输出	新行和关联行	转换将写入包含至少一个来自数据源的记录的每个群集。 群集可以包括来自索引数据存储的记录。 由于群集可以包含一个记录，因此输出将包含数据源中的所有记录。

相关主题：

- “匹配属性配置” 页面上 431
- “输出属性配置” 页面上 431

连接输出端口

将匹配转换输出端口连接到映射中的数据目标。 选择包含要写入到数据目标中的记录数据的端口。

该转换包括一系列用于群集化数据的预设端口。 选择可指示记录重复状态并标识存储每个记录的数据源的预设端口。

以下端口包含可用来查找重复记录并确定源或记录的数据：

- **ClusterSize** 端口指出群集中的记录数量。 如果记录所属的群集的大小大于 1，则转换会将该记录视为其他记录的重复项。
- **ClusterID** 端口标识记录所属的群集。 使用 ClusterID 数据查找与当前记录重复的记录。
- **PersistenceStatus** 端口使用代码值来描述映射源中的索引数据与数据存储中的索引数据之间的关系。
- **PersistenceStatusDesc** 端口返回 PersistenceStatus 端口代码上的值的文本说明。

可以使用其他端口查看群集记录之间的关系。 链接端口值和驱动程序端口值指出每个群集中的记录间的相似度范围。

在当前示例中，将所有端口连接到数据目标。 要查看端口上的输出数据，请运行数据查看器。

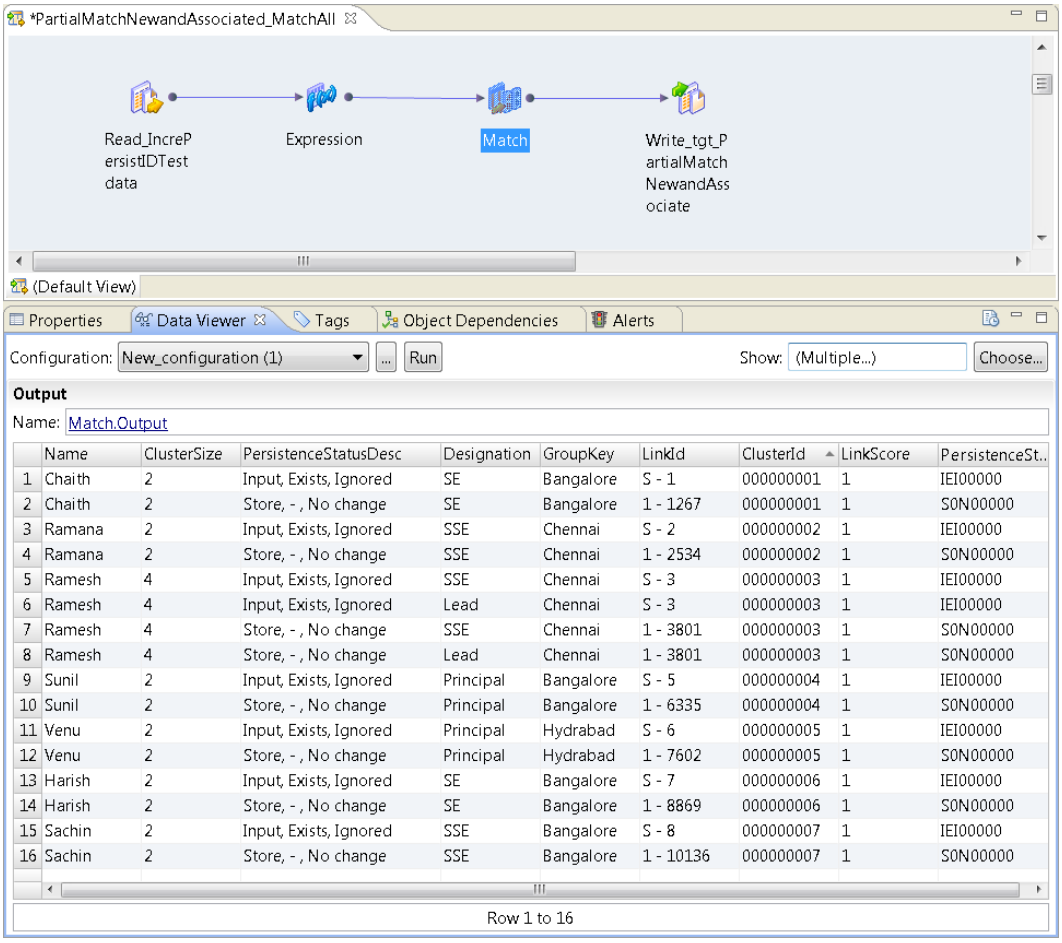
相关主题：

- “持久性状态代码和持久性状态说明” 页面上 408
- “状态代码值和状态说明值” 页面上 408

运行数据查看器

运行数据查看器可查看匹配分析的结果。默认情况下，数据查看器显示匹配转换上的所有输出端口。运行映射时，将用来自输出端口的数据更新数据目标。

下图显示了数据查看器中的输出数据：



	Name	ClusterSize	PersistenceStatusDesc	Designation	GroupKey	LinkId	ClusterId	LinkScore	PersistenceSt..
1	Chaith	2	Input, Exists, Ignored	SE	Bangalore	S - 1	000000001	1	IEI00000
2	Chaith	2	Store, - , No change	SE	Bangalore	1 - 1267	000000001	1	S0N00000
3	Ramana	2	Input, Exists, Ignored	SSE	Chennai	S - 2	000000002	1	IEI00000
4	Ramana	2	Store, - , No change	SSE	Chennai	1 - 2534	000000002	1	S0N00000
5	Ramesh	4	Input, Exists, Ignored	SSE	Chennai	S - 3	000000003	1	IEI00000
6	Ramesh	4	Input, Exists, Ignored	Lead	Chennai	S - 3	000000003	1	IEI00000
7	Ramesh	4	Store, - , No change	SSE	Chennai	1 - 3801	000000003	1	S0N00000
8	Ramesh	4	Store, - , No change	Lead	Chennai	1 - 3801	000000003	1	S0N00000
9	Sunil	2	Input, Exists, Ignored	Principal	Bangalore	S - 5	000000004	1	IEI00000
10	Sunil	2	Store, - , No change	Principal	Bangalore	1 - 6335	000000004	1	S0N00000
11	Venu	2	Input, Exists, Ignored	Principal	Hydrabad	S - 6	000000005	1	IEI00000
12	Venu	2	Store, - , No change	Principal	Hydrabad	1 - 7602	000000005	1	S0N00000
13	Harish	2	Input, Exists, Ignored	SE	Bangalore	S - 7	000000006	1	IEI00000
14	Harish	2	Store, - , No change	SE	Bangalore	1 - 8869	000000006	1	S0N00000
15	Sachin	2	Input, Exists, Ignored	SSE	Bangalore	S - 8	000000007	1	IEI00000
16	Sachin	2	Store, - , No change	SSE	Bangalore	1 - 10136	000000007	1	S0N00000

数据查看器验证文件是否包含与主数据集中的数据重复的记录。

考虑数据查看器中的以下数据值：

- 数据集中的每个记录都属于包含两个或多个记录的群集。因此，每个记录至少与一个其他记录重复。转换将群集大小 1 分配给唯一记录。数据源不包含主数据集所不包含的任何记录。
- **PersistenceStatusDesc** 数据标识记录源，并指出匹配转换是否将记录添加到索引表中。该列指出每个输入记录都存在于主数据集中。该转换不会向主数据索引中添加数据。

结论

匹配分析的结果表明，职员记录文件不包含主数据集所不包含的任何记录。永久性状态说明指出映射使用来自数据源的任何数据更新索引表。放弃职员记录文件。

从地区办事处收到其他更新时，可以另外创建一个文件并将其与主数据集进行比较。 可以重新使用映射和索引表。 由于在数据库表中存储主数据集的索引数据，因此无需重新生成索引数据。

第 32 章

规范器转换

本章包括以下主题：

规范器转换概览

规范器转换是将一个源行转换为多个目标行的主动转换。当规范器转换收到包含多次出现数据的行时，会为多次出现的数据的每个实例返回一行。

规范器转换将多次出现的数据解析到单独的输出行。例如，关系源行可能包含四个季度的销售数据。规范器转换将为出现的每个销售数据生成单独的输出行。

定义规范器转换时，可以配置说明源数据结构的输入行层次结构， 也可以定义转换输入层次结构中的记录。记录即一组字段的容器。当一组字段在源数据中多次出现时，可定义记录。输入层次结构决定可以如何配置转换输出组。

规范器转换可转换来自关系表或平面文件源的数据。

多次出现的字段

当某个字段在源数据中多次重复时，可以在输入行层次结构中将该字段定义为多次出现的字段。每次出现源中多次出现的字段或字段组时，规范器转换可返回单独的行。

源行可能包含各商店四个季度的销售数据：

商店	销售(1)	销售(2)	销售(3)	销售(4)
商店 1	100	300	500	700
商店 2	250	450	650	850

在定义规范器输入层次结构时，可以将四个销售字段合并成一个多次出现的字段。定义字段名称（例如 Qtr_Sales），并将其配置为在源中出现四次。

当输出组包含商店数据和销售数据时，规范器转换将针对每个 Store 和 Qtr_Sales 组合返回一行。输出行包含标识输出行中的 Qtr_Sales 实例的索引。

该转回将返回以下行：

商店	Qtr_Sales	Qtr (GCID)
商店 1	100	1
商店 1	300	2
商店 1	500	3
商店 1	700	4
商店 2	250	1
商店 2	450	2
商店 2	650	3
商店 2	850	4

当输出组包含单次出现的列和多次出现的列时，规范器将在每个输出行中针对单次出现的列将返回重复数据。例如，对于每个 Qtr_Sales 实例将重复出现商店 1 和商店 2。

源行可能包含多个级别的多次出现数据。您可以将规范器转换配置为，根据您定义输入层次结构的方式在每个级别返回单独的行。

生成的列 ID

对于多次出现的字段的每个实例，规范器转换将返回一个生成列 ID (GCID) 输出端口。

生成的列 ID 端口是多次出现数据的实例的索引。例如，如果某个字段在源记录中出现四次，Developer 工具会基于多次出现数据出现在行中的具体实例，在生成的列 ID 端口中返回值 1、2、3 或 4。

多次出现的记录

可以定义规范器转换源数据中多次出现的记录。记录即字段组。在需要定义多次出现的源字段组时，可在规范器转换中定义记录。

多次出现的记录示例

以下“客户”行包含客户信息及家庭地址信息和业务地址信息：

```
CustomerID
FirstName
LastName
Home_Street
```

```
Home_City
Home_State
Home_Country
Business_Street
Business_City
Business_State
Business_Country
```

在配置规范器转换时，可以定义包含客户字段和多次出现的地址记录的输入结构。地址记录出现两次。配置规范器转换输出组时，可向 CustomerID、FirstName 和 LastName 字段以外的不同目标返回地址记录。

以下示例显示了包含多次出现的地址记录的输入结构：

```
CustomerID
FirstName
LastName
Address (occurs twice)
   Street
   City
   State
   Country
```

子记录是指记录内的记录。在定义记录和子记录时，可定义源行中字段的输入层次结构。每个记录是层次结构中的一个节点，在定义转换输出时可引用它们。

例如，对于每种地址类型，源行可能包含多个电话号码：

```
CustomerID
FirstName
LastName
Home_Street
Home_City
Home_State
Home_Country
Telephone_No
Cell_Phone_No
Alternate_Phone_No
Business_Street
Business_City
Business_State
Business_Country
Business_Telephone_No
Business_Cell_Phone_No
Business-Alternate_Phone1
```

可定义一种“地址”为“电话”父项的输入层次结构。在定义规范器转换输出时，可以向客户信息以外的不同目标返回地址和电话号码。

定义的输入层次结构与以下示例类似：

```
CustomerID
FirstName
LastName
Address  (occurs twice)
  Street
  City
  State
  Country
  Phone
    Telephone_No (occurs three times)
```

输入层次结构定义

在创建规范器转换时，可定义说明源中记录和字段的输入层次结构。在转换的**规范器**视图中定义输入层次结构。

Developer tool 将基于您在输入层次结构中定义的字段创建转换输入端口。在定义转换输出组前，先定义输入组结构。

定义输入层次结构时，必须定义与源数据结构相对应的输入结构。源数据可能包含多组多次出现的字段。要定义结构，可以将在同一级别出现的记录配置为源中的另一记录。或者，可以定义其他记录内出现的记录。

输入层次结构示例

以下源行包含客户字段和出现两次的地址记录：

```
CustomerID
  FirstName
  LastName
  Address
    Street
    City
    State
    Country
  Address1
    Street1
    City1
    State1
    Country1
```

在**规范器**视图中定义输入结构时，可以添加 CustomerID、FirstName 和 LastName 作为字段。定义“地址”记录，并在地址中包括“街道”、“城市”、“省/自治区/直辖市”和“国家/地区”字段。将地址出现次数值更改为 2。

下图显示了**规范器**视图中的输入层次结构：

Normalizer

Name	Level	Occurs	Type	Precision	Scale
CustomerID	1	1	string	10	0
FirstName	1	1	string	10	0
LastName	1	1	string	10	0
⊟ Address	1	2			
Street	2	1	string	10	0
City	2	1	string	10	0
State	2	1	string	10	0
Country	2	1	string	10	0

规范器视图中的**出现次数**列标识源行中字段或记录的实例数。更改多次出现的字段或记录的**出现次数**列中的值。在本例中，客户字段出现一次，“地址”记录出现两次。

规范器视图中的**级别**列指示字段或记录在输入层次结构中的位置。客户字段位于层次结构的第 1 级别。“地址”记录也是第 1 级别。

规范器转换输入端口

在**规范器**视图中定义输入层次结构时，Developer 工具将创建规范器转换输入端口。更改输入层次结构中的字段时，Developer 工具将更改输入端口。

在**概览**视图中可查看规范器转换输入端口。在**概览**视图中可对输入端口重新排序。要更改输入端口，请在**规范器**视图中更新输入层次结构。

在输入层次结构中将某个字段定义为多次出现时，Developer 工具将为多次出现字段的每个实例创建一个输入端口。当某个记录为多次出现时，Developer 工具将为记录中字段的每个实例创建一个输入端口。

输入端口示例

下图显示了 Developer 工具为客户数据和多次出现的地址数据创建的输入端口：

Ports

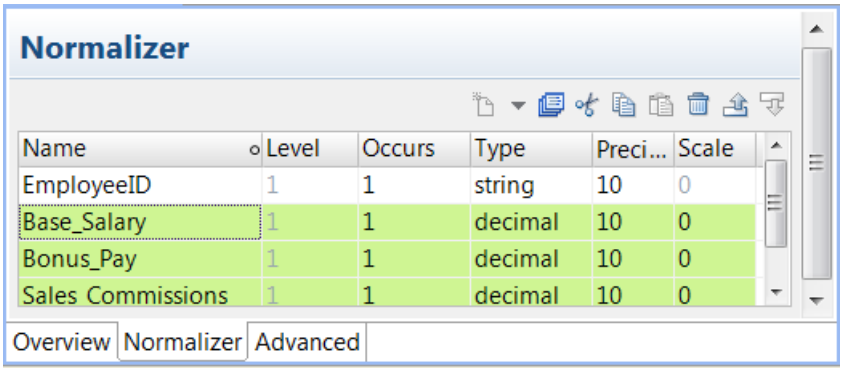
	Name	Type	Precision	Scale	Location
	Input				
1	CustomerID	string	10	0	CustomerID
2	FirstName	string	10	0	FirstName
3	LastName	string	10	0	LastName
4	Street	string	10	0	Address.Street
5	Street1	string	10	0	Address.Street
6	City	string	10	0	Address.City
7	City1	string	10	0	Address.City
8	State	string	10	0	Address.State
9	State1	string	10	0	Address.State
10	Country	string	10	0	Address.Country
11	Country1	string	10	0	Address.Country

合并字段

在**规范器**视图中，可以将类似数据的字段合并成单个多次出现的字段。在映射中从另一对象拖动端口来创建规范器转换时，可能需要合并字段。

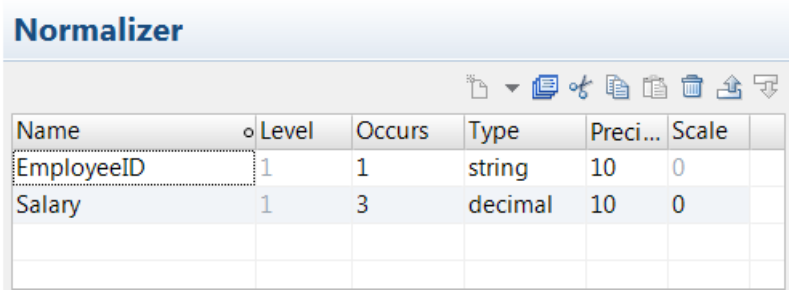
源行可能包含多个具有不同类型薪资数据的字段，例如 Base_Salary、Bonus_Pay 和 Sales_Commissions。可以合并这些字段而创建一个出现三次的“薪资”字段。

下图显示了一个员工行，在“规范器”视图中针对其选择了三种类型的薪资：



可以将这三种类型的薪资数据合并成一个出现三次的“薪资”字段。

下图显示了“薪资”字段：



合并字段

在“规范器”视图中，将类型相同的字段合并成一个多次出现的字段。

1. 单击**规范器**视图。
2. 选择想要合并的字段。
3. 单击**合并**按钮。

   此时将显示**合并字段**对话框。
4. 输入合并字段的名称、类型、精度、小数位数和多次出现字段的出现次数。
5. 单击**确定**。

平展字段

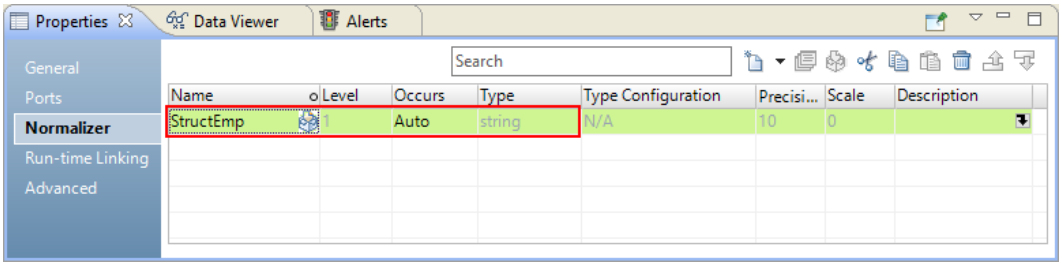
可以在运行于 Spark 引擎上的映射中平展复杂数据类型字段。在**规范器**视图中平展字段可以修改通过复杂端口传递的层次结构数据。

平展操作的输出取决于复杂数据类型。平展数组或结构数据类型时，规范器转换会为复杂数据类型的每个元素创建一个行。平展映射数据类型时，规范器转换会创建两个列，分别用于映射键元素和映射值元素。

针对嵌套数据类型的平展操作会在第一级提取元素。要在所有级别平展嵌套数据类型，请使用 Developer tool 中的**平展复杂端口**层次结构转换向导。**全部平展**选项在每个级别提取元素并返回原始数据类型的关系数据。有关层次结构转换向导的详细信息，请参阅 *Data Engineering Integration 用户指南*。

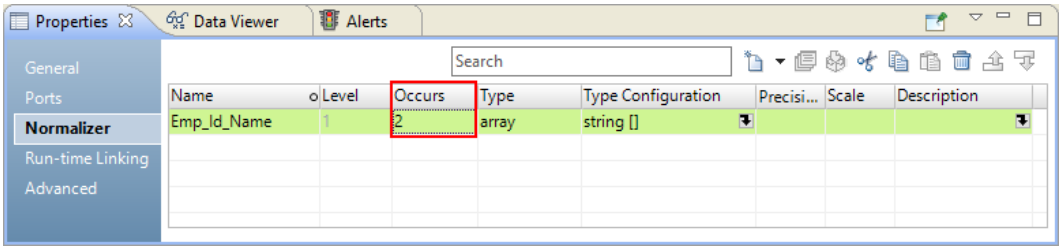
平展操作将“规范器”视图中“出现次数”列的值更改为“自动”，并在已平展字段的旁边添加平展图标。值“自动”表示转换将平展复杂数据类型的所有元素。

下图显示了平展为字符串字段的结构，其旁边有一个平展图标，并且“出现次数”值为“自动”：



无法平展多次出现的字段。例如，无法平展“出现次数”值为 2 的数组字段。

下图显示了无法平展的数组数据类型的多次出现字段：



平展数组

规范器转换将一维数组平展为原始数据类型，将 n 维数组平展为 (n-1) 维数组。转换创建的行数与数组的大小相同。

例如，如果平展具有 10 个字符串元素的数组端口，输出将返回 10 个字符串端口。如果平展 3 维数组，输出则会返回 2 维数组。

有一张包含一个字符串端口 Name 和一个数组端口 Phones 的表。您想要平展数组端口。表中包含以下值：

Name	Phones
Adams	[205-128-6478, 722-515-2889, 650-213-4020]
Jane	[650-321-4506]

平展数组端口时，输出如下：

Name	Phones	GCID_Phones
Adams	205-128-6478	1
Adams	722-515-2889	2
Adams	650-213-4020	3
Jane	650-321-4506	1

可以编辑已平展字段的“出现次数”值以提取数组中特定数量的元素。该值必须是一个大于 1 的正整数。该值确定要提取的元素数量。例如，可以将“出现次数”的值更改为 2 以提取数组的前两个元素。输出如下所示：

Name	Phones	GCID_Phones
Adams	205-128-6478	1
Adams	722-515-2889	2
Jane	650-321-4506	1

平展结构

转换将结构平展为结构中元素的数据类型的字段。要平展结构数据类型，所有 struct 元素的数据类型都必须相同。转换会为结构数据类型中的每个元素创建一个行。

例如，您想要平展以下结构字段：

```
customer_address{
 city : string
 state : string
 zip : string
 }
```

表中包含以下值：

Name	customer_address

Clara	{
	New York
	NY
	10032
	}

平展结构端口时，输出如下：

Name	customer_address	GCID_customer_address
Clara	New York	1
Clara	NY	2
Clara	10032	3

Struct 元素的数据类型不同但至少前两个元素为同一数据类型时，可以展平同一数据类型的连续元素的结构数据。要提取同一数据类型的连续 Struct 元素，请编辑“出现次数”值。该值必须是一个大于 1 的正整数。例如，结构 emp_address 包含以下元素：

```
emp_address{
 city : string
 state : string
 zip : int
 country : string
 }
```

您可以将“出现次数”的值定义为 2，以提取 city 和 state Struct 元素。如果将值定义为 3 或 4，映射验证将失败。

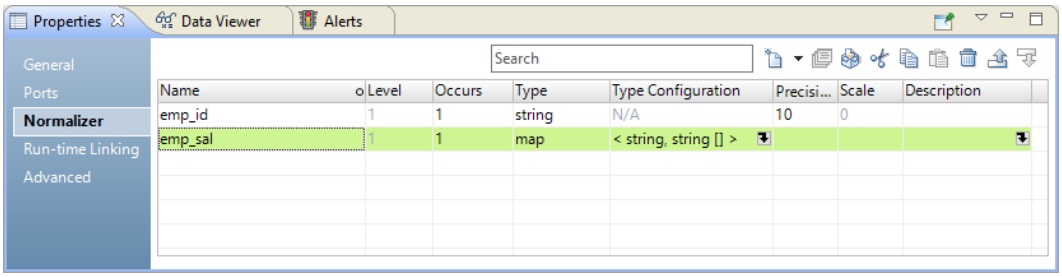
平展映射

转换将映射平展为两个字段，分别代表映射中的键元素和值元素。对于已平展的映射字段，无法将“出现次数”的值从“自动”更改为整数值。

例如，您想要平展具有字符串键和整数值数组的以下映射字段 emp_sal：

```
<emp_name -> [base_sal, bonus, commision]>
```

下图显示了要在“规范器”视图中平展的映射字段：



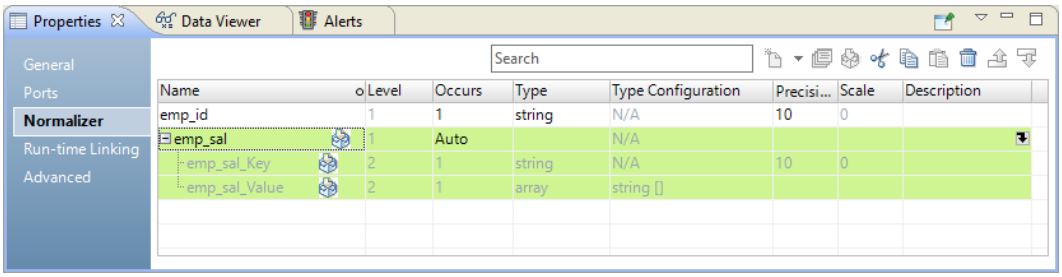
表中包含以下值：

emp_id	emp_sal
12200	<Greg -> [4000, 1000, 500]>
12201	<Patricia -> [3800, 1500, 1000]>

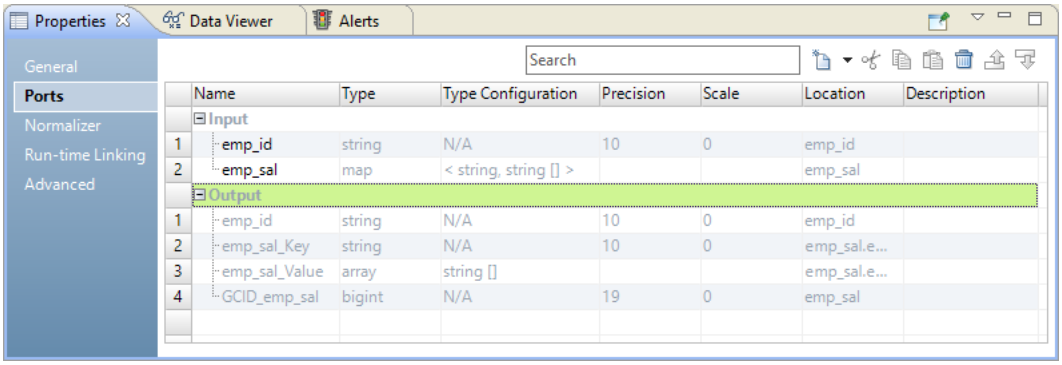
平展映射端口时，输出将为映射键返回字符串字段，为映射值返回数组字段，如下所示：

emp_id	emp_sal_Key	emp_sal_Value	GCID_emp_salary
12200	Greg	[4000, 1000, 500]	1
12201	Patricia	[3800, 1500, 1000]	1

下图显示已在“规范器”视图中平展为字符串键字段和数组值字段的映射字段：



下图显示“端口”视图中的“输出”组：

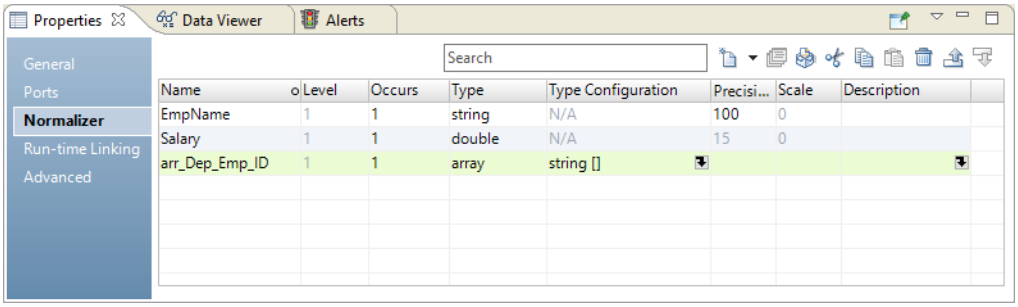


平展字段

平展复杂数据类型字段可修改层次结构数据或转换为关系数据。

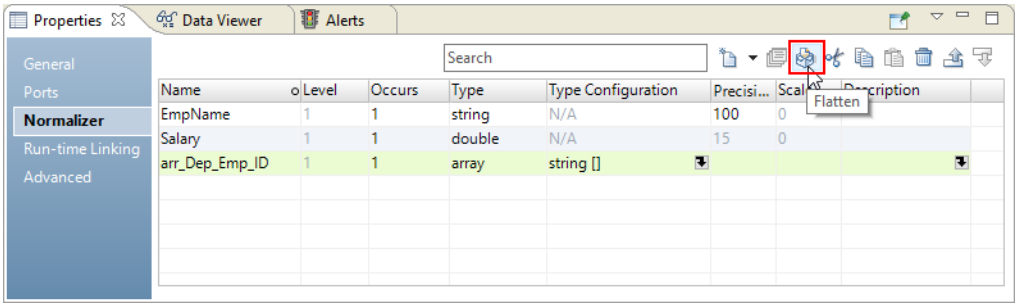
1. 单击**规范器**视图。
2. 选择复杂数据类型字段。

   下图显示了具有字符串元素的数组类型字段：



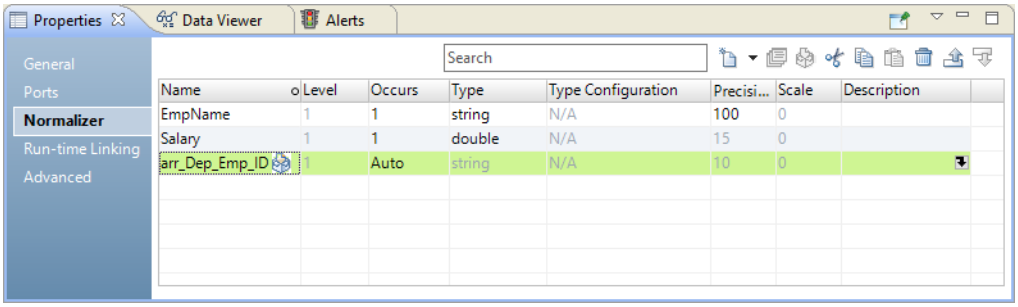
3. 单击**平展**按钮。

下图显示了“平展”按钮：

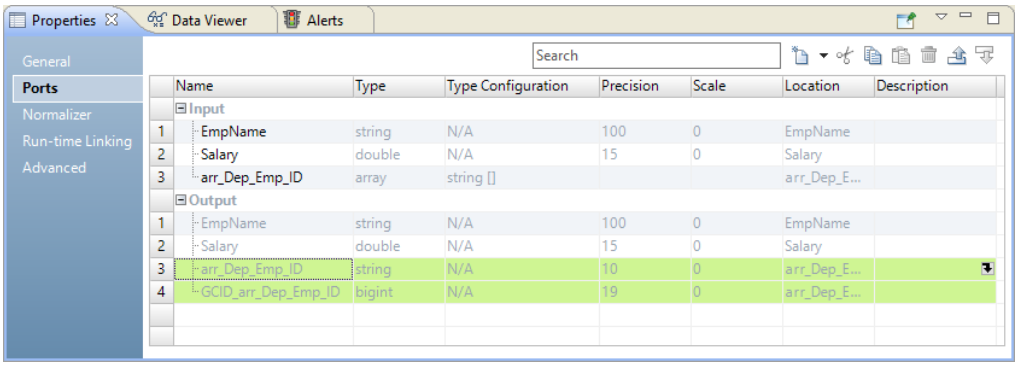


平展操作将复杂数据类型字段替换为已平展的字段，并将**出现次数**的值更改为“自动”。已平展字段的数据类型取决于平展的复杂数据类型。

下图显示了字符串类型的已平展字段：



下图显示了**端口**视图中的已平展字符串输出端口和 GCID 输出端口：



规范器转换输出组和端口

在规范器转换的“概览”视图中可定义输出组和端口。在定义转换输入层次结构后，即可定义输出组。

默认情况下，Developer 工具至少生成一个输出组。输出组包含输入端口的所有 1 级字段和第一个多次出现的字段。如果要在“规范器”视图中定义多个多次出现的字段，Developer tool 将针对每个附加的多次出现字段创建一个输出组。

以下源行包含客户数据、多次出现的销售字段和多次出现的电话字段。

```
CustomerID
LastName
```

```
FirstName
Sales (occurs 4 times)
Phone (occurs 3 times)
```

Developer 工具根据输入结构将创建两个输出组：

```
Output
CustomerID
LastName
FirstName
Sales

Output1
Phone
GCID_Phone
```

由于源数据包含多个多次出现的字段，所以 Developer 工具创建 Output1 组。Developer 工具不会为您在记录中定义的字段创建组。定义记录时，必须定义包含记录中字段的输出组。

以下源行包含客户字段和出现两次的地址记录：

```
CustomerID
  FirstName
  LastName
  Address
    Street
    City
    State
    Country
  Address1
    Street1
    City1
    State1
    Country1
```

Developer 工具将创建一个包含以下字段的输出组。

```
CustomerID
FirstName
LastName
```

Developer 工具将为 1 级客户字段创建默认输出组。默认输出组不包括“地址”记录。您必须配置在输出中返回“地址”数据的方式。

根据构造输出行所需的方式创建输出组。如果源行包含客户数据和地址数据，可以为客户字段创建一个输出组。再为地址字段创建另一个输出组。或者，可以更新默认输出组并在其中添加地址字段。以下示例显示了基于输出组配置的不同输出结果。

创建输出组

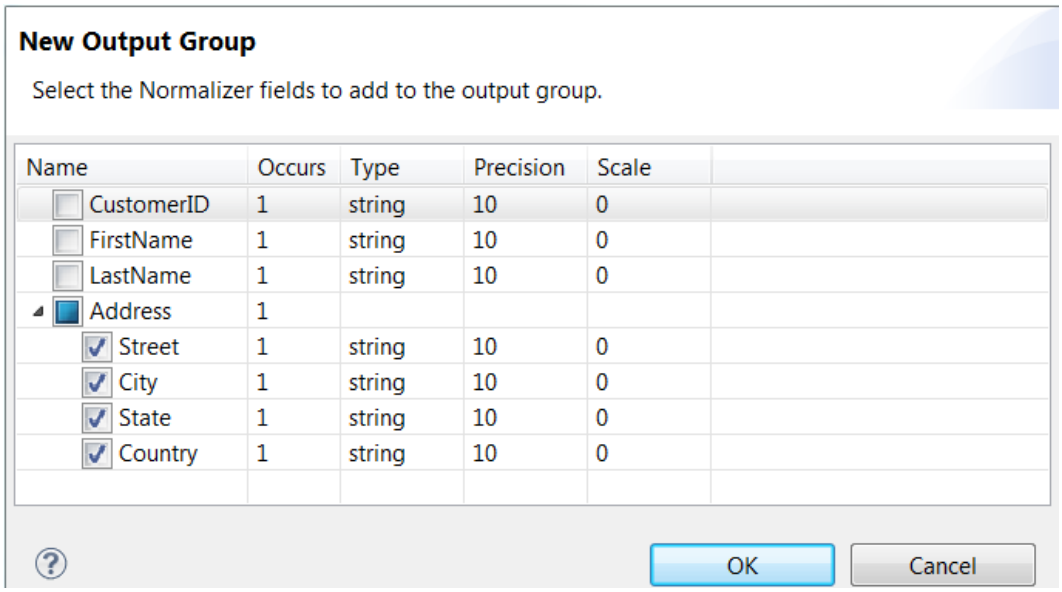
在规范器转换的**概览**视图中创建输出组。

打开规范器转换的**概览**视图时，Developer 工具将显示其基于输入层次结构创建的默认组。

创建新输出组时，对话框将显示输入层次结构中的字段和记录列表。选择组中要包含的字段或记录。

输出组示例

下图显示了**新建输出组**对话框：



选择“地址”记录时，Developer 工具将创建一组与“地址”记录中的字段相对应的输出端口。Output1 组包含“街道”、“城市”、“省/自治区/直辖市”和“国家/地区”端口。可以更改输出组中的端口。

下图显示了**概览**视图中的 Output 组和 Output1 组：

	Output				
1	CustomerID	string	10	0	CustomerID
2	FirstName	string	10	0	FirstName
3	LastName	string	10	0	LastName
	Output1				
1	Street	string	10	0	Address.Street
2	State	string	10	0	Address.State
3	Country	string	10	0	Address.Country
4	GCID_Address	bigint	19	0	Address

可以将规范器转换配置为从 Output 组向“客户”表返回行。

“客户”表将会收到类似于以下行的数据：

```
100, Robert, Bold
200, James, Cowan
```

可以从 Output1 组向“地址”表返回行。“地址”表将会收到“街道”、“城市”、“省/自治区/直辖市”、“国家/地区”和 GCID。

“地址”表将会收到类似于以下行的数据：

```
100 Summit Dr, Redwood City, CA, United States,1
41 Industrial Way, San Carlos, CA, United States,2
85 McNulty Way, Los Angeles, CA, United States,1
55 Factory Street, Los Vegas, NV, United States,2
```

GCID 标识输出行中的客户地址实例。在本例中，规范器转换将返回两个“地址”记录的实例。每个输出行包含一个 GCID 值为 1 或 2 的值。

更新输出组

可以更新规范器转换输出组。可以在组中添加或删除字段。

默认情况下，在定义输入层次结构时，Developer 工具将创建 1 级输出组。Developer 工具不包括组中的记录。可以更新默认输出组并在其中添加记录。

要更新输出组，请突出显示组名称并单击**新建** > **更新组**。**编辑输出组**对话框将显示输入层次结构中的字段。选择组中要包括的字段。

更新输出组示例

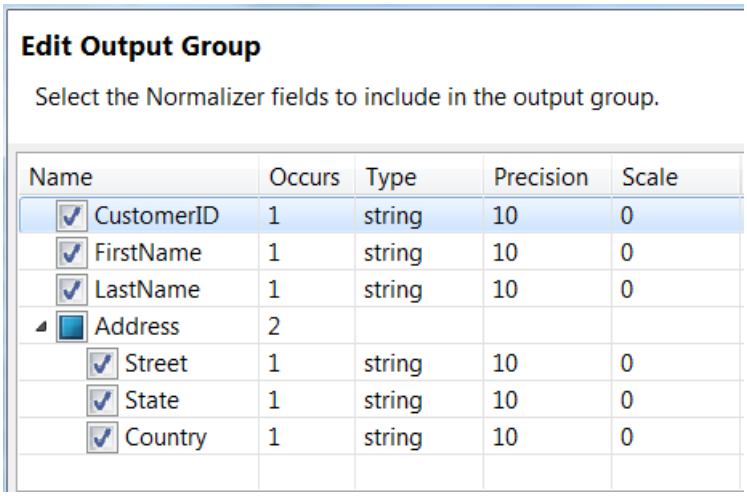
在上一示例中，Developer 工具创建了一个包含 CustomerID、FirstName 和 LastName 字段的默认输出组。

下图显示了默认输出组：

	Output				
1	CustomerID	string	10	0	CustomerID
2	FirstName	string	10	0	FirstName
3	LastName	string	10	0	LastName

可以更新默认输出组并在其中添加“地址”记录。

下图显示了**编辑输出组**对话框：



在本例中，CustomerID、FirstName 和 LastName 是 1 级节点。“地址”记录也是 1 级节点。规范器转换可与客户数据同行返回地址。由于“地址”为多次出现，Developer 工具将向输出组中添加 GCID_Address 索引。

下图显示了输出组中的端口：

	Output				
1	CustomerID	string	10	0	CustomerID
2	FirstName	string	10	0	FirstName
3	LastName	string	10	0	LastName
4	Street	string	10	0	Address.Street
5	State	string	10	0	Address.State
6	Country	string	10	0	Address.Country
7	GCID_Address	bigint	19	0	Address

当客户字段和多次出现的地址字段都在输出组中时，规范器转换将对每个地址数据实例返回相同的客户字段。

以下示例显示了规范器转换从输出组生成的行：

```
100, Robert, Bold, 100 Summit Dr, Redwood City, CA, United States,1
100, Robert, Bold, 41 Industrial Way, San Carlos, CA, United States,2
200, James, Cowan, 85 McNulty Way, Los Angeles, CA, United States,1
200, James, Cowan, 55 Factory Street, Los Vegas, NV, United States,2
```

GCID 端口包含地址实例编号。GCID 值为 1 或 2。

输出组的键生成

可以配置序列生成器转换来生成键，以链接规范器转换从同一源行返回的每个输出行。

在映射中，可以在规范器转换之前添加序列生成器转换。序列生成器转换将为每个源行添加一个序号。当规范器转换从同一源行返回多个输出组或行时，每个输出行会收到相同的序号。可以使用该编号作为目标表之间主键-外键关系中的键。

例如，规范器转换在某个输出组中返回客户信息，并在另一个输出组中返回订单信息。您可以使用序号将表中的客户信息链接到另一个表中的订单信息。

规范器转换高级属性

在**高级**选项卡上配置规范器转换的属性。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

生成第一级输出组

默认情况下，Developer 工具会生成规范器转换第一级输出组。在“规范器”视图中定义多个多次出现的字段时，Developer 工具将生成附加输出组。

Developer 工具可创建包含所有 1 级单次出现字段和第一个 1 级多次出现字段的输出组。当转换包含多个多次出现的字段时，Developer 工具可创建附加输出组。

Developer 工具不会为 1 级记录创建输出组。在定义 1 级记录时，必须配置在输出中放置它们的位置。

要阻止 Developer 工具为多次出现的字段创建输出组，请禁用**自动生成第一级输出组**高级属性。

创建规范器转换

可以创建可重用或不可重用规范器转换。 可重用转换可存在于多个映射中。不可重用转换存在于单个映射内。

1. 要创建转换，请使用以下方法之一：

选项	说明
可重用	在**对象浏览器**视图中选择一个项目或文件夹。单击**文件** > **新建** > **转换**。选择规范器转换，然后单击**下一步**。
不可重用	在映射或 Mapplet 中，将规范器转换从**转换**选项板拖动到编辑器。在映射中，可以从源数据对象或转换中拖动端口来定义转换。

   此时将显示**新建规范器转换**向导。
2. 输入转换的名称。
3. 单击**下一步**。

   此时将显示**规范器定义**页面。
4. 要添加记录，单击**新建**按钮，然后选择**记录**。
5. 要添加字段，单击**新建**按钮，然后选择**字段**。

   要向记录中添加字段，必须在添加字段之前选择记录。
6. 或者，双击**出现**列中的值以更改字段或记录的出现次数。
7. 单击**下一步**。

   此时将显示**规范器端口**页面。
8. 要添加输出组，请单击**新建**按钮，然后选择**新建输出组**。
9. 要编辑输出组，选择想要编辑的输出组。单击**新建**按钮，然后选择**编辑输出组**。
10. 单击**完成**。

    此时，转换将显示在编辑器中。

创建规范器转换 从上游源

可以创建空规范器转换，并将端口从源数据对象或转换拖动到规范器转换来创建输入和端口。

1. 创建一个包括源或转换的映射，以将源数据传递到规范器转换。
2. 要创建规范器转换，请从“转换”选项板中选择规范器转换并将该转换拖动到编辑器中。

   此时将显示“规范器”对话框。
3. 单击**完成**创建一个空转换。
4. 从映射中的源或转换选择端口，并将它们拖动到规范器转换中。

   规范器转换中将显示输入和输出端口。Developer 工具将创建一个输入组和一个输出组。
5. 打开**规范器**视图，以更新默认组并根据需要将字段组织到记录中。
6. 要将多个字段合并为一个多次出现的字段，请在**规范器**视图中选择这些字段，然后单击**合并**选项。

   为多次出现的字段选择一个名称。

规范器映射示例

零售组织可收到组织中商店的销售总额。组织将收到一行数据，其中包含商店信息和四个销售额。每个销售额代表当年一个季度的总销售额。

以下示例显示了如何定义规范器转换向 Store 目标和 Sales 目标返回销售数据。Store 目标对于每个商店收到一行。Sales 目标对于每个商店收到四行。每行包含一个季度的销售数据。

序列生成器转换将为每个商店生成唯一的 ID。规范器转换随每个输出行返回 StoreID。

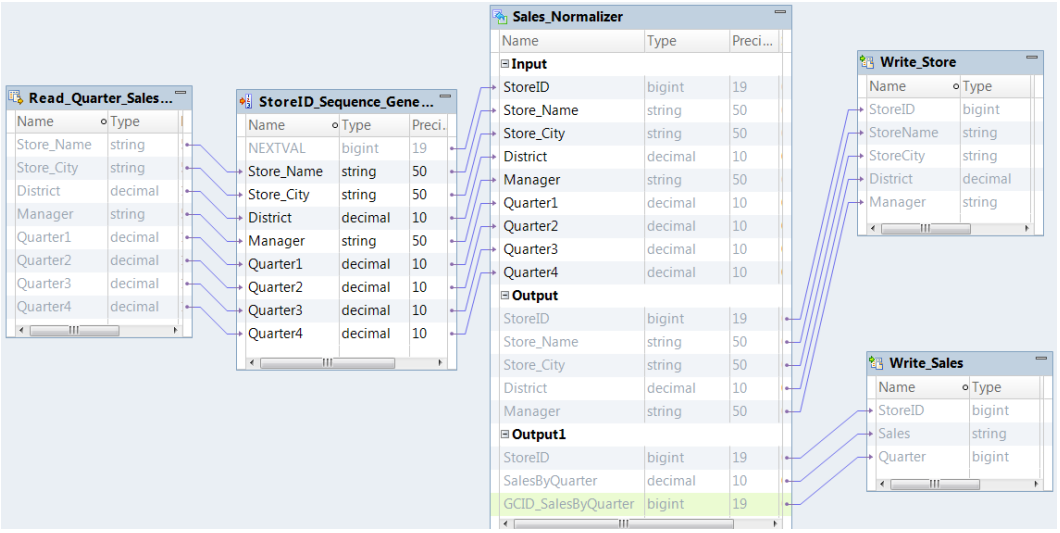
创建包含一个读取转换、一个序列生成器转换、规范器转换和两个写入转换的映射。

规范器示例映射

创建一个包含规范器转换的映射以规范化平面文件源中多次出现的季度销售数据。

规范器转换为每季度销售数据生成一个单独的输出行并将规范化的销售金额写入 Sales 目标。规范器转换将商店信息写入 Store 目标。

下图显示了规范器转换映射：



映射包含以下对象：

Read_STORE

包含多次出现字段的数据源。

StoreID 序列生成器转换

生成 storeID 键以链接 Store 表和 Sales 表的序列生成器转换。

Sales_Normalizer

规范化多次出现的销售数据的规范器转换。

Write_Store

从规范器转换接收商店信息的目标。

Write_Sales

从规范器转换接收销售数字的目标。

规范器示例定义

源即包含商店信息和季度销售数据的平面文件。在**规范器**视图中定义源数据的结构。

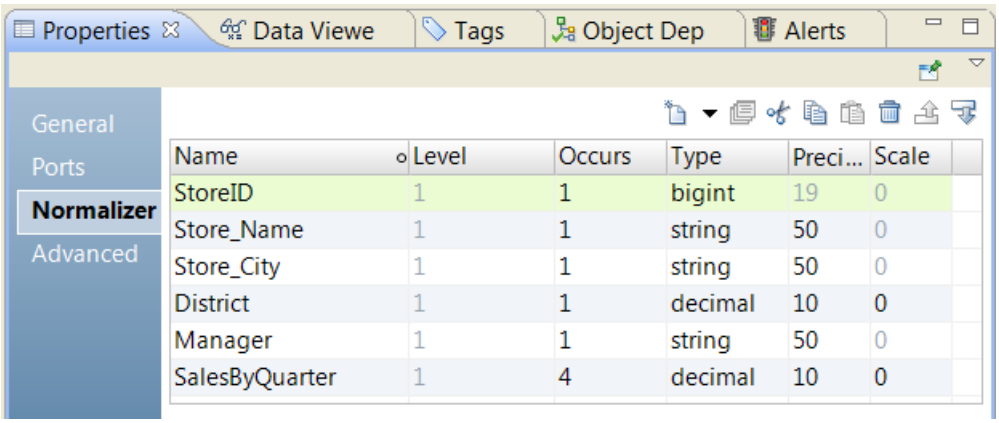
STORE 平面文件包含以下源数据：

StoreID	Store_Name	Store_City	地区	经理	季度 1	季度 2	季度 3	季度 4
1	BigStore	纽约	东	Robert	100	300	500	700
2	SmallStore	Phoenix	西	Radhika	250	450	650	850

将平面文件作为读取转换添加到映射，然后创建空的规范器转换。将端口从 Read_STORE 数据对象拖动至规范器转换以创建规范器定义。

规范器视图包含 Store_Name、Store_City、“地区”和“经理”字段的一个实例。**规范器**视图包含名为 QUARTER 的字段的四个实例。合并 QUARTER 字段以创建一个出现四次的 SalesByQuarter 字段。

下图显示了包含合并 Quarter 字段的规范器定义：



规范器示例输入和输出组

修改输入层次结构后，规范器转换将包含一个输入组和一个默认输出组。您需要将输出端口重组到两个组中。一个组需要包含商店信息，一个组需要包含销售信息。

对于源中的每个字段输入组都包含一个端口。输出组包含用于商店字段的端口和多次出现的 SalesByQuarter 字段的端口。此外，输出组还包含生成的列 ID — GCID_SalesByQuarter，它与多次出现的 SalesByQuarter 字段相对应。

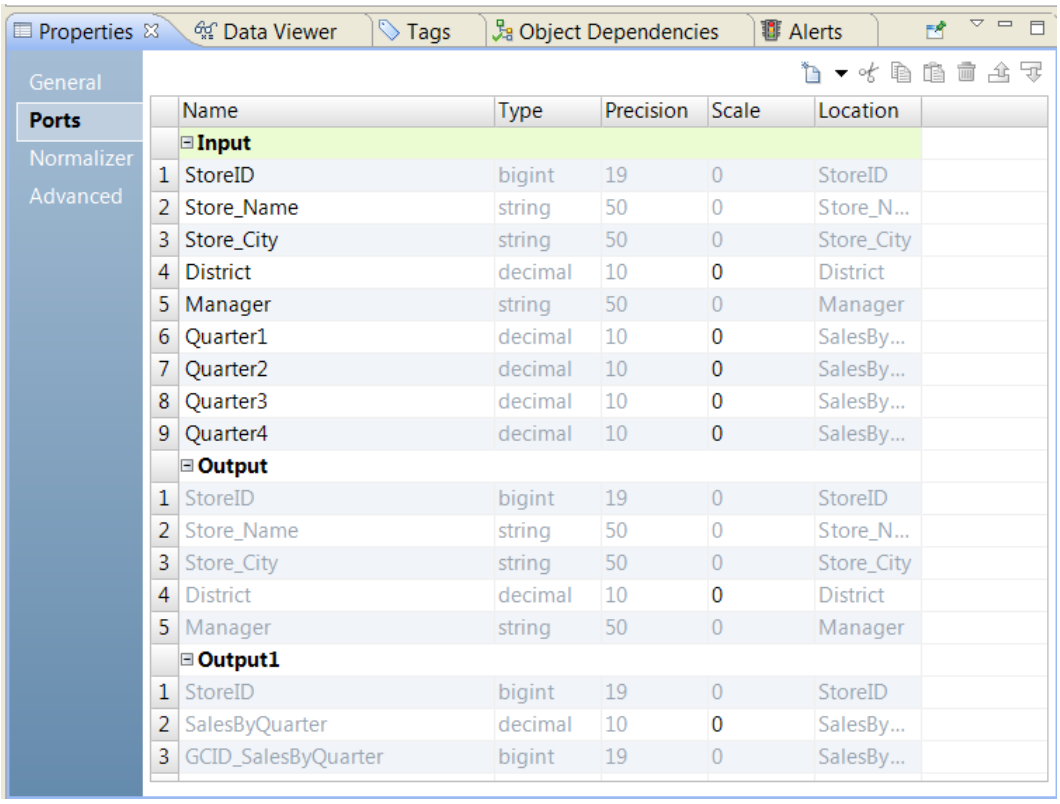
要向不同的目标返回季度销售数据，请在**概览**视图中创建一个新组。在 Output1 组中添加以下字段：

```
StoreID
SalesByQuarter
GCID_SalesByQuarter
```

更新默认输出组。删除以下字段：

```
SalesByQuarter
GCID_SalesByQuarter
```

下图显示了**概览**视图中的输入组和输出组：



	Name	Type	Precision	Scale	Location
	Input				
1	StoreID	bigint	19	0	StoreID
2	Store_Name	string	50	0	Store_N...
3	Store_City	string	50	0	Store_City
4	District	decimal	10	0	District
5	Manager	string	50	0	Manager
6	Quarter1	decimal	10	0	SalesBy...
7	Quarter2	decimal	10	0	SalesBy...
8	Quarter3	decimal	10	0	SalesBy...
9	Quarter4	decimal	10	0	SalesBy...
	Output				
1	StoreID	bigint	19	0	StoreID
2	Store_Name	string	50	0	Store_N...
3	Store_City	string	50	0	Store_City
4	District	decimal	10	0	District
5	Manager	string	50	0	Manager
	Output1				
1	StoreID	bigint	19	0	StoreID
2	SalesByQuarter	decimal	10	0	SalesBy...
3	GCID_SalesByQuarter	bigint	19	0	SalesBy...

StoreID 是生成的用来链接商店信息和销售信息的键。验证两个输出组是否都返回了 StoreID。

规范器示例映射输出

向映射添加写入转换，并将规范器转换输出端口连接到数据对象。

运行映射时，规范器转换将向 Store 目标写入以下行：

StoreID	Store_Name	Store_City	地区	经理
1	BigStore	纽约	东	Robert
2	SmallStore	Phoenix	西	Radhika

规范器转换将向 Sales 目标写入以下行：

StoreID	SalesByQuarter	GCID_SalesByQuarter
1	100	1
1	300	2
1	500	3
1	700	4
2	250	1
2	450	2
2	650	3
2	850	4

非本地环境中的规范器转换

非本地环境中的规范器转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射和流映射中无限支持。
- Databricks Spark 引擎。无限支持。

第 33 章

合并转换

本章包括以下主题：

合并转换概览

合并转换属于被动转换，可从多个输入列中读取数据值，并创建单个输出列。

使用合并转换以首选格式创建数据。 例如，可以将 Customer_Firstname 与 Customer_Surname 字段组合在一起创建名为 Customer_FullName 的字段。

在合并转换中，可以创建多个合并策略。 合并转换提供了用于创建策略的向导。

配置合并策略

要配置合并策略，请在合并转换的**策略**视图中编辑相应设置。

1. 选择**策略**视图。
2. 单击**新建**。

   此时将打开 **新建策略** 向导。
3. 单击**输入**字段为策略选择输入端口。
4. 要定义置于合并项目之间的合并字符，请单击**选择**。 如果未选择合并字符，则合并转换将默认使用空格字符。
5. （可选）选择**将空字符串包含在合并输出中**以便将空的输入字符串包含在输出中。
6. 单击**完成**。

合并转换高级属性

配置有助于确定数据集成服务如何为合并转换处理数据的属性。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的合并转换

非本地环境中的合并转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中无限支持。在流映射中不受支持。
- Databricks Spark 引擎。无限支持。

第 34 章

解析器转换

本章包括以下主题：

解析器转换概览

解析器转换是一种被动转换，可将输入数据值解析到新端口中。 此转换会根据值包含的信息类型以及值在输入字符串中的位置将这些值写入新端口中。

希望更改数据集结构时，可以使用解析器转换。 解析器转换可以向数据集中添加列并可以向新列中写入数据值。 如果数据列在一个列中包含多个值，而您希望根据数据值包含的信息类型将这些数据值写入到各个列，则可以使用解析器转换。

解析器转换会将数据值解析到所定义的输出端口中。 如果此转换可以识别某个输入数据值，但已定义的输出端口不可用，则它会将此值写入到溢出端口中。 如果此转换无法识别某个输入数据值，则它会将此值写入到未解析的数据端口中。

解析器转换模式

创建解析器转换时，可以选择标志解析模式或基于模式的解析模式。

您可以选择以下模式之一：

- 标志解析模式。 使用此模式可解析与引用数据对象（如标志集、正则表达式、概率模型和引用表）中的值匹配的输入值。 您可以在一个转换中使用多个标志解析策略。
- 基于模式的解析模式。 使用此模式可解析与模式集中的值匹配的输入值。

何时使用解析器转换

如果某个列中的数据字段包含多个类型的信息，则要将字段值移动到新列，请使用解析器转换。解析器转换可以为数据集中每种类型的信息创建一个新列。

以下示例介绍了可以使用解析器转换执行的一些结构更改类型：

为联系人数据创建新列

可以创建一个数据结构，将单个列中的姓名数据解析为多个列。例如，可以创建称呼列、名字列、中间名列和姓氏列。

可以使用一个概率模型配置转换，该模型表示输入端口上的人名结构。可以使用一个输入端口数据样本定义该模型。

可以创建一个标志解析策略，以便将该概率模型应用于输入端口，并将姓名值写入新列。此转换会根据输入字符串中每个值的位置以及该值所代表的姓名类型将姓名值写入新列。

注意: 还可以使用基于模式的解析策略来解析联系人数据。配置基于模式的解析策略时，可定义表示输入端口上的姓名结构的模式。

创建地址列

可以创建一个数据结构，以便将一个地址数据列解析为多个列，以描述可传递地址。

为此转换配置引用表，这些表包含可识别的地址元素（例如，邮政编码、省/自治区/直辖市名称和城市名称）。创建一个标志解析策略，以便将每个地址元素写入一个新端口。

您无法使用引用表解析输入字符串中的街道地址数据，因为街道名称和编号数据过于宽泛，无法从引用表中捕获。但是，可以使用“溢出”端口捕获该数据。在解析了某一地址中所有城市、省/自治区/直辖市和邮政编码数据后，剩余数据将包含街道信息。

例如，使用一个标志解析策略，将以下地址拆分成多个地址元素：

```
123 MAIN ST NW STE 12 ANYTOWN NY 12345
```

此解析策略可以将这些地址元素写入以下列：

列名	数据
溢出	123 MAIN ST NW STE 12
城市	ANYTOWN

列名	数据
状态	NY
邮政编码	12345

创建产品数据列

可以创建一个数据结构，以便将单个产品数据列解析为多个列，以描述产品清单明细。

为此转换配置标志集，这些标志集包含多个清单元素（例如，尺寸、颜色和重量）。创建一个标志解析策略，以便将每个清单元素写入一个新端口

例如，使用标志解析策略，将以下上色说明拆分成单独的清单元素：

```
500ML Red Matt Exterior
```

此解析策略可以将这些值写入以下列：

列名	数据
大小	500ML
颜色	红色
样式	Matt
外部	Y

引用数据在解析器转换中的使用

Informatica Developer 安装有多个可与解析器转换一起使用的引用数据对象。 您还可以在 Developer 工具中创建引用数据对象。

向解析器转换添加引用数据对象时，该转换会将匹配对象中某个值的字符串写入您指定的新列中。

下表介绍了可以使用的引用数据类型：

引用数据类型	说明
模式集	基于字符串中各个值的相对位置标识数据值。
概率模型	将模糊匹配功能添加到标志解析操作。该转换可以使用概率模型来推理字符串中的信息类型。要启用模糊匹配功能，请在 Developer 工具中编译概率模型。
引用表	查找与数据库表中的条目相匹配的字符串。

引用数据类型	说明
正则表达式	标识与您所定义的条件相匹配的字符串。可以使用正则表达式在一个较大的字符串中查找某一字符串。
标志集	基于字符串所包含的信息类型来标识这些字符串。 Informatica 将不同类型的标志定义（例如文字、电话号码、邮政编码以及产品代码定义）与标志集安装在一起。

模式集

模式集包含表达式，可用于标识“为标志添加标签”操作的输出中的数据模式。 可以使用模式集分析标志化的数据输出端口，并将匹配的字符串写入一个或多个输出端口。 在使用模式解析模式的解析器转换中使用模式集。

例如，可以将解析器转换配置为使用可标识姓名和首字母的模式集。 在“为标志添加标签”模式下，该转换使用模式集分析标签转换的输出。 可以配置解析器转换以将输出中的姓名和首字母写入各个端口。

概率模型

概率模型可依据标志所包含的信息类型以及标志在输入字符串中的位置对标志进行标识。

概率模型包含引用数据值和标签值。引用数据值代表连接到转换的输入端口上的数据。标签值描述引用数据值包含的信息类型。请为模型中的每个引用数据值分配标签。

要将引用数据值链接到概率模型中的标签，您可以对该模型进行编译。编译过程会在数据值与标签之间生成一系列逻辑关联。当您运行读取模型的映射时，数据集成服务会将模型逻辑应用到转换输入数据。数据集成服务将返回最能准确描述输入数据值的标签。

您可以在 Developer tool 中创建概率模型。模型存储库存储概率模型对象。Developer tool 将数据值、标签和编译数据写入 Informatica 目录结构中的一个文件。

注意: 如果将概率模型添加至标志解析操作，然后编辑概率模型中的标签配置，则会使该操作失效。 更新概率模型中的标签配置后，请重新创建任何使用该模型的解析操作。

引用表

引用表是一个至少包含两列的数据库表。 一列包含标准版本或所需版本的数据值，另一列包含备选版本的数据值。 将引用表添加到转换时，该转换将在输入端口数据中搜索同时显示在引用表中的值。 可以使用任何对所处理的数据项目有用的数据来创建引用表。

正则表达式

在解析操作上下文中，正则表达式是可用于标识输入数据中的一个或多个字符串的表达式。 解析器转换会将所标识的字符串写入一个或多个输出端口。 可以在使用标志解析模式的解析器转换中使用正则表达式。

解析器转换使用正则表达式匹配输入数据中的模式，并将所有匹配的字符串解析到一个或多个输出。 例如，您可以使用正则表达式标识输入数据中的所有电子邮件地址，并将每个电子邮件地址组件解析到不同的输出。

标志集

标志集包含能够标识特定标志的表达式。 可以在使用标志解析模式的解析器转换中使用标志集。

使用标志集可以将特定标志标识为解析操作的一部分。 例如，您可以使用标志集解析所有采用“AccountName@DomainName”格式的电子邮件地址。

标志解析操作

在标志解析模式中，解析器转换解析与标志集、正则表达式、概率模型或引用表条目中数据相匹配的字符串。

要执行标志解析，请在转换的**策略**视图上添加策略。 您可以向每个策略添加一个或多个操作。 该转换提供一个向导，用于创建策略。

您可以向标志解析策略添加以下类型的操作：

使用标志集解析

使用预定义或用户定义标志集来解析输入数据。 标志集操作可以使用将数据写入到一个或多个输出的自定义正则表达式。

您还可以使用概率模型来标识和分析输入数据值。

使用引用表解析

使用引用表可解析输入数据。

该转换按操作在策略中出现的顺序执行操作。

标志解析端口

使用适用于您的数据的设置配置标志解析端口。

标志解析模式下的解析器转换具有以下端口类型：

输入

包含要传递给解析器转换的数据。 该解析器转换将使用**策略**选项卡上指定的**输入联接字符**将所有输入端口合并为一个组合数据字符串。 如果未指定输入联接字符，则该转换将默认使用空格字符。

已解析的输出端口

用户定义的输出端口，其中包含已成功解析的字符串。 如果多个解析策略使用同一输出，则转换将使用**策略**选项卡上指定的**输出联接字符**将该输出合并为一个组合数据字符串。 如果未指定输出联接字符，则该转换将默认使用空格字符。

溢出

包含已成功解析的字符串，这些字符串不符合转换中定义的输出数量。 例如，如果转换只有两个“WORD”输出，则字符串“John James Smith”将生成一个溢出输出“Smith”。 解析器转换将为添加的每个策略创建一个溢出端口。

如果选择“详细溢出”选项，则转换将为模型中的每个标签创建一个溢出端口。

未解析

包含无法由转换成功解析的字符串。 解析器转换将为添加的每个策略创建一个未解析端口。

概率匹配中的输出端口

如果配置了一个解析策略以使用概率匹配技术，则解析器转换将为每个输出端口添加一个端口来存储匹配得分。

下表介绍了端口类型：

端口类型	概率匹配中创建的端口
已解析的输出端口	[标签名称] 输出 [标签名称] 得分输出
溢出数据端口	[溢出数据] 输出 [溢出数据] 得分输出
未解析的数据端口	[未解析的数据] 输出 [未解析的数据] 得分输出

标志解析属性

在解析器转换的**策略**视图中，配置标志解析操作的属性。

常规属性

常规属性适用于在策略中定义的所有标志解析操作。 使用常规属性命名策略、指定输入和输出端口并指定策略是否支持概率匹配技术。

下表介绍了常规属性。

属性	说明
名称	提供策略名称。
输入	标识策略操作可以读取的输入端口。
输出	标识策略操作可以写入的输出端口。
说明	描述策略。 该属性为可选。
使用概率匹配技术	指定策略可以使用概率模型来标识标志。
输入联接字符	指定用于联接输入数据端口的字符。 该转换会将所有输入端口合并为组合数据字符串，然后解析完整的字符串。
输出联接字符	当多个解析操作使用相同的输出时，指定用于联接输出数据值的字符。
已启用反向	将策略配置为从右至左解析数据。 已对概率匹配禁用此属性。
已启用溢出反向	将策略配置为从右至左解析溢出数据。 已对概率匹配禁用此属性。
已启用详细溢出	为每个解析操作创建唯一溢出字段。
分隔符	指定用于将输入数据分隔到独立标志中的分隔符。 默认值为空格。

概率模型属性

配置标志解析策略时，可以选择概率模型来代替标志集。 选择**使用标志集解析**操作，然后选择相应选项以使用概率匹配技术。

下表介绍了概率模型属性：

属性	说明
名称	请提供操作名称。
筛选文本	使用您输入的字符或通配符筛选标志集列表、概率模型或正则表达式。
概率模型	标识选定的概率模型。

引用表属性

将添加标签操作配置为使用引用表时，将应用引用表属性。

下表介绍了引用表属性：

属性	说明
名称	请提供操作名称。
引用表	指定操作用于解析输入值的引用表。
区分大小写	确定输入字符串是否必须与引用表条目的大小写相匹配。
将匹配项替换为有效值	将解析数据替换为引用表中“有效”列中的数据。
输出	为解析数据指定输出端口。

标志集属性

配置解析操作以使用标志集时，将应用标志集属性。

选择**使用标志集解析**操作以使用标志集解析输入。 清除此选项将使用概率匹配技术。

下表介绍了标志集属性：

属性	说明
Name	请提供操作名称。
标志集(仅限单个输出)	指定操作用于解析数据的标志集。操作向单个端口写入数据。
正则表达式(适用于单个或多个输出)	指定操作用于解析数据的正则表达式。如果在输入字段中发现多个字符串，操作会向多个端口写入数据。
输出	标识操作对其进行写入操作的输出端口。

您可以添加、编辑、导入或删除标志集或正则表达式。您也可以筛选标志集列表。

下表描述了用于执行任务的属性：

属性	说明
筛选文本	筛选标志集或正则表达式的列表。使用文本字符和通配符作为筛选器。
添加	用于定义自定义标志集或正则表达式。
编辑	编辑自定义标志集的内容。
导入	从模型存储库的文件夹中导入标志集或正则表达式的不可重用副本。如果更新标志集或正则表达式的源对象，数据集成服务不会更新不可重用副本。
删除	删除自定义标志集或正则表达式。

基于模式的解析模式

在基于模式的解析模式下，解析器转换可以解析由多个字符串组成的模式。

在基于模式的解析模式下，您可以使用以下方法定义模式：

- 使用引用表中定义的模式解析输入数据。 您可以通过使用“为标志添加标签”模式的标签创建器转换中经过剖析的输出来创建模式引用表。
- 使用定义的模式解析输入数据。
- 使用从模型存储库中的可重用模式集导入的模式解析输入数据。 对可重用模式集进行更改不会更新您在解析器转换中添加的数据。

可以使用“+”和“*”通配符定义模式。 使用“*”字符可匹配任意字符串，使用“+”字符可匹配前一个字符串的一个或多个实例。 例如，使用“WORD+”可查找由某个单词标志的多个连续实例组成的字符串，使用“WORD *”可查找由某个单词标志及其后的一个或多个任意类型的标志组成的字符串。

在解析器转换中可以使用这些方法中的多个实例。 此转换按照实例在**配置**视图中列出的顺序来使用实例。

注意: 在基于模式的解析模式下，解析器转换需要用到使用“为标志添加标签”模式的标签创建器转换的输出。在创建使用基于模式的解析模式的解析器转换之前，需要先创建和配置标签创建器转换。

基于模式的解析端口

使用适用于您的数据的设置配置基于模式的解析端口。

使用基于模式的解析模式的解析器转换具有以下端口类型：

Label_Data

将此端口连接到使用“为标志添加标签”模式的标签创建器转换的 Labeled_Output 端口。

Tokenized_Data

将此端口连接到使用“为标志添加标签”模式的标签创建器转换的 Tokenized_Data 输出端口。

Parse_Status

如果找到输入模式的匹配项，此端口将输出已匹配值。 如果未找到匹配项，它将输出不匹配。

溢出

已成功解析但不符合转换中定义的输出数的字符串。 例如，如果只定义了两个“WORD”输出，则默认情况下，字符串“John James Smith”将产生溢出输出“Smith”。

已解析

用户定义的端口中已成功解析的字符串。

配置标志解析策略

要配置标志解析策略，请在标志解析模式下打开解析器转换，然后选择**策略**视图。

1. 选择**策略**视图。
2. 单击**新建**。

   此时将打开 **新建策略** 向导。
3. 单击**输入**字段为策略选择端口。
4. 配置策略属性，然后单击**下一步**。
5. 选择操作，然后单击**下一步**。
6. 配置操作属性并为成功解析的数据选择输出端口。
7. 或者，单击**下一步**向策略中添加更多操作。
8. 将所有操作添加到策略后，单击**完成**。
9. 或者，向转换中添加更多策略。
10. 或者，更改转换处理策略和操作的顺序。 选择某个策略或操作，然后单击**上移**或**下移**。

配置模式解析策略

要配置模式解析策略，请在模式解析模式下打开解析器转换，然后选择**模式**视图。

在将转换配置为解析模式之前，请验证**模式**视图是否显示所需的输出端口名称。 解析器转换会将标志解析到所选的输出端口中。 创建其他输出端口（如果需要）。

1. 选择**模式**视图。
2. 将一个或多个模式添加到策略中。 您可以通过以下方式添加模式：
   - 输入数据值以创建模式。 单击**新建**，然后选择**新建模式**。

     如果选择**新建模式**，请单击**在此处输入模式**，然后输入一个或多个标志类型。输入的标志必须与输入数据字段中的标志结构相匹配。 添加描述输入端口中的标志结构所需的模式。
   - 从引用表中导入数据值。 单击**新建**，然后选择**新建引用表**。

     如果选择**新建引用表**，请浏览模型存储库并选择包含标志结构列表的引用表。引用表必须包含两列。引用表中的第二列必须包含数值。
   - 从模式集中导入数据值。 单击**导入**，然后在模型存储库中选择一个可重用的模式集。

     如果选择**导入**，请浏览模型存储库中的内容集，然后选择一个可重用的模式集。

   注意: 您可以使用**筛选文本**字段筛选引用表和模式集的列表。

   在“模式”列中可以混合使用模式集和引用表。

3. 将“模式”列中的每个标志分配给一个输出端口。
   - 要将标志分配给输出端口，请在端口列中双击，然后从菜单中选择标志名称。
   - 要将多个标志解析到一个输出中，请在端口列中双击，然后选择**自定义**。 将多个标志分配给一个端口并选择要使用的分隔符。

   将模式中每行中的标志分配给一个或多个输出端口。
4. 保存转换。

解析转换高级属性

配置有助于确定数据集成服务如何为解析器转换处理数据的属性。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的解析器转换

非本地环境中的解析器转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射和流映射中无限支持。
- Databricks Spark 引擎。无限支持。

第 35 章

Python 转换

Python 转换提供了使用 Python 代码定义转换功能的界面。

Python 是一种使用简单语法、动态键入和动态绑定的语言，这使 Python 成为提高工作效率或参与快速应用程序开发的理想选择。在数据工程映射中使用 Python 代码时，Python 代码被嵌入到生成的 Scala 代码中，Spark 或 Databricks Spark 引擎运行该代码来处理大型、多样化和快速变化的数据集。

您还可以使用 Python 转换进行机器学习。在转换中，可以指定包含预训练模型的资源文件，并在 Python 代码中加载预训练模型。例如，可以加载一个预训练模型来对输入数据进行分类或创建预测。

在使用 Python 转换之前，在 Hadoop 连接或 Databricks 连接属性中配置相应的 Spark 高级属性。然后，确保群集上的工作节点包含 Python 的安装。

有关安装 Python 的详细信息，请参阅《*Data Engineering Integration 指南*》。

只能在 Spark 或 Databricks Spark 引擎上运行 Python 转换。不能在本地环境中运行 Python 转换。

有关 Python 转换的详细信息，请参阅《*Data Engineering Integration 用户指南*》。

第 36 章

等级转换

本章包括以下主题：

等级转换概览

等级转换是一种主动转换，用于将记录限制为最高或最低范围中的值。 可以使用等级转换返回端口或组中的最大或最小数值。 也可以使用等级转换返回映射排序中排在最前或最后的字符串。

在映射运行期间，数据集成服务缓存输入数据直到可以执行等级计算。

等级转换与转换函数 MAX 和 MIN 不同。 等级转换返回一组最高值或最低值，而不是只返回一个值。 例如，使用等级转换可以选择给定区域中的前 10 名销售员。 等级转换也可用于生成财务报表，可以使用该转换找出薪金和管理费用最低的三个部门。 尽管 SQL 语言提供了许多用于处理成组数据的函数，但是无法使用标准 SQL 函数在一组行中确定最高或最低等级。

可以将表示同一行集的所有端口连接到该转换。落在该等级（基于配置转换时设置的某个度量）内的行将传递给等级转换。

作为一种主动转换，等级转换可能会更改传递它的行的数量。您可能将 100 行传递给等级转换，但只选择排名前 10 位的行。 前 10 行将从等级转换传递到另一转换。

可以将某一转换的端口连接到等级转换。 还可以创建局部变量和编写非汇总表达式。

为字符串值评级

可以配置等级转换以返回字符串端口的最高值或最低值。 数据集成服务根据为部署的映射选择的排序顺序对字符串排序。

配置包含映射的应用程序时，选择数据集成服务用于运行映射的排序顺序。 可以选择二进制排序或按特定语言（如法语或德语）排序。 如果选择二进制排序，数据集成服务将计算每个字符串的二进制值，并使用二进制值对字符串排序。 如果选择某种语言，数据集成服务将使用该语言的排序顺序按字母顺序对字符串排序。

等级转换属性

创建等级转换时，可以配置以下属性：

- 输入缓存目录。
- 选择最高或最低等级。
- 选择用于确定等级的值所在的输入/输出端口。 只能选择一个端口来定义等级。
- 选择要评级的行数。
- 定义等级组，如每个制造商的价格最低的 10 种产品。

动态映射中的等级转换

您可以在动态映射中使用等级转换。可以在转换中配置动态端口，并引用生成的端口。

如果在等级转换中引用了某个生成的端口，但该生成的端口在运行时不存在，映射将失败。

如果您将某个动态端口指定为“等级”端口，则该动态端口最多只能包含 1 个生成的端口。

如果您将某个动态端口指定为“分组依据”端口，则数据集成服务会将所有生成的端口都视为“分组依据”端口。如果您将某个生成的端口指定为“分组依据”端口，并将父动态端口指定为“等级”端口或“分组依据”端口，则映射无效。

您可以对“等级”端口和“分组依据”端口进行参数化。对“等级”端口使用端口类型参数，对“分组依据”端口使用端口列表类型参数。

等级转换端口

等级转换可以包括连接到映射中的另一转换的输入端口、输入/输出端口或输出端口。该转换还可以包括传递端口和变量端口。

等级转换具有以下端口类型：

输入

从上游转换接收数据。可以将输入端口指定为输入/输出端口。该转换必须至少有一个输入端口。

动态端口

可以接收多个列以创建动态数量的生成端口的端口。生成的端口是动态端口中表示单一列的端口。可以创建输入、输出和变量动态端口。

输出

将数据传递到下游转换。可以将输出端口指定为输入/输出端口。该转换必须至少有一个输出端口。

传递

原样传递数据。

变量

用于局部变量。可以使用变量端口存储要在表达式中使用的值或计算。变量端口不能是输入端口或输出端口。变量端口在该转换内传递数据。

等级索引

Developer 工具会为每个等级转换创建一个 RANKINDEX 端口。 数据集成服务将使用该等级索引端口来存储组中每一行的等级位置。

例如，您可能会创建一个等级转换来标识公司中薪资最高的前 50 位员工。 可以将 SALARY 列标识为用于度量等级的输入/输出端口，并配置该转换，使其筛选掉除前 50 位员工之外的所有行。

在等级转换标识了属于最高或最低等级的所有行之后，便可分配等级索引值。 对于薪资最高的前 50 位员工，薪资最高的员工的等级索引为 1。 薪资第二位的员工的等级索引为 2，以此类推。 在度量最低等级（例如，清单中价格最低的 10 个产品）时，等级转换会按从低到高的顺序分配等级索引。 因此，价格最低的产品的等级索引为 1。

如果两个等级值匹配，则它们将获得相同的等级索引值，因此，转换将跳过下一个值。 例如，如果要查看某个国家/地区排名前五位的零售商店，并且其中两家商店的销售额相同，则返回数据可能类似于以下形式：

RANKINDEX	SALES	STORE
1	10000	Orange
1	10000	Brea
3	90000	Los Angeles
4	80000	Ventura

RANKINDEX 仅用作输出端口。 可以将等级索引传递给映射中的其他转换，或者直接传递给目标。

等级端口

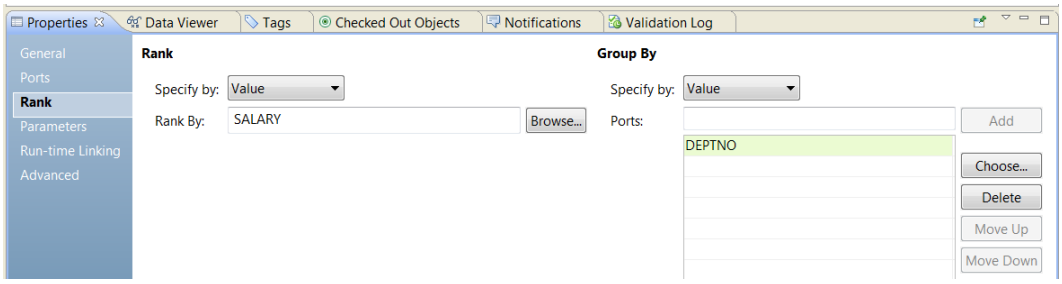
等级端口确定为值划分等级所依据的列。

必须指定一个输入/输出或输出端口作为等级端口。例如，创建一个等级转换以基于薪资对每个部门的前几名员工进行等级划分。薪资端口包含每个员工的薪资。将薪资输入/输出端口指定为等级端口。

在**属性**视图的**等级**选项卡上选择等级端口。可以对等级端口使用参数。要使用参数，请选择**按参数指定**。浏览端口参数或创建端口参数。参数默认值为端口或生成的端口的名称。

必须将等级端口链接到另一个转换。

下图显示了**等级**选项卡：



注意: 等级端口不支持二进制数据类型。

定义分组依据端口

可以配置等级转换，以便为分级行创建组。

例如，如果要按制造商选择最昂贵的 10 个产品，则可能首先要为每个制造商定义一个组。在**等级**选项卡的**分组依据**面板上，可以设置一个输入、输入/输出或输出端口作为分组依据端口。

对于该组端口中的每个唯一值，该转换将创建一组符合等级定义的行（每个等级的最高点或最低点以及某一特定点）。

等级转换将以两种不同的方式更改行数。 通过筛选除最高等级或最低等级的行之外的所有行，可以减少传递给转换的行数。通过定义组，可为每个组创建一组分级的行。

例如，如果要创建一个等级转换，按季度为前五名销售人员分级，则等级索引将在每个季度对销售人员进行编号，编号依次为 1 至 5：

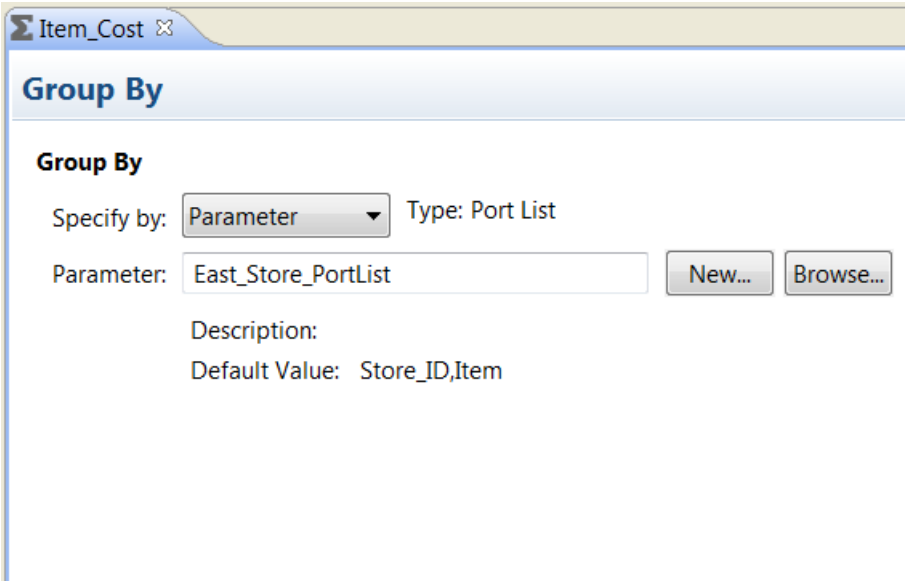
RANKINDEX	SALES_PERSON	SALES	QUARTER
1	Sam	10,000	1
2	Mary	9,000	1
3	Alice	8,000	1
4	Ron	7,000	1
5	Alex	6,000	1

在**属性**视图的**高级**选项卡上设置要包括在等级中的行数。

分组依据参数

可以配置一个端口列表参数，其中包含要加入到组中的一个或多个端口。 通过在转换中从端口列表选择端口，可以创建端口列表参数。

下图显示了在您使用参数来标识组中的端口时出现的**分组依据**选项卡：



可以浏览端口列表参数，也可以单击**新建**创建端口列表参数。 如果选择创建端口列表参数，则可在转换中从端口列表选择端口。

等级缓存

当您运行使用等级转换的映射时，数据集成服务将在内存中创建索引缓存和数据缓存以运行该转换。如果数据集成服务需要的空间大于内存缓存中的可用空间，它会将溢出数据存储在缓存文件中。

当您运行使用等级转换的映射时，数据集成服务会将输入行与数据缓存中的行进行比较。如果输入行的等级高于缓存行，则数据集成服务会将缓存行替换为输入行。 如果通过配置等级转换来对行进行分组，则数据集成服务会对每个组中的行进行等级划分。

数据集成服务为等级转换创建以下缓存：

- 索引缓存，按照分组依据端口中的配置存储组值。
- 数据缓存，按照分组依据端口存储信息。

等级转换高级属性

配置有助于确定数据集成服务如何为等级转换处理数据的属性。

在**高级**选项卡上配置以下属性：

Top/Bottom

指定是否要获取列的最高或最低等级。

等级数

要包括在最高或最低等级中的行数。

区分大小写的字符串比较

指定数据集成服务在确定字符串等级时是否使用区分大小写的字符串比较。 清除此选项可使数据集成服务忽略字符串的大小写。 默认情况下，此选项被选中。

缓存目录

数据集成服务创建索引缓存文件和数据缓存文件的目录。请确认该目录存在并含有足够的磁盘空间可用于缓存文件。

输入多个以分号分隔的目录，以提高缓存分区期间的性能。缓存分区会为处理转换的每个分区创建一个单独的缓存。

默认值为 CacheDir 系统参数。您可以为此属性配置另一个系统参数或用户定义的参数。

等级数据缓存大小

数据集成服务在映射运行开始时为转换的数据缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

等级索引缓存大小

数据集成服务在映射运行开始时为转换的索引缓存分配的内存量。选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

相关主题：

- “缓存大小” 页面上 67

非本地环境下的等级转换

非本地环境下的等级转换处理取决于运行此转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的等级转换

Blaze 引擎的某些处理规则与数据集成服务的处理规则不同。

等级转换的数据缓存已经过优化，可以使用可变长度存储通过等级转换的 binary 和 string 数据类型。系统对最大为 8 MB 的记录启用此优化。如果记录大小大于 8 MB，则会禁用可变长度优化。

当使用可变长度在数据缓存中存储传递等级转换的数据时，会将等级转换优化为使用已排序输入，且在运行时映射中在等级转换前插入一个传递排序器转换。要查看排序器转换，请查看优化的映射或查看 Blaze 验证环境中的执行计划。

在数据缓存优化期间，等级转换的数据缓存和索引缓存设置为“自动”。排序器转换的排序器缓存与等级转换的数据缓存设置为相同大小。要配置排序器缓存，必须配置等级转换的数据缓存大小。

Spark 引擎上的等级转换

Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

在下列情况下，映射验证会失败：

- 区分大小写处于禁用状态。

数据缓存优化

无法将转换的数据缓存优化为使用可变长度存储数据。

流映射中的等级转换

流映射具有不适用于批处理映射的其他处理规则。

映射验证

在下列情况下，映射验证会失败：

- 等级转换与配置了不相等查找条件的被动查找转换处于同一流管道中。
- 等级转换在联接器转换的上游。
- 流管道中包含多个等级转换。
- 流管道中包含汇总器转换和等级转换。

Databricks Spark 引擎上的等级转换

Databricks Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

在下列情况下，映射验证会失败：

- 区分大小写处于禁用状态。

数据缓存优化

无法将转换的数据缓存优化为使用可变长度存储数据。

第 37 章

读取转换

本章包括以下主题：

读取转换概览

读取转换是在从源读取数据时发生的被动转换。读取转换不可重用。

您可以从物理数据对象或逻辑数据对象创建读取转换。如果要基于从 PowerExchange® 适配器源导入的物理数据对象创建读取转换，映射编辑器可能会提示您必须指定读取操作才能基于该数据对象创建读取转换。

可以根据用于创建读取转换的数据对象类型为该转换配置不同的属性。例如，如果从关系数据对象创建读取转换，则可以配置 SQL 替代并定义约束。可以配置的属性还取决于您是否已为转换配置参数。

读取转换可以包含动态源。可将读取转换配置为动态更新其端口、元数据和其他属性。有关配置动态源的信息，请阅读*《Informatica Developer 映射指南》*中的“动态映射”一章。

读取转换属性

创建读取转换后，可为其配置属性。

在属性选项卡上配置读取转换属性。 您可以使用的选项卡取决于读取转换所表示的源的类型。

下表介绍了每个属性选项卡并标识了对应选项卡所针对的源类型：

属性选项卡	说明	源类型
常规	指定转换属性和行为。 对于关系和自定义数据对象源，将转换输入端口与源同步。	全部
数据对象	指定转换数据源。	- 关系 - 平面文件 - 自定义数据对象
端口	按关联的数据对象设置端口定义。	- 关系 - 自定义数据对象 - 逻辑数据对象
格式	平面文件数据源的输入设置	平面文件
查询	指定源查询。	- 关系 - 自定义数据对象 - 逻辑数据对象
运行时	定义运行时行为。	- 关系 - 平面文件 - 自定义数据对象
源	选择源表并配置源详细信息。	- 关系 - 自定义数据对象
数据对象参数	设置参数属性。	- 平面文件 - 自定义数据对象 - 逻辑数据对象
运行时链接	配置转换之间使用参数和/或链接策略确定在运行时要链接哪些端口的组到组链接。	全部
高级	设置跟踪级别和行顺序。 对于关系源，设置在运行时创建或替换目标表的选项。	全部

常规属性

您可以配置读取转换的名称和说明。 还可以配置以下属性：

列元数据更改时：

可用于关系源。 选择以下选项之一：

- 同步输出端口。 Developer tool 使用模型存储库为数据对象所存储的元数据更改来更新读取转换输出端口。
- 不同步。 读取转换不显示数据对象中的元数据更改。

物理数据对象

可用于平面文件和自定义源。 用于创建转换的对象。

您可以选择数据对象名称并配置其属性。

数据对象属性

在“数据对象”选项卡上，可以指定或更改读取转换源并使关系、平面文件和自定义数据对象源动态化。

可以配置以下属性：

指定依据

要指定读取转换的源列和元数据，请选择下列选项之一：

- 值。 读取转换使用关联的数据对象指定源列和元数据。
- 参数。 读取转换使用参数指定源列和元数据。

数据对象

如果是从现有数据对象创建的读取转换，则该字段会显示对象的名称。 单击**浏览**可将数据对象更改为与读取转换相关联。

在运行时，从数据源获取数据对象列

启用此选项时，数据集成服务会将元数据和数据定义更改从源表提取到读取转换。

查询属性

配置关系资源或自定义数据对象的 SQL 查询。

在**查询**选项卡上配置属性时，可选择配置简单或高级属性。

在**简单**属性视图中，您将默认 SQL 语句配置为 Define Distinct 语句并编辑该语句的提示、联接、筛选和排序条件。

在**高级**属性视图中，可以定义自定义 SQL 查询。 您可以从关联数据对象的列中或从参数中进行选择，也可以创建一个新参数来表示数据对象。

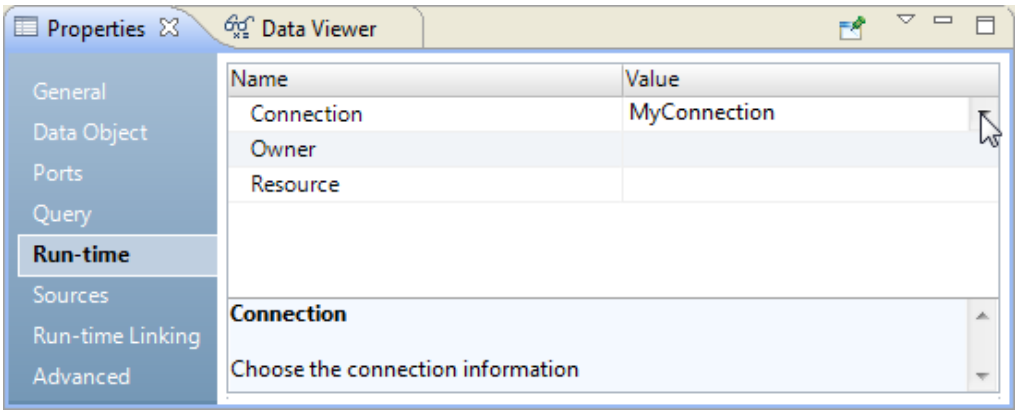
运行时属性

可以在**运行时**选项卡上配置以下读取转换属性：

连接

可用于关系源。 转换所使用的连接。 单击字段右侧可更改连接。

下图显示了要单击的下拉按钮的位置：



源属性

为关系资源和自定义数据对象配置源的详细信息。 将关系数据对象导入到存储库后，您可以更改其定义。 您可以在存储库中添加和删除端口，还可以定义主键以及配置多个关系数据对象之间的关系。

源选项卡可以配置以下设置：

所有源

使用“添加”和“删除”按钮可添加和删除转换的其他源。

“常规”选项卡

更改所选源的名称和说明。 单击源名称可更改其他详细信息。

“键”选项卡

将资源列指定为键。

“关系”选项卡

添加和删除多个关系资源之间的关系。

高级属性

配置高级属性以确定数据集成服务如何为读取转换处理数据。

在高级选项卡上配置以下属性：

跟踪级别

控制映射日志文件中的详细信息量。

PreSQL

数据集成服务在读取源之前对源数据库运行的 SQL 命令。

Developer tool 不会验证 SQL。

PostSQL

数据集成服务在写入目标后对源数据库运行的 SQL 命令。

Developer tool 不会验证 SQL。

约束

用于表级引用完整性约束的 SQL 语句。仅适用于关系源。

同步关系数据对象

可以在物理数据对象的源更改时同步这些物理数据对象。同步物理数据对象时，Developer tool 将从所选源重新导入对象元数据。

可以同步所有物理数据对象。同步关系数据对象或自定义数据对象时，可以保留或覆盖您在 Developer tool 中定义的键关系。

要同步映射对象，可从多种方法中进行选择：

同步关系资源。

要同步任何物理数据对象，请在**对象浏览器**视图中右键单击对象，然后选择**同步**。

将转换端口与物理数据对象同步。

在转换的“数据对象”选项卡中，选择**运行时，从数据源获取数据对象列**选项。

在运行时，“数据集成服务”会从数据源提取元数据和数据定义更改，并刷新模型存储库中的数据对象定义。

要预览数据集成服务如何提取元数据和数据定义更改，使用已解析的参数查看映射。

元数据更改时同步端口

在转换的“常规”选项卡中，选择用于同步端口的选项。此选项的确切标签取决于您配置的转换的类型。例如，对于读取转换，该选项为**元数据更改时同步输出端口**。

在映射运行时，数据集成服务会将转换中的列元数据与数据源中的元数据同步。

更改源数据对象

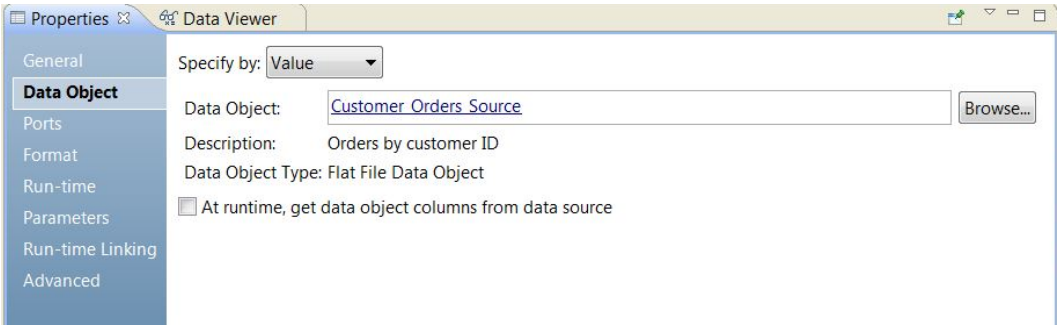
读取转换以模型存储库中的物理数据对象或逻辑数据对象为基础。 您可以在配置读取转换时更改数据对象。 可以参数化数据对象以在运行时更改数据对象。 例如，您可能会使用与生产映射运行所使用的源文件不同的源文件来测试映射。

从物理数据对象创建转换时，有关数据对象的信息将显示在转换属性的**数据对象**选项卡上。您可以单击数据对象名称来查看模型存储库中的物理数据对象定义。

您可以通过浏览到模型存储库中的其他物理数据对象来更改转换的数据对象。当您更改数据对象时，转换将使用您选择的数据对象的运行时属性和高级属性。

可以在运行时根据数据源更改来更新数据对象的结构。数据源是数据对象所表示的物理文件或数据库表。启用数据集成服务从数据源获取数据列时，数据集成服务会检查数据源的结构。数据集成服务会根据数据源来更新转换实例中的数据对象端口，但不会更改模型存储库中的物理数据对象定义。

下图显示了**数据对象**选项卡：



数据对象选项卡具有以下字段：

指定依据

选择**值**可输入特定数据对象名称。选择**参数**可参数化数据对象。

数据对象

模型存储库中的数据对象的名称。您可以单击**数据对象**链接来打开存储库中的数据对象定义。还可以浏览模型存储库中的其他数据对象。

说明

存储库中的数据对象的说明。只读。

数据对象类型

描述数据对象的类型，如平面文件数据对象、关系表对象或自定义数据对象。

运行时，从数据源获取数据对象列

数据集成服务从数据对象所引用的数据文件或表中获取元数据和数据定义更改，并在运行时更新转换实例的数据对象结构。

要预览数据集成服务如何在运行时获取元数据和数据定义更改，请查看具有已解析参数的映射。

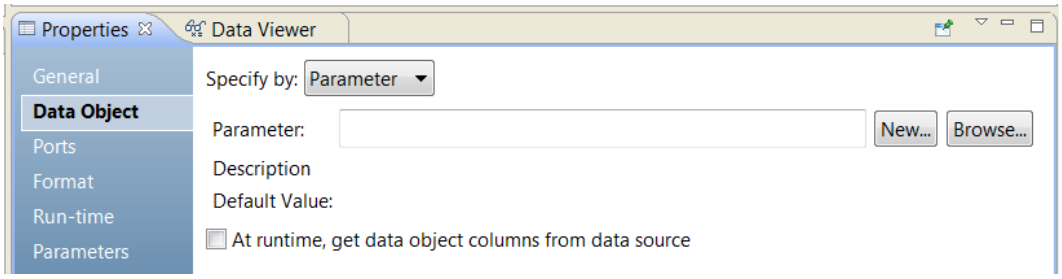
将读取转换参数化

可以将读取转换参数化并在运行时更改数据对象。

要参数化某个数据对象，请在**数据对象**选项卡中选择**按参数指定**。 **数据对象**选项卡中的属性发生变化。

要参数化该数据对象，请创建一个资源类型参数或浏览查找已创建的资源参数。 参数默认值是模型存储库中物理数据对象的名称。 创建默认参数值时，需要从存储库中的数据对象列表中选择物理数据对象。

下图显示了按参数指定数据对象时出现的**数据对象**选项卡。



数据对象选项卡按参数列出以下选项：

参数

您配置为数据对象的资源参数的名称。 只读。

说明

参数的说明。 只读。

新建

创建资源参数。 在模型存储库中浏览参数默认值并选择数据对象。

浏览

浏览资源参数并选择该参数。

默认值

您为数据对象配置的资源参数的默认值。 默认值是物理数据对象名称以及该对象在模型存储库中的路径。 只读。

读取转换参数

您可以参数化读取转换的某些属性以及从其创建读取转换的可重用物理数据对象的某些属性。

创建物理数据对象时，您将配置读取和写入属性。向映射添加物理数据对象时，您在该数据对象中为读取属性配置的参数会显示在**数据对象参数**选项卡中。

可在物理数据对象中为以下读取属性配置参数：

- 控制文件目录
- 控制文件名
- 默认小数位数
- 分隔符
- 平面文件分隔符
- 合并文件目录
- 源文件名
- 源文件目录

向映射添加物理数据对象时，可以在读取转换的**数据对象参数**选项卡中查看参数。您可以将这些参数公开为映射参数以在运行时替代参数值。

注意: 无法将用户定义的参数嵌套在参数化源内。如果源数据对象已参数化，则无法将用户定义的参数公开为映射参数而在运行时替代参数值。映射会改为使用默认值。

可以为读取转换配置以下映射参数：

- 连接（关系）
- 数据对象
- 链接解析顺序
- 资源名称（关系）
- 表所有者名称（关系）

可以在映射的**数据对象参数**选项卡中查看这些参数。

约束

约束是数据行上的值必须满足的条件表达式。

当您设置约束时，需要输入对每个数据行计算结果均为 TRUE 的表达式。

数据集成服务可以读取关系源、逻辑数据对象、物理数据对象或虚拟表中的约束。 要设置可重用物理数据对象的约束，请创建一个自定义数据对象。

数据集成服务读取约束时，它可能会根据应用的优化方法删除数据行中未计算为 TRUE 的行。

在设置约束前，您必须验证源数据是否满足约束所设置的条件。 例如，源数据库有一个 AGE 列，其中多个行的 AGE 均小于 70。您可以在源数据库上将约束设置为 AGE < 70。这样，数据集成就会使用 AGE < 70 这一约束从源数据库读取记录。如果数据集成服务读取到 AGE >= 70 的记录，则会删除 AGE >= 70 的行。

在数据库中，当您连接到数据库时，可以使用 SQL 命令来设置数据库环境的约束。数据集成服务在每次连接到数据库时运行连接环境 SQL。

创建读取转换

创建读取转换时，需要根据用于创建转换的资源来选择下列方法之一：

从模型存储库中的数据对象创建转换。

执行以下步骤从模型存储库中的数据对象创建转换：

1. 在编辑器中打开映射。
2. 将数据对象从**对象浏览器**拖动到编辑器视图。
3. 选择**读取**并单击**确定**。
   映射中的读取转换包含数据对象的端口和属性。

使用映射编辑器创建转换。

如果希望配置详细的读取转换设置，请使用此方法。 如果要基于参数进行读取转换，则也可以使用此方法。

要在映射编辑器中创建读取转换，请参阅“在映射编辑器中创建读取转换” 页面上 487。

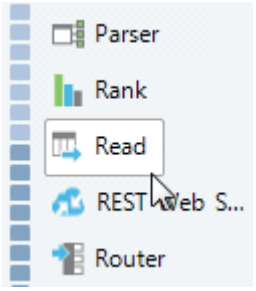
在映射编辑器中创建读取转换

可以创建读取转换来表示映射中的数据源、列元数据和属性。

执行以下步骤：

1. 选择以下方法之一创建读取转换：
   - 在映射编辑器中右键单击，然后选择**添加转换**。
     此时将打开**添加转换**对话框。

     选择读取转换，然后单击**下一步**。
   - 在映射选项板中向下滚动，找到读取转换图标后双击该图标。
     下图显示了读取转换图标：

   将打开**新建读取转换**对话框。
2. 要将平面文件、关系资源或自定义数据对象用作源，请执行以下步骤：
   a. 选择**物理数据对象**作为数据对象类型。
   b. 单击**浏览**选择平面文件、关系资源或自定义数据对象。

      此时将打开**选择数据对象**窗口。
   c. 选择一个数据对象，然后单击**确定**。
   d. （可选）将转换配置为在运行时从源提取数据对象列。选择**在运行时，从数据源获取数据对象列**。

      映射运行时，数据集成服务将刷新读取转换的列元数据。
3. 要将参数用作源，请执行以下步骤：
   a. 选择**使用参数创建**。

b. 单击**新建**创建一个新参数，或单击**浏览**选择现有参数。

c. 选择一个参数，然后单击**确定**。

d. （可选）将转换配置为在运行时从源提取数据对象列。选择**在运行时，从数据源获取数据对象列**。映射运行时，数据集成服务将刷新读取转换的列元数据。

4. 要将逻辑数据对象用作源，请执行以下步骤：

a. 选择**逻辑数据对象**作为数据对象类型。

b. 单击**浏览**选择数据对象，然后单击**确定**。

5. （可选）键入读取转换的名称。

6. 单击**完成**。

第 38 章

关系到层次结构转换

本章包括以下主题：

关系到层次结构转换概览

关系到层次结构转换会对关系输入进行处理，并将其转换为层次结构输出。关系到层次结构转换从输入端口读取关系输入，并在转换输出端口将数据转换为层次结构输出。要将关系输入转换为层次结构输出，请使用一个架构对象来定义层次结构。

您可以使用“关系到层次结构转换”向导来创建可反映关系输入端口的层次结构。您可以在转换的**概览**视图中查看层次结构输出端口的映射。

创建转换后，您即可将数据从层次结构输出端口传递到映射中的另一个转换。

注意: 关系到层次结构转换最多可以在一个 .xsd 文件中处理 10,000 个架构元素。要处理超过 10,000 个元素，请将数据拆分为多个文件。

在关系模型中，每个表架构都会确定一个称为主键的列，以唯一地标识每一行。使用外键可以标识表中各行之间以及与其他表中某行的关系。向导在创建转换时会生成一些键。您可以更改自动生成的转换，以及添加、编辑或删除端口。

您可以将输入关系端口链接到层次结构中的节点。将层次结构中的相关元素或属性的主键链接至输入中的关系组。主键标识关系表中的各行。

将层次结构中的相关元素或属性的外键链接至输入中的关系组。关系输入中的外键是表中指向其他表主键的一列。

输入关系端口和层次结构节点必须使用兼容的数据类型。

示例 - 关系到层次结构转换

Electronics Superstore 公司的财务部门必须处理公司员工的工资。他们需要将存储在关系数据库中的员工数据转换为其支付系统能够处理的层次结构格式。

映射需要使用关系到层次结构转换，该转换输入员工详细信息（如员工姓名、员工 ID、员工地址和员工银行帐户数据），并采用某种可用的层次结构格式输出这些详细信息。

在关系输入中，Bank_ID 元素是 Employee 表的主键，Bank 表的外键：

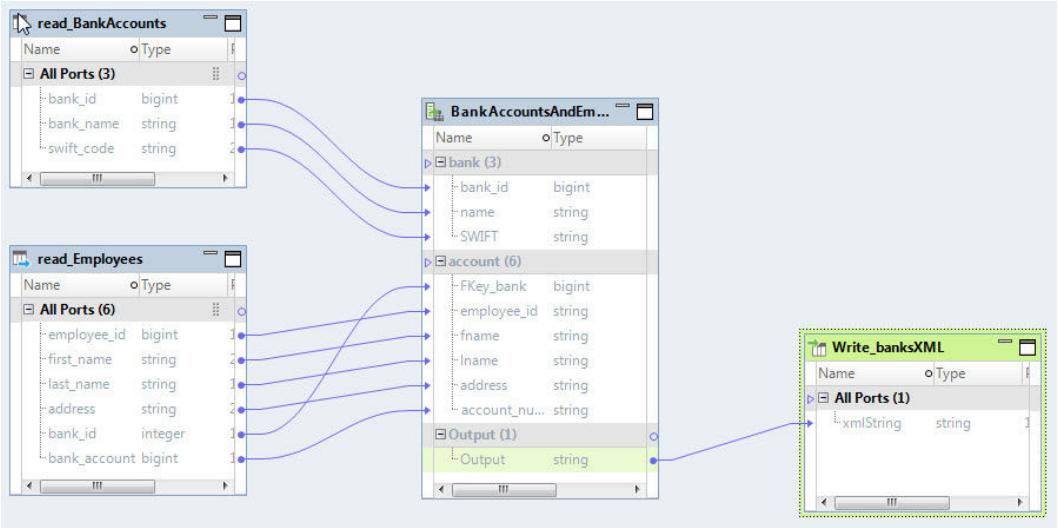
Employee_ID	Last_Name	First_Name	Address	Bank_ID	Bank_Account
9173327437	Sandrine	Jacques	74 Mobile Avenue	74845	8723487234
9174562342	Race	Tom	266 Crouse St.	9234734	45324734
8484526471	Jones	Charles	3815 LaValle Boulevard	389236	234638437
7023847265	Smith	Delilah	193 Short Drive	74845	8723463432
9174596725	Frederick	George	17 Serenity Road	9234734	6342636699

Bank_ID	Bank_Name	SWIFT_Code
74845	National Bank	9173327
9234734	International Bank	9174562
389236	Star National Bank	8484526

在采用层次结构格式的 Payment 输出中，这些表中的元素被组合在一起：

```
<banks>
    <bank name="National Bank" SWIFT="9173327">
        <account id="8723487234">
            <employee_id>9173327437</employee_id>
            <fname>Sandrine</fname>
            <lname>Jacques</lname>
            <address>74 Mobile Avenue</address>
        </account>
        <account id="8723463432">
            <employee_id>9082745558</employee_id>
            <fname>Delilah</fname>
            <lname>Smith</lname>
            <address>193 Short Drive</address>
        </account>
    </bank>
    <bank name="International Bank" SWIFT="9174562">
        <accounts>
            <account id="45324734">
                <employee_id>5534398889</employee_id>
                <fname>Race</fname>
                <lname>Tom</lname>
                <address>266 Crouse St.</address>
            </account>
            <account id="6342636699">
                <employee_id>9174596725</employee_id>
                <fname>Frederick</fname>
                <lname>George</lname>
                <address>17 Serenity Road</address>
            </account>
        </accounts>
    </bank>
    <bank name="Star National Bank" SWIFT="8484526">
        <accounts>
            <account id="234638437">
                <employee_id>8484526471</employee_id>
                <fname>Jones</fname>
                <lname>Charles</lname>
                <address>3815 LaValle Boulevard</address>
            </account>
        </accounts>
    </bank>
</banks>
```

下图显示了本示例中的映射：



该映射包含以下对象：

Read_BankAccounts

包含银行数据的源。

Read_Employees

包含员工数据的源。

BankAccountsAndEmployees_To_PaymentsSystemXML

关系到层次结构转换，该转换将包含员工和银行帐户信息的关系输入，转换为支付系统能够处理的一种 XML 格式。

Write_BanksXML

每次运行映射时用来存储转换后数据的文件的目标路径。

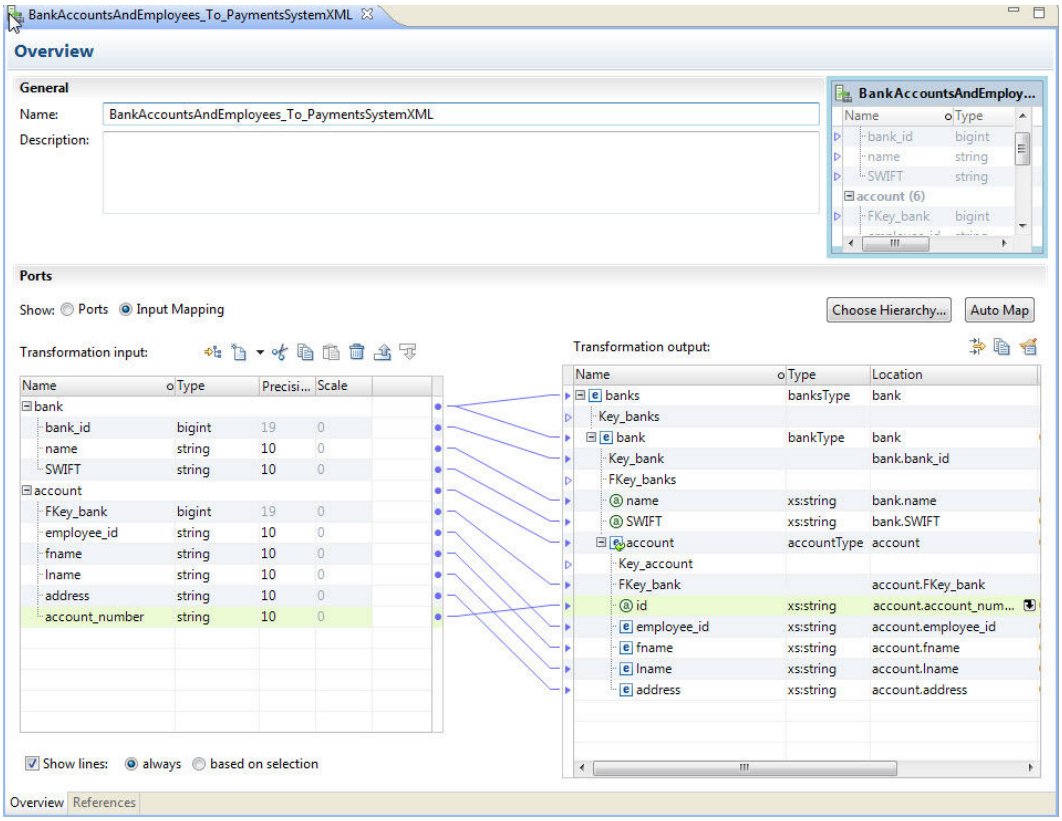
该映射使用 Read_BankAccount 和 Read_Employees 文件来提供关系输入。该映射通过 BankAccountsAndEmployees_To_PaymentsSystemXML 转换来处理和转换数据，然后将输出存储在 Write_BanksXML 平面文件中列出的目标路径下。

输入关系端口与“概览”视图

为了将关系数据转换为层次结构数据，向导会创建一个反映关系输入端口的层次结构。您可以使用**概览**视图将关系端口链接至层次结构端口。

要查看关系输入与层次结构输出之间的链接，请使用**概览**视图。选择**输入映射**。**端口**面板将显示在**概览**视图中。

下图显示了**端口**面板：



端口面板的左侧是**转换输入**区域，其中包含关系元素和组。右侧则是**转换输出**区域，其中包含层次结构架构节点。

可以在**转换输入**区域中创建端口，然后将关系元素链接至架构节点。还可以将指针从架构中的节点拖至**转换输入**区域中的空字段以创建端口。将关系端口连接至架构节点时，Developer tool 将显示两者间的链接。

关系到层次结构转换端口

在转换的**概览**视图中定义了关系到层次结构转换端口。

关系到层次结构转换可以从缓冲区中读取关系数据输出。输出端口会将层次结构数据返回给缓冲区。

架构引用

关系到层次结构转换需要使用一个层次结构架构来定义转换中的输出层次结构。要在转换中使用架构，可以定义架构引用。

可以在转换的**引用**视图中定义转换架构引用。

关系到层次结构转换会引用模型存储库中的架构对象。在创建转换之前，存储库中可以存在架构对象。

一个架构可引用其他多个架构。**引用**视图显示了关系到层次结构转换所引用的每个架构的命名空间和前缀。

关系到层次结构转换开发

使用“新建转换”向导自动生成关系到层次结构转换。选择一个架构或层次结构示例文件来定义输出层次结构。

创建关系到层次结构转换

1. 在 Developer tool 中，依次单击**文件** > **新建** > **转换**。
2. 选择关系到层次结构转换，然后单击**下一步**。
3. 输入转换的名称并浏览要放置该转换的模型存储库位置，然后单击**下一步**。
4. 要选择一个架构，请选用下列方法之一：
   - 要使用模型存储库中的某个架构来定义输出层次结构，请在**架构对象**字段的附近，浏览存储库以从中选择该架构文件。
   - 要导入架构文件，请单击**创建新的架构对象**。在**新建架构对象**窗口中，您可以浏览并选择一个架构文件，也可以选择通过某个示例层次结构文件创建一个架构。
5. 选择输出层次结构的根元素。在**层次结构根元素**对话框中，选择架构中作为输出层次结构文件的根元素的元素。为了帮助选择根对象，您可以添加一个示例层次结构文件。要添加一个示例文件，请在**示例文件**字段的附近，浏览文件系统并从中选择该文件。
6. 单击**完成**。

   向导将在存储库中创建该转换。

创建端口

在**概览**视图中配置端口。

1. 要查看映射，请在**概览**视图的**端口**区域中，选择**输入映射**。
2. 选择输入端口类型、精度和小数位数。
3. 展开**端口**网格中的树。在左侧，**转换输入**面板会显示关系输入；在右侧，**转换输出**面板会显示预期的层次结构输出。
4. 要将节点定义为根元素，请单击**选择层次结构**。

   Developer tool 仅在**转换输入**区域中显示根级别中的节点以及根级别下的节点。
5. 要查看连接端口与层次结构节点的行，请单击**显示行**。选择查看所有连接行或仅查看选定端口的行。
6. 要将输入组或端口添加到**转换输入**区域，请使用以下方法之一：
   - 将**转换输出**区域中的一个简单或复杂元素拖至**转换输入**区域的某个空列中。如果该节点是组节点，则 Developer tool 将添加关系组，而不添加端口。
   - 要添加关系组，请选择一行，右键单击并选择**新建** > **组**。
   - 要添加关系端口，请右键单击并选择**新建** > **字段**。
7. 要清除层次结构节点设置的端口位置，请使用以下方法之一：
   - 在**转换输出**区域中选择一个或多个节点，然后右键单击并选择**清除**。
   - 选择将关系输入端口连接到层次结构节点的一行或多行，然后右键单击并选择**删除**。
8. 要显示层次结构中的输出端口，请单击**显示为层次结构**。每个子组都显示在其父组下方。

第 39 章

REST Web 服务使用者转换

本章包括以下主题：

REST Web 服务使用者转换概览

REST Web 服务使用者转换是一种主动转换，其作为 Web 服务客户端连接到 REST Web 服务，以访问或转换数据。 使用 REST Web 服务使用者转换可连接到 REST Web 服务。 REST Web 服务使用者转换可以向 REST Web 服务发送请求以及从 REST Web 服务接收响应。

REST Web 服务使用者转换通过您在该转换中或在 HTTP 连接中定义的 URL 连接到 Web 服务。 还可以使用 HTTPS 连接。REST Web 服务使用者转换可以使用 TLS 1.2、TLS 1.1 或 TLS 1.0。

对于 Web 服务所支持的每一个操作，REST Web 服务均包含一个 HTTP 方法。 当数据集成服务连接到 REST Web 服务时，其可以发送获取数据、发布数据、放置数据或删除数据的请求。 该请求可对单个资源或资源集合执行。 数据集成服务发送请求消息后，其从 Web 服务接收响应消息。

请求和响应消息包含 XML 或 JSON 数据，其中包含的元素可形成一个层次结构。 如果请求或响应消息包含多次出现元素，则元素组将形成 XML 或 JSON 层次结构中的级别。 如果一个级别嵌在另一个级别中，则组之间产生关联。

在 REST Web 服务使用者转换中，方法输入和方法输出定义请求和响应消息的结构。 方法输入和方法输出包含一些映射，这些映射定义如何将消息元素映射到输入和输出端口。

REST Web 服务使用者转换支持代理服务器。 还可以将 Microsoft SharePoint 应用程序与 REST Web 服务使用者转换连接。

示例

联机存储定义产品数据库的资源。 数据库通过部件号标识每个产品。

Web 服务客户端通过 REST Web 服务访问产品详细信息。 Web 服务使用以下 URL：

`http://www.HypoStores.com/products/ProductDetails`

需要检索有关特定产品的详细信息（如说明和单价），并将详细信息传递给映射中的转换下游。创建 REST Web 服务使用者转换以检索有关产品的详细信息，并将这些信息传递给其他转换。

下表显示了可以配置的转换详细信息：

转换详细信息	值
HTTP 方法	Get
基本 URL	`http://www.HypoStores.com/products/ProductDetails`
输入参数端口	Part_No
输出端口	说明，Unit_Price
方法输出	<响应消息的结构。>

方法输出包括输出映射，可定义响应消息中的元素如何映射到输出端口。

当数据集成服务向 Web 服务发送请求时，其将参数端口中的值附加到基础 URL。 例如，为检索有关部件 0716 的详细信息，数据集成服务使用以下 URL：

`http://www.HypoStores.com/products/ProductDetails?Part_No=0716`

当数据集成服务接收响应时，其将响应消息中的产品说明和单价转换为输出端口的数据。

也可以将 Part_No 作为参数传递，并在运行映射时置换值中游。

REST Web 服务使用者转换流程

REST Web 服务使用者转换根据输入端口和方法输出中的数据创建请求消息。 其基于方法输出将响应消息中的元素转换为输出端口的数据。

REST Web 服务使用者转换的输入端口包含来自映射中的上游转换的关系数据。 数据集成服务使用方法输入将来自输入端口的数据转换为请求消息中的元素。

为连接 Web 服务，数据集成服务读取您在转换属性或 HTTP 连接中配置的基础 URL。 其标识要获取、发布、放置或删除的资源，方法是将来自 URL 端口或参数端口值附加到基础 URL。

当数据集成服务接收响应时，其将响应消息中的数据传递给转换的输出端口。 数据集成服务根据方法输出的配置方式传递数据。 输出端口包含关系数据。 数据集成服务将输出端口中的数据发送到映射中的下游转换或者发送到目标中。

REST Web 服务使用者转换配置

创建 REST Web 服务使用者转换时，选择 HTTP 方法并定义方法输入和输出。 如果选择 Get 方法，则不定义方法输入。

HTTP 请求消息中的输入元素映射到输入端口。 HTTP 响应消息中的输出元素映射到输出端口。 Developer 工具为第一级元素创建端口。

配置转换时，完成以下任务：

1. 选择 HTTP 方法。
2. 配置用以表示请求消息和响应消息的表头和正文中的元素的端口。
3. 配置输入映射。
4. 配置输出映射。
5. 配置高级属性，如 Web 服务的连接和 URL。

如果 REST Web 服务需要进行身份验证，应创建 HTTP 连接对象。

消息配置

数据集成服务生成请求消息，并基于方法输入和输出以及您在 REST Web 服务使用者转换中配置的端口来解释响应消息。

输入端口代表请求消息的不同部分。 可以添加标识要检索或更改的资源的输入端口。 还可以添加代表请求消息中的 HTTP 表头、cookie 信息和元素的输入端口。

输出端口代表响应消息中要发送到下游转换或发送到映射中的目标的元素。 可以代表响应消息中的 HTTP 表头、cookie 信息、响应代码和元素的输出端口。

资源标识

为标识 HTTP 请求中的资源，数据集成服务会将特定输入端口中的值附加到基础 URL。 在 HTTP 连接或转换属性中定义基础 URL。 使用 URL 或参数端口标识特殊资源。

如果 Web 服务通过唯一的字符串标识资源，则使用 URL 端口。

例如，HypoStores REST Web 服务通过以下 URL 使用部件号标识部件：

```
http://www.HypoStores.com/products/ProductDetails/<Part_No>
```

要标识部件，应定义以下转换详细信息：

1. 将基础 URL 设置为以下 URL：

   ```
   http://www.HypoStores.com/products/ProductDetails
   ```

2. 定义 URL 端口，并通过 URL 端口将部件号传递给转换。

如果映射将部件号 500 传递给 URL 端口，则数据集成服务将在请求消息中使用以下 URL：

```
http://www.HypoStores.com/products/ProductDetails/500
```

如果 Web 服务通过参数标识资源的位置，则使用参数端口。

例如，您想通过“Part_No”参数将部件号传递给 HypoStores REST Web 服务。

要标识部件，应定义以下转换详细信息：

1. 将基础 URL 设置为以下 URL：

   ```
   http://www.HypoStores.com/products/ProductDetails
   ```

2. 创建参数名为“Part_No”的参数端口，并通过该参数端口将部件号传递给转换。

如果映射将部件号 600 传递给参数端口，则数据集成服务在请求消息中使用以下 URL：

```
http://www.HypoStores.com/products/ProductDetails?Part_No=600
```

创建多个参数端口可定义多个参数。 数据集成服务使用 & 符号分隔各个参数。

例如，您要从 REST Web 服务检索员工详细信息，并通过“First_Name”和“Last_Name”参数传递员工的名字和姓氏。 创建参数名为“First_Name”和“Last_Name”的参数端口。 如果映射将名字“John Smith”传递给转换，数据集成服务将在请求消息中使用类似以下内容的 URL：

```
http://www.HypoStores.com/employees/EmpDetails?First_Name=John&Last_Name=Smith
```

如果不指定 URL 或参数端口，数据集成服务将使用转换属性的基础 URL 或 HTTP 连接标识资源。 HTTP 连接中的基础 URL 将替代转换中的基础 URL。

HTTP 方法

创建 REST Web 服务使用者转换时，选择数据集成服务在请求消息中使用的 HTTP 方法。 创建转换后无法更改 HTTP 方法。

将转换配置为使用以下 HTTP 方法之一：

Get

从 Web 服务检索资源或资源集合。例如，可以检索产品表或检索有关一个产品的信息。

Post

向 Web 服务发送数据。使用 Post 方法创建资源或资源集合。例如，可以添加新存储事务的详细信息。

放置

替换资源或资源集合。 如果数据不存在，Put 方法将发布数据。 例如，可以更新客户配送地址。

删除

删除资源或资源集合。 例如，可以删除不再为组织工作的员工的记录。

HTTP Get 方法

数据集成服务使用 HTTP Get 方法从 REST Web 服务检索数据。 使用 Get 方法可以检索资源或资源集合。

如果将 REST Web 服务使用者转换配置为使用 Get 方法，则应配置输入端口、方法输出和输出端口。 无需配置方法输入。

示例

您要检索 HypoStores 产品数据库中部件号 500 的说明和价格。Web 服务使用以下 URL 标识部件：

```
http://www.HypoStores.com/products/ProductDetails?Part_No=<Part_No>
```

输入以下基础 URL：

```
http://www.HypoStores.com/products/ProductDetails
```

下表显示了您可定义的输入端口：

端口类型	参数名	输入值
参数	Part_No	500

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	Part_Desc	…<desc>ACME 圆珠笔， 12-pk，黑色，0.7 mm</desc>…
输出	Price_USD	…<price>9.89</price>…

HTTP Post 方法

数据集成服务使用 HTTP Post 方法将数据发送到 REST Web 服务。 Web 服务决定 Post 方法执行的实际功能。 可以使用 Post 方法创建资源或者资源集合。

配置 REST Web 服务使用者转换以使用 Post 方法时，应配置输入端口、方法输入、方法输出和输出端口。

示例

您希望将新部件 501 发布到 HypoStores 产品数据库中。 Web 服务对部件 501 使用以下 URL：

```
http://www.HypoStores.com/products/ProductDetails/501
```

输入以下基础 URL：

```
http://www.HypoStores.com/products/ProductDetails
```

下表显示了可定义的输入端口：

端口类型	端口名称	输入值
URL	URL_Part_No	501
输入	Part_Desc	ACME 圆珠笔，12-pk，黑色，0.5 mm
输入	Price_USD	9.89

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	响应	<Web 服务返回的响应>

HTTP Put 方法

数据集成服务使用 HTTP Put 方法通过 REST Web 服务更新数据。 使用 Post 方法更新资源或资源集合。 如果数据不存在，数据集成服务将创建资源或资源集合。

配置 REST Web 服务使用者转换以使用 Put 方法时，应配置输入端口、方法输入、方法输出和输出端口。

示例

您要更新 HypoStores 产品数据库中部件号 501 的单价。 Web 服务对部件 501 使用以下 URL：

```
http://www.HypoStores.com/products/ProductDetails/501
```

输入以下基础 URL：

```
http://www.HypoStores.com/products/ProductDetails
```

下表显示了可定义的输入端口：

端口类型	端口名称	输入值
URL	URL_Part_No	501
输入	Price_USD	9.99

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	响应	<Web 服务返回的响应>

HTTP Delete 方法

数据集成服务使用 HTTP Delete 方法通过 REST Web 服务删除数据。 使用 Delete 方法可以删除资源或资源集合。

配置 REST Web 服务使用者转换以使用 Delete 方法时，应配置输入端口、方法输入、方法输出和输出端口。

示例

您希望从 HypoStores 产品数据库中删除部件号 502。 Web 服务使用以下 URL 标识部件：

```
http://www.HypoStores.com/products/ProductDetails?Part_No=<Part_No>
```

输入以下基础 URL：

```
http://www.HypoStores.com/products/ProductDetails
```

下表显示了您可定义的输入端口：

端口类型	参数名	输入值
参数	Part_No	502

下表显示了您可定义的输出端口：

端口类型	端口名称	返回值
输出	响应	<Web 服务返回的响应>

REST Web 服务使用者转换端口

一个 REST Web 服务使用者转换可以包含多个输入端口和多个输出端口。 您可以根据 XML 或 JSON 层次结构的结构成组创建端口。

查看转换端口时，如果不需要查看 XML 或 JSON 层次结构，则显示端口。 显示端口时，可以定义组、定义端口以及将元素从方法输入和方法输出映射到输入端口和输出端口。

一个 REST Web 服务使用者转换可以包含多个输入组和多个输出组。 创建端口后，可以创建组，然后将这些端口添加到组中。基于 XML 或 JSON 中输入或输出层次结构的结构在组层次结构中定义端口。 通过添加键，可以将子组与父组关联。

在层次结构中，除了最下面的组外，其他所有组都必须具有主键。在层次结构中，除了根组以外，其他所有组都必须具有外键。

该转换包含一个名为 RequestInput 的根输入组。必须在这个根输入组中添加主键。主键必须为 String、Bigint 或 Integer 类型。可以将根输入组中的任何端口配置为传递端口。

要将元素映射到端口，可单击**位置**列中的字段，然后展开**选择位置**对话框中的层次结构。 接着，从层次结构中选择一个元素。

输入端口

输入端口代表您要传递给 Web 服务的上游转换或源中的数据。 可以配置多个输入端口。 每个输入端口都映射到请求消息中的一个元素。

要添加输入端口，请选择输入组，单击**新建**按钮旁边的箭头，然后选择**字段**。

输出端口

输出端口代表响应消息中要传递给下游转换或传递给目标的元素。 可以配置多个输出端口。 每个输出端口都映射到响应消息中的一个元素。

要添加输出端口，请选择输出组，单击**新建**按钮旁边的箭头，然后选择**字段**。

传递端口

这些传递端口会通过转换传递数据而不进行更改。 可以将根输入组中的任何端口配置为传递端口。

要添加传递端口，可在根输入组中添加一个端口。 然后右键单击该端口并选择**映射**。

参数端口

如果资源的 URL 使用参数，则可以使用参数端口标识该资源。在根输入组中添加参数端口。

参数端口具有端口名称和参数名称。如果参数名称包含不允许在端口名称中使用的字符，请输入与端口名称不同的参数名称。例如，您要将参数“Cust-ID”传递到 Web 服务，但数据集成服务不允许在端口名称中使用短线字符 (-)。输入“Cust-ID”作为参数名称，但输入“CustID”作为端口名称。

数据集成服务将每个参数端口的参数名称和值以“名称=值”对的形式附加到基础 URL。可以配置多个参数端口。数据集成服务使用 & 符号分隔请求中的多个参数。

例如：

```
http://www.HypoStores.com/customers/CustDetails?Last_Name=Jones&First_Name=Mary
```

如果在转换中定义参数端口和 URL 端口，数据集成服务会将 URL 端口值附加到基础 URL，后接参数名称和参数值。

要添加参数端口，可右键单击根输入组，然后选择**新建** > **参数端口**。输入参数名称和端口名称。

URL 端口

URL 端口允许您通过静态 URL 标识资源。 数据集成服务将 URL 端口的值附加到基本 URL，以标识资源。

例如：

```
http://www.HypoStores.com/products/ProductDetails/<URL_port_value>
```

将 URL 端口添加到根输入组。

可以配置多个 URL 端口。 数据集成服务使用斜杠字符 (/) 分隔每个 URL 端口中的值。 如果您在转换中定义了 URL 端口和参数端口，数据集成服务会将 URL 端口值附加到基本 URL，后跟参数名称和值。

要添加 URL 端口，请右键单击根输入组，然后选择**新建** > **URL 端口**。

HTTP 表头端口

HTTP 表头端口代表请求消息中的 HTTP 表头。可以配置多个 HTTP 表头端口。

要将表头信息传递给请求中的 Web 服务，可将端口添加到根输入组中。 可以为根输入组配置一个 HTTP 表头端口。 如果将 HTTP 表头添加到根输入组，可以将其配置为传递端口。

HTTP 表头端口具有端口名称和 HTTP 表头名称。 如果 HTTP 表头名称包含不允许在端口名称中使用的字符，请输入与端口名称不同的 HTTP 表头名称。 例如，您要将表头名称“Content-Type”传递到 Web 服务，但数据集成服务不允许在端口名称中使用短线字符 (-)。 输入“Content-Type”作为 HTTP 表头名称，但输入“ContentType”作为端口名称。

要添加 HTTP 表头端口，可右键单击根输入组，然后选择**新建** > **HTTP 表头**。 输入表头名称和端口名称。

Cookie 端口

可以配置 REST Web 服务使用者转换以使用 Cookie 身份验证。 远程 Web 服务器会根据 cookie 跟踪 Web 服务使用者用户。 某个映射多次调用一个 Web 服务时，可以提高性能。

要将 cookie 信息传递给请求中的 Web 服务，可将端口添加到根输入组中。 可以为根输入组配置一个 cookie 端口。 如果将 cookie 端口添加到根输入组，可以将其配置为传递端口。

要从响应提取 cookie 信息，可在输出组中添加 cookie。 可以为每个输出组配置一个 cookie 端口。

将 cookie 端口映射到 Web 服务请求消息时，Web 服务提供程序将在响应消息中返回 cookie 值。可以将该 cookie 值传递到映射中的其他转换下游，也可以将该 cookie 值保存在文件中。 如果将 cookie 值保存在文件中，可以将该 cookie 配置为 REST Web 服务使用者转换的输入。

要添加 cookie 端口，可右键单击根输入组，然后选择**新建** > **其他端口**。 接着，选择 **Cookie**，并单击**确定**。

输出 XML 端口

输出 XML 端口代表 Web 服务的响应。 输出 XML 端口是字符串端口。

将输出 XML 端口添加到输出组。 可以为每个输出组配置一个输出 XML 端口。

右键单击根输入组，然后选择**新建** > **其他端口**。 接着，选择**输出 XML**，并单击**确定**。

响应代码端口

响应代码端口代表 Web 服务的 HTTP 响应代码。 响应代码端口是 Integer 端口。

将响应代码端口添加到输出组。 可以为每个输出组配置一个响应代码端口。

要添加响应代码端口，可右键单击根输入组，然后选择**新建** > **其他端口**。 接着，选择**响应代码**，并单击**确定**。

REST Web 服务使用者转换输入映射

查看转换端口时，显示输入映射以查看方法输入层次结构。 显示输入映射时，可以定义输入组，定义输入端口，并将输入端口映射到方法输入元素。

输入映射包含以下区域：

端口

可在**端口**区域中创建转换输入组和输入端口。

方法输入

方法输入区域显示 REST Web 服务使用者转换发送到 Web 服务的请求消息中的元素。 如果使用架构对象创建转换，该架构对象将定义方法输入层次结构。

创建输入端口后，将**端口**区域的输入端口映射到**方法输入**区域中的元素。 将输入端口映射到方法输入中的元素时，端口的位置显示在**方法输入**区域的位置列中。

如果选择映射输入第一级层次结构，Developer tool 会将方法输入的第一级中的元素映射到输入端口。 Developer tool 还会创建端口以执行映射。如果第一级层次结构中包含的多次出现父元素含有一个或多个多次出现子元素，则 Developer tool 不映射第一级层次结构。

可以选择查看将输入端口连接到方法输入中的元素的行。

将输入端口映射到元素的规则和指导

将输入端口映射到方法输入层次结构中的元素时，请查看以下规则：

- 可以将一个输入端口映射到层次结构中的一个元素。 可以将同一端口映射到层次结构中任意数量的键。
- 输入端口和元素的数据类型必须兼容。
- 可以将一个输入组中的端口映射到方法输入中的多个层次结构级别。
- 必须将输入端口映射到方法输入中的键。 映射到键的任何端口的数据类型都必须为字符串、整型或长整型。将数据映射到方法输入中位于请求消息所含层次结构级别之上的所有级别中的键。 对于映射级别及之上的所有级别，需要包括外键。

  注意: 如果只映射方法输入层次结构的最低级别，则无需将输入端口映射到键。
- 必须将 RequestInput 根元素映射到方法输入定义的 Rest_Consumer_input 组的子元素。
- 可以将多个 String、Bigint 或 Integer 输入端口映射到**方法输入**区域中的键以创建复合键。 为复合键单击**位置**字段时，可以重新对输入端口进行排序，或者删除其中一个端口。
- 如果 Web 服务生成 JSON 文档，应确保 xmlRoot 是响应层次结构中的第一个节点。如果 xmlRoot 不是具有 JSON 响应的 Web 服务的第一个节点，则可能会显示空值。

将输入端口映射到方法输入为 JSON 的元素时，请查看以下规则：

- 确保传递到输入端口的数据不包含以多个零开头的数字。如果数据以多个零开头，则这些零在生成的 JSON 请求的相应值中被截断，值的数据类型会从字符串更改为数值数据类型。

将输入端口映射到方法输入

显示转换输入映射时，可以定义输入组，定义输入端口，以及将输入端口映射到方法输入元素。

1. 打开 REST Web 服务使用者转换。
2. 在**端口**视图中，显示输入映射。
3. 为根输入组定义主键。

4. 要将输入组或端口添加到**端口**区域，请使用以下方法之一：

方法	说明
拖动元素。	将一个组或一个子元素从**方法输入**区域拖动到**端口**区域的一个空列中。 如果将组拖动到**端口**区域，Developer tool 会添加一个不包含端口的组。
手动添加组或端口。	要添加组，请单击**新建**按钮旁边的箭头，然后单击**组**。 要添加端口，请单击**新建**按钮旁边的箭头，然后单击**字段**。
从其他转换中拖动端口。	在编辑器中，将端口从其他转换拖动到 REST Web 服务使用者转换。
复制端口。	从其他转换中选择端口，并将其复制到**端口**区域。要复制端口，可以使用键盘快捷键，或者使用 Developer tool 中的**复制**和**粘贴**按钮。
选择**映射第一级层次结构**。	Developer tool 会将方法输入第一级中的元素映射到输入端口和组。 Developer tool 还会创建输入端口和组以执行映射。

5. 如果手动创建端口或从其他转换复制端口，请单击**方法输入**区域中的**位置**列，然后从列表中选择端口。
6. 要将输入端口映射为复合键，请使用以下方法之一：

方法	说明
拖动输入端口。	选择两个或多个输入端口，然后将它们拖动到方法输入层次结构的键中。
从**选择位置**对话框中选择输入端口。	在方法输入层次结构中单击键的**位置**列，然后选择输入端口。

7. 要清除元素的位置，请使用以下方法之一：

方法	说明
单击**清除**。	在**方法输入**区域中选择一个或多个元素，然后单击**清除**。
删除将端口连接到元素的行。	在方法输入中选择将输入端口连接到元素的一个或多个行，然后按**删除**。

REST Web 服务使用者转换输出映射

查看转换端口时，系统将显示输出映射，从中可以查看方法输出层次结构。 显示输出映射时，可以定义输出组，定义输出端口，并将方法输出元素映射到输出端口。

输出映射包含以下区域：

方法输出

方法输出区域显示了 Web 服务返回给 REST Web 服务使用者转换的响应消息中的元素。 如果使用架构对象创建转换，该架构对象将定义方法输出层次结构。

端口

可在**端口**区域中创建转换输出组和端口。

创建输出端口后，将**方法输出**区域中的元素映射到**端口**区域中的端口。 将方法输出中的元素映射到输出端口时，元素的位置将显示在**端口**区域的**位置**列中。

选择映射输出层次结构的第一级时，Developer 工具会将方法输出的第一级中的元素映射到输出端口。 开发程序工具还会创建端口以执行映射。如果层次结构的第一级中包含的多次出现父元素含有一个或多个多次出现子元素，则 Developer 工具不映射第一级层次结构。

可以选择在层次结构中显示输出端口。每个子组都显示在父组下方。还可以选择查看将方法输出中的元素连接到输出端口的行。

如果从存储库中删除了关联的架构对象，则 Developer 工具会将方法元素的位置保留在输出映射中。 显示输出映射时，**端口**区域仍将在输出端口的**位置**列中显示方法元素的位置。 如果将其他架构与转换关联，则 Developer 工具会检查是否每个位置都有效。 如果方法元素位置不再有效，则 Developer 工具将在输出映射的**端口**区域中清除该位置。

将元素映射到输出端口的规则和准则

将方法输出层次结构中的元素映射到输出端口时，请查看以下规则：

- 方法输出元素和输出端口的数据类型必须兼容。
- 不能将一个元素映射到一个组中的多个输出端口。
- 除非端口是传递端口，否则每个输出端口都必须具有有效位置。
- 如果要将多次出现的子元素拖动到空的输出端口，则必须将该组与其他输出组关联。 选择组后，开发程序工具将创建用于关联组的键。
- 将多次出现的元素拖动到包含父元素的组时，可以配置要包含的子元素的出现次数。也可以将父组替换为转换输出中多次出现的子组。
- 如果 Web 服务生成 JSON 文档，应确保 xmlRoot 是响应层次结构中的第一个节点。如果 xmlRoot 不是具有 JSON 响应的 Web 服务的第一个节点，则输出端口中可能会显示空值。

自定义视图选项

可以更改方法输出层次结构，以便在**方法输出**区域中显示 cookie 端口、传递端口和键。 还可以显示用于定义元素排序方式的分组结构。

在**方法输出**区域中单击**自定义视图**。 启用以下任意选项：

“序列”、“选项”和“全部”

显示一行，指示元素定义是“序列”、“选项”还是“全部”。

序列组中的元素顺序必须是在层次结构中指定的顺序。

选项组中至少有一个元素必须显示在响应消息中。

“全部”组中的元素必须全部包含在响应消息中。

键

在**方法输出**区域中查看键。 **方法输出**区域包含每个组的键。 可以在**端口**区域中将键添加到输出端口。

传递端口

方法输出区域显示传递端口。 传递端口是指通过转换传递数据而不更改数据的端口。可以将方法输出中的传递端口映射到任何 REST Web 服务使用者转换输出组。 传递端口会一次性接收数据，因此，该端口处于响应消息中的根级别。

将方法输出映射到输出端口

显示转换输出映射时，可以定义输出组，定义输出端口，并将方法输出元素映射到输出端口。

1. 打开 REST Web 服务使用者转换。
2. 在**端口**视图中，显示输出映射。
3. 要将输出组或端口添加到**端口**区域，请使用以下方法之一：

方法	说明
拖动元素。	将**方法输出**区域中的一个组或一个子元素拖动到**端口**区域的一个空列中。 如果将组拖动到**端口**区域，Developer tool 会添加一个不包含端口的组。
手动添加组或端口。	要添加组，请单击**新建**按钮旁边的箭头，然后单击**组**。要添加端口，请单击**新建**按钮旁边的箭头，然后单击**字段**。
从其他转换中拖动端口。	在编辑器中，将端口从其他转换拖动到 REST Web 服务使用者转换。
复制端口。	从其他转换中选择端口，并将其复制到**端口**区域。要复制端口，可以使用键盘快捷键，或者使用 Developer tool 中的**复制**和**粘贴**按钮。

4. 如果手动创建端口或从其他转换复制端口，请单击**端口**区域中的**位置**列，然后从列表中选择元素。
5. 要清除端口的位置，请使用以下方法之一：

方法	说明
单击**清除**。	在**端口**区域中选择一个或多个端口，然后单击**清除**。
删除将元素连接到端口的行。	在方法输出中选择将元素连接到输出端口的一个或多个行，然后按**删除**。

REST Web 服务使用者转换高级属性

配置有助于确定数据集成服务如何为 REST Web 服务使用者转换处理数据的属性。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

连接

标识连接到 Web 服务的 HTTP 连接对象。 在 Developer tool 中创建和编辑 HTTP 连接。配置 HTTP 连接时，应配置基础 URL、Web 服务所需的安全类型和连接超时时限。

REST Web 服务使用者转换使用 URL 连接到 Web 服务。 可以在转换属性或 HTTP 连接中定义 URL。

在以下情况下配置 HTTP 连接：

- 不使用 URL 输入端口。
- Web 服务需要进行 HTTP 身份验证或使用 SSL 证书。

- 您希望更改默认连接超时时限。

XML 架构验证

在运行时验证响应消息。**选择 XML 无效错误**或**不验证**。

已排序输入

允许数据集成服务在未处理所有输入数据的情况下生成输出。当输入数据按 XML 输入层次结构中的键排序时，请启用已排序输入。

URL

REST Web 服务的基础 URL。 HTTP 连接中的基础 URL 将替代该值。

格式

Web 服务响应的格式。 选择 **XML** 或 **JSON**，具体视 Web 服务响应而定。

REST Web 服务使用者转换创建

可以创建可重用或不可重用的 REST Web 服务使用者转换。 可重用转换可存在于多个映射中。不可重用转换存在于单个映射内。

创建 REST Web 服务使用者转换时，可以手动定义元素和 XML 层次结构，或者可以从架构对象导入元素和层次结构。 架构对象可以是 XML 文件或文本文件。

创建 REST Web 服务使用者转换

创建 REST Web 服务使用者转换时，选择方法并基于所选方法定义方法输入和方法输出。

1. 要创建 REST Web 服务使用者转换，请使用以下方法之一：

方法	说明
可重用	在“对象浏览器”视图中选择项目或文件夹。单击**文件** > **新建** > **转换**。选择 REST Web 服务使用者转换，然后单击**下一步**。
不可重用	在映射或 Mapplet 中，将 REST Web 服务使用者转换从转换选项板拖动到映射或 Mapplet 编辑器中。

2. 输入转换名称，并选择位置和 HTTP 方法。
3. 单击**下一步**。
4. 要定义方法输入，请使用以下方法之一：

方法	说明
创建为空	手动定义 XML 元素和层次结构。
从架构对象中的元素创建	从架构对象导入 XML 元素和层次结构。

方法输入定义区域显示转换输入组和输入端口。 **输入映射**区域显示请求消息层次结构。

5. 定义输入组和输入端口，并将输入端口映射到输入元素。
6. 单击**下一步**。

7. 要定义方法输出，请选择**创建为空**或**从架构对象中的元素创建**。

   方法输出定义区域显示转换输入组和输入端口。 **输出映射**区域显示请求消息层次结构。
8. 定义输出组和输出端口，并将元素映射到输出端口。
9. 单击**完成**。

解析包含数组的 JSON 响应消息

当元素是复杂类型的子元素且该元素的最大出现次数为无限大时，架构无效。JSON 解析器会限制提取元素的多个实例。

若在架构中使用复杂类型的 choice 顺序指示器，属于复杂类型的子元素的最大出现次数必须为 0 或 1。如果将最大出现次数更改为 1 以使架构有效，则一次只能提取元素的一个实例。

在架构中，可以在复杂类型的 choice 顺序指示器中使用无限大的最大出现次数。

JSON 响应消息示例

您具有以下架构，其中复杂类型元素 xmlRoot 包含名为 Likes 且最大出现次数为无限大的元素：

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://
www.w3.org/2001/XMLSchema">
   <xs:element name="xmlRoot">
     <xs:complexType>
       <xs:all>
         <xs:element type="xs:byte" name="Age"/>
         <xs:element type="xs:string" name="FirstName"/>
         <xs:element type="xs:string" name="Likes" maxOccurs="unbounded" minOccurs="0"/>
         <xs:element type="xs:string" name="FamilyName"/>
       </xs:all>
     </xs:complexType>
   </xs:element>
 </xs:schema>
```

您可以按以下格式更改 JSON 响应：

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xs="http://
www.w3.org/2001/XMLSchema">
   <xs:element name="xmlRoot">
     <xs:complexType>
       <xs:choice maxOccurs="unbounded">
         <xs:element type="xs:byte" name="Age"/>
         <xs:element type="xs:string" name="FirstName"/>
         <xs:element type="xs:string" name="Likes" />
         <xs:element type="xs:string" name="FamilyName"/>
       </xs:choice>
     </xs:complexType>
   </xs:element>
 </xs:schema>
```

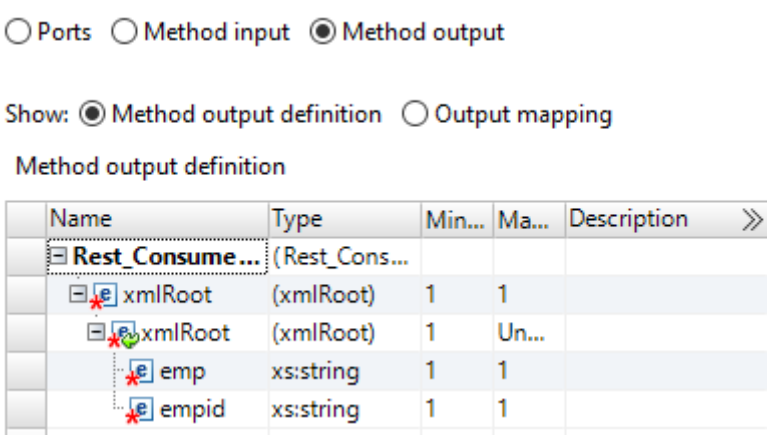
`<xs:choice maxOccurs="unbounded">` 允许内容以任何顺序重复一次或更多次。

响应消息中的未命名数组

REST Web 服务使用者转换仅支持在响应消息中使用未命名数组，不支持在请求消息中使用未命名数组。要解析在“方法输出定义”中定义的未命名数组架构，complexType 或简单类型数组元素的父元素必须名为 `xmlRooot`。

在 Rest Web 服务使用者转换中，必须将 xmlRoot 定义为最大出现次数设置为无限的 xmlRoot 元素的子元素，并将未命名数组中的元素设置为 xmlRoot 元素的子元素。

下图显示了为未命名数组定义的方法输出：



第 40 章

路由器转换

本章包括以下主题：

路由器转换概览

路由器转换是一种主动转换，用于根据一个或多个条件将数据路由到多个输出组。在映射中，将输出组路由到不同的转换或不同的目标。

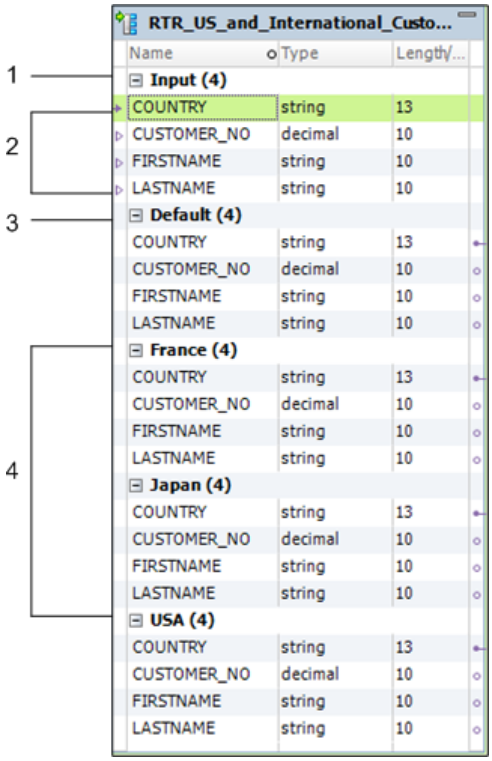
路由器转换与筛选器转换类似，因为这两个转换都使用条件来测试数据。筛选器转换会根据一个条件来测试数据，并删除不满足该条件的数据行。路由器转换会根据一个或多个条件来测试数据，并可将不满足其中任意条件的数据行路由到默认的输出组。

如果需要根据多个条件测试相同的输入数据，请在映射中使用一个路由器转换（而不是创建多个筛选器转换来执行同一任务）。路由器转换效率更高。例如，要根据三个条件测试数据，可以使用一个路由器转换，而不是三个筛选器转换。在映射中使用路由器转换时，数据集成服务将一次性处理所有传入数据。在映射中使用多个筛选器转换时，数据集成服务将处理每个转换的传入数据。

路由器转换由输入和输出组、输入和输出端口、组筛选条件以及高级属性（在 Developer tool 中配置）组成。

当 Spark 引擎运行具有“路由器转换”的映射时，Spark 引擎会处理一次上游映射管道并将数据暂存在 HDFS 上，每个下游分支都可以使用此数据。

下图显示了一个路由器转换示例及其组件：



1. 输入组
2. 输入端口
3. Default 输出组
4. 用户定义的输出组

动态映射中的路由器转换

您可以在动态映射中使用路由器转换。 您可以在转换中配置动态端口，并在组筛选条件中引用生成的端口。

在组筛选条件中使用动态端口时，该动态端口可以包含多个生成的端口。 组筛选条件扩展为包括每个生成的端口。 对于表达式而言，每个生成的端口必须是有效类型。

您可以对组筛选条件进行参数化。 使用表达式类型参数来指定筛选器。

使用组

路由器转换具有以下类型的组：

- 输入
- 输出

输入组

路由器转换包含一个输入组。该输入组包含添加到该转换中的所有输入端口。

输出组

等级转换包括以下类型的输出组：

用户定义的组

可创建用户定义的组以便基于传入数据测试条件。 用户定义的组由输出端口和组筛选条件组成。 可以在**组**选项卡上使用 Developer 工具创建和编辑用户定义的组。 为要指定的每个条件创建一个用户定义的组。

数据集成服务将使用该条件来计算传入数据的每一行。 它将在处理默认组之前测试每个用户定义组的条件。 数据集成服务将根据所连接的输出组的顺序确定每个条件的计算顺序。 数据集成服务将处理连接到转换或映射目标的用户定义组。

如果某行满足多个组筛选条件，则数据集成服务将多次传递此行。

默认组

您创建一个用户定义的组后，Developer 工具将创建默认组。 Developer 工具不允许编辑或删除默认组。 该组没有与其相关联的组筛选条件。 如果所有组条件的计算结果均为 FALSE，则数据集成服务会将相应行传递至默认组。 如果希望数据集成服务丢弃默认组中的所有行，请不要将其连接到转换或映射目标。

从列表中删除最后一个用户定义的组后，Developer 工具将删除默认组。

Developer 工具将复制输入组的输入端口中的属性信息，以便为每个输出组创建一组输出端口。 无法更改或删除输出端口或其属性。

使用组筛选条件

可以根据一个或多个组筛选条件测试数据。可以使用表达式编辑器在**组**选项卡上创建组筛选条件。

您可以输入任意表达式以返回单一值。还可以为条件指定一个常量。对于传递给转换的每一行，组筛选条件都会根据该行是否满足指定条件来返回 TRUE 或 FALSE。零 (0) 相当于 FALSE。任何非零值均等效于 TRUE。您可以使用单个数值型端口作为筛选条件。数据集成服务会将计算结果为 TRUE 的数据行传递给与用户定义的每个组相关联的每个转换或目标。

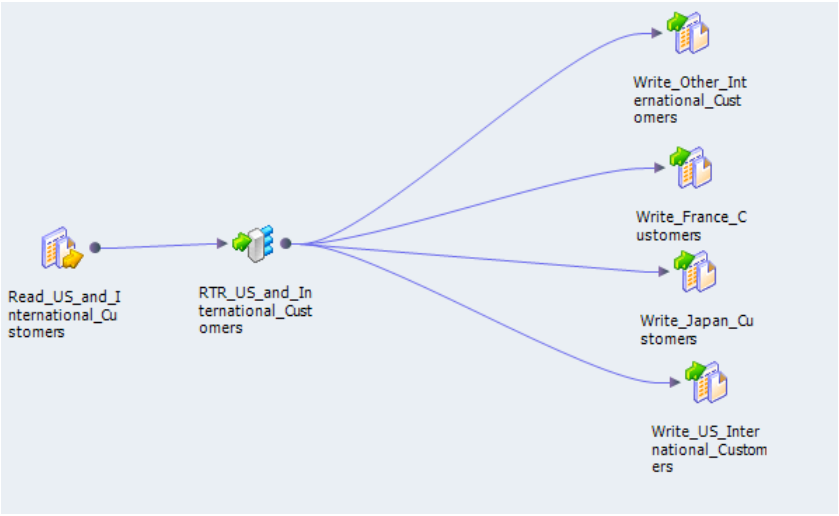
注意: 您不能使用单个动态端口来返回布尔值。

例如，您的客户分布在九个国家/地区，您希望对其中三个国家/地区的数据执行不同的计算。您可以在映射中使用一个路由器转换将此数据筛选到三个不同的表达式转换中。

可以使用参数作为组筛选条件中的元素。还可以使用系统参数或用户定义的参数。可以在表达式编辑器中创建参数并将其添加到表达式。

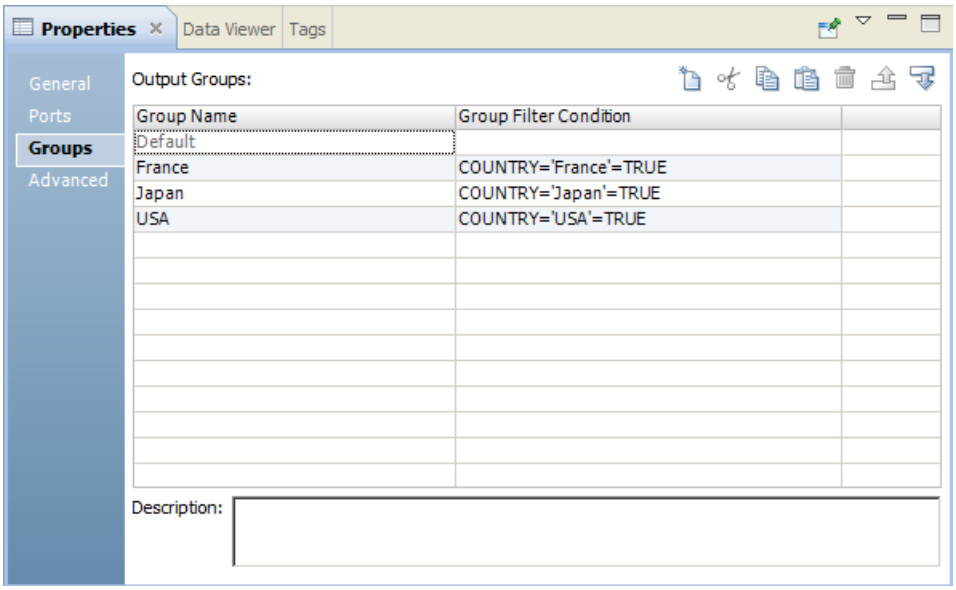
默认组不具有组筛选条件。但是，您可以创建一个表达式转换以根据另外六个国家/地区的数据来执行计算。

下图显示了一个使用路由器转换的映射，该转换将根据多个条件来筛选数据：



要根据三个不同国家/地区的数据执行多个计算，请在**组**选项卡中创建三个由用户定义的组，并指定三个组筛选条件。

下图显示了用于筛选客户数据的组筛选条件：



下表显示了用于筛选客户数据的组筛选条件：

组名称	组筛选条件
法国	customer_name=‘France’=TRUE
日本	customer_name=‘Japan’=TRUE
USA	customer_name=‘USA’=TRUE

在该映射中，数据集成服务会将计算结果为 TRUE 的数据行传递给与用户定义的每个组（Japan、France 和 USA）相关联的每个转换或目标。如果所有条件的计算结果均为 FALSE，则数据集成服务会将该行传递给默认

组， 然后，数据集成服务会将其他六个国家/地区的数据传递给与默认组相关联的转换或目标。如果希望数据集成服务丢弃默认组中的所有行，请不要将其连接到转换或映射目标。

路由器转换会将数据传递给满足条件的每个组。如果数据满足三个输出组的条件，则路由器转换会将此数据传递给这三个输出组。

例如，在路由器转换中配置了以下组条件：

组名称	组筛选条件
输出组 1	employee_salary > 1000
输出组 2	employee_salary > 2000

如果路由器转换处理的输入行数据的 employee_salary=3000，则会将此数据路由到输出组 1 和 2。

组筛选条件中的动态端口

您可以在组筛选条件中使用动态端口。 数据集成服务将筛选条件应用于动态端口中的每个生成的端口。

例如，动态端口 MyDynamicPort 包含三个小数端口：

```
Salary
Bonus
Stock
```

您可以配置以下组筛选条件：

```
MyDynamicPort > 100
```

组筛选条件扩展到以下表达式：

```
Salary > 100 AND Bonus > 100 AND Stock > 100
```

对于表达式而言，每个生成的端口必须是有效类型。

参数化组筛选器

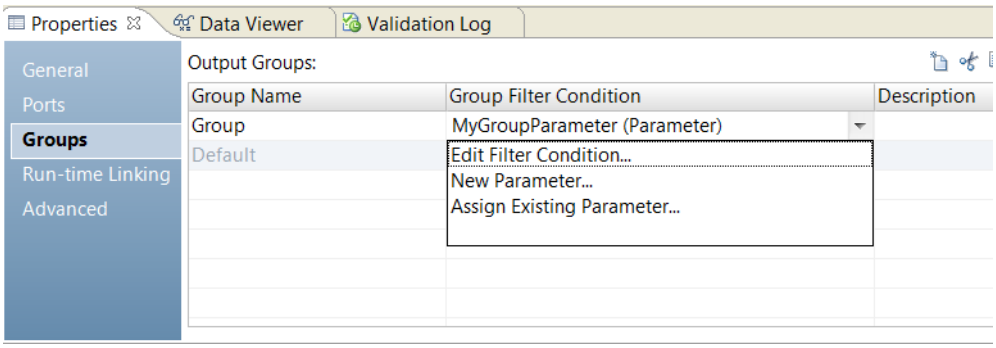
可以使用表达式参数定义组筛选器。 表达式参数是包含完整表达式的参数。

例如，表达式参数可能包含表达式

```
Employee_Salary > 1000.
```

要参数化组筛选器，请在**组**选项卡的**组筛选条件**字段中选择**新建参数**。 定义该参数并在表达式编辑器中输入表达式。 表达式参数不能包含其他参数。

下图显示了转换属性中的**组**选项卡：



当您使用表达式参数时，Developer tool 无法验证表达式。 如果表达式在运行时无效，映射可能会失败。

添加组

添加组时，Developer 工具会将属性信息从输入端口复制到输出端口。

1. 单击**组**选项卡。
2. 单击**新建**按钮。
3. 在**组名称**部分输入组的名称。
4. 单击**组筛选条件**字段打开**表达式编辑器**。
5. 输入组筛选条件。
6. 单击**验证**检查条件的语法。
7. 单击**确定**。

使用端口

路由器转换具有输入端口和输出端口。 输入端口位于输入组中，输出端口位于输出组中。

您可以通过以下方式创建输入端口：从其他转换复制输入端口或者在**端口**选项卡中手动创建输入端口。

Developer 工具会通过从输入端口复制以下属性来创建输出端口：

- 端口名称
- 数据类型
- 精度
- 小数位数
- 默认值

对输入端口进行更改后，Developer 工具会更新输出端口以反映这些更改。 您不能编辑或删除输出端口。

Developer 工具会根据输入端口名称创建输出端口名称。 对于每个输入端口，Developer 工具会在每个输出组中创建相应的输出端口。

在映射中连接路由器转换

将转换连接到映射中的路由器转换时，请考虑以下规则：

- 可以将一个组连接到一个转换或目标。
- 可以将组中的一个输出端口连接到多个转换或目标。
- 可以将一个组中的多个输出端口连接到多个转换或目标。
- 无法将多个组连接到一个目标或单个输入组转换。
- 将每个输出组与不同输入组连接时，可以将多个组连接到多个输入组转换，但联接器转换除外。

路由器转换高级属性

配置高级属性以帮助确定数据集成服务如何为路由器转换处理数据。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境下的路由器转换

非本地环境下的路由器转换处理取决于运行此转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射和流映射中无限支持。
- Databricks Spark 引擎。无限支持。

第 41 章

序列生成器转换

本章包括以下主题：

序列生成器转换概览

序列生成器转换是一种可生成数值的被动转换。使用序列生成器转换可创建唯一主键值、替换缺少的主键或循环生成一系列有序数字。

序列生成器包含传递端口和输出端口。将 NEXTVAL 端口连接到其他转换的输入端口。映射运行时集成服务将增加序列。

可以基于新序列或序列数据对象创建序列生成器转换。序列数据对象指创建和维护值序列的对象。

序列生成器端口

序列生成器转换具有传递端口和一个输出端口 NEXTVAL。 不能编辑或删除输出端口。

传递端口

可以向序列生成器转换中添加一个端口作为传递端口。传递端口是接收输入数据并将相同数据返回映射而不进行任何更改的输入和输出端口。

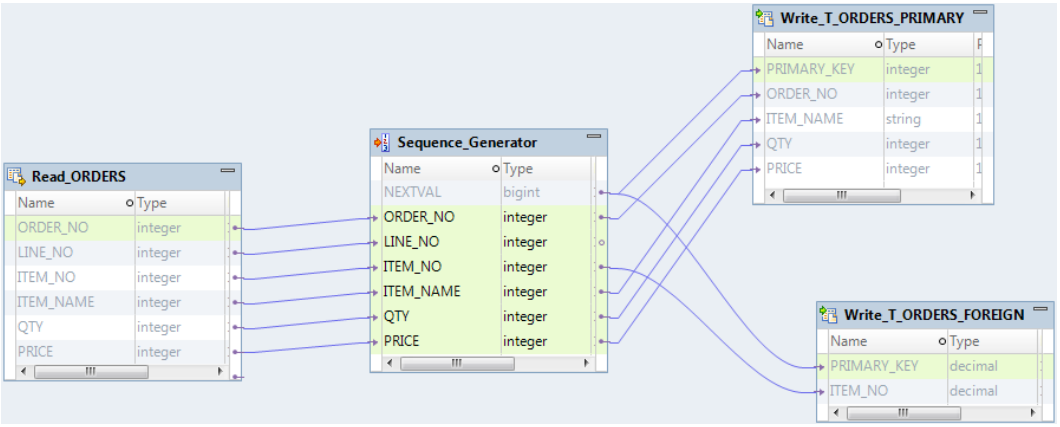
您必须至少将一个输入端口添加到转换中，并在将 NEXTVAL 输出端口链接到目标之前将该端口与上游源或转换连接。要将传递端口添加至转换，请从映射中的上游源或转换中将某个端口拖动到序列生成器转换中。

NEXTVAL 端口

您可以将 NEXTVAL 连接到某个转换以针对该转换中的每行生成唯一值。 将 NEXTVAL 端口连接到下游转换或目标以生成数字序列。 如果将 NEXTVAL 连接到多个转换，集成服务将为每个转换都生成相同的数字序列。

连接 NEXTVAL 端口以基于“起始值”和“增量值”属性生成序列。 如果未将序列生成器配置为在序列范围内循环，则 NEXTVAL 端口将一直生成序列号，直到配置的“结束值”。

下图显示了连接到一个源和两个目标以生成主键值和外键值的序列生成器转换 NEXTVAL 端口的映射：



对序列生成器转换进行如下配置时，集成服务将为 T_ORDERS_PRIMARY 和 T_ORDERS_FOREIGN 目标表生成相同主键值：起始值 = 1，增量值 = 1。

创建键

可以通过将 NEXTVAL 端口连接到目标或下游转换来使用序列生成器转换创建主键值或外键值。可以使用的值必须在 1 到 9,223,372,036,854,775,807 范围之内，最小间隔为 1。此外，还可以创建复合键来标识表中各行。

要创建复合键，可以配置集成服务，使其在一组更少的值之间进行循环。例如，假定有三个商店生成订单号，则可以配置序列生成器转换，使其在值 1 到 3 之间循环，增量为 1。将 ORDER_NO 端口连接到序列生成器转换时，生成的值将创建唯一的复合键。

以下示例显示了复合键和订单号：

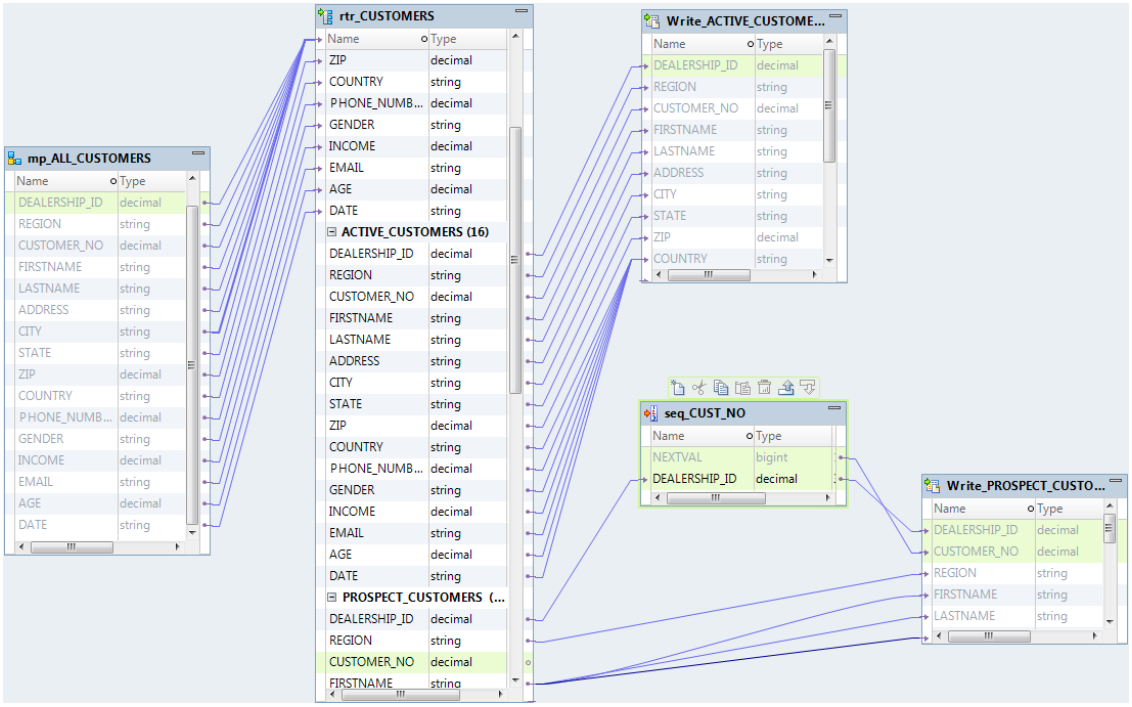
COMPOSITE_KEY	ORDER_NO
1	12345
2	12345
3	12345
1	12346
2	12346
3	12346

替换缺少的值

在使用序列生成器转换替换缺少的键时，可以同时使用路由器转换从已分配值的列中筛选出空值。可以将路由器转换连接到序列生成器转换，并使用 NEXTVAL 生成一个数值序列来填充空值。

例如，要替换 CUSTOMER_NO 列中的空值，可使用包含客户数据的源创建一个映射。可以添加一个路由器转换，以筛选出从具有空值的列中分配了客户编号的客户。可以添加一个序列生成器转换，以生成唯一的 CUSTOMER_NO 值。可以添加要写入数据的客户目标。

下图显示了一个映射，该映射将替换 CUSTOMER_NO 列中的空值：



序列生成器转换属性

配置集成服务用来生成连续值的转换属性。

下表列出了可为序列数据对象和新序列配置的属性：

属性	说明
起始值	在使用“循环”选项时希望集成服务使用的生成序列的起始值。如果选择了“循环”，则集成服务在达到结束值时会回到此值。 默认值为 0。 最大值为 9,223,372,036,854,775,806。
结束值	集成服务所生成的最大值。如果集成服务在会话期间达到此值且未将序列配置为循环，则会话将失败。 最大值为 9,223,372,036,854,775,807。

属性	说明
增量值	NEXTVAL 端口中两个相邻值之间的差值。 默认值为 1。 必须为正整数。 最大值为 2,147,483,647。
循环	如果启用，则集成服务会从起始值重新开始在序列范围中进行循环。 如果禁用，则集成服务会在配置的结束值处停止序列。如果集成服务在达到结束值后仍有要处理的行，则会话会失败并出现溢出错误。

起始值

使用“循环”可生成重复的序列，例如对应于一年中月份的数字 1-12。

1. 输入序列中希望集成服务用于“起始值”的最低值。
2. 输入要用作“结束值”的最高值。
3. 选择“循环”。

循环时，如果集成服务达到为序列配置的结束值，其将返回并再次启动循环（从配置的起始值开始）。

结束值

结束值是您希望集成服务生成的最大值。如果集成服务达到结束值，且序列生成器未配置为在序列中循环，则映射运行将失败并显示一个溢出错误。

可以将结束值设置为 1 到 9,233,372,036,854,775,807 之间的任何整数。如果将 NEXTVAL 端口连接到下游整数端口，应将“结束值”设置为一个不大于最大整数值的值。例如，如果将 NEXTVAL 端口连接到小整数端口，设置结束值时最大值为 32,767。如果 NEXTVAL 超过下游端口的数据类型最大值，映射将失败。

增量值

集成服务根据序列生成器转换中的“当前值”属性和“增量”属性，在 NEXTVAL 端口中生成一个序列。

“当前值”属性是集成服务开始为每个会话创建序列的值。 “增量”属性是集成服务添加到现有值以创建序列中的新值的整数。 默认情况下，“当前值”设置为 1，“增量”设置为 1。

例如，可以创建当前值为 1,000、增量为 10 的序列生成器转换。 如果要通过映射来传递三行，则集成服务将生成以下值集：

```
1000
1010
1020
```

在值范围内循环

可以为序列生成器转换建立一个值范围。如果使用循环选项，序列生成器转换会在达到结束值时重复该范围。

例如，如果您设置一个从 10 开始到 50 结束的序列范围，并设置增量值 10，则序列生成器转换会创建值 10、20、30、40、50。该序列将从 10 重新开始。

序列生成器高级属性

配置集成服务用来生成连续值的高级属性。

下表列出了可为序列数据对象和新序列配置的高级属性：

属性	说明
重置	如果启用，则在映射运行完成时，集成服务会将序列数据对象重置为起始值。如果禁用，则在映射运行完成后，集成服务将对当前值递增并在下次映射运行中使用该值。 该属性会对可重用序列生成器转换和使用可重用序列数据对象的不可重用序列生成器转换禁用。
跟踪级别	集成服务写入映射日志中的转换的相关详细信息的级别。可以选择精简、普通、详细初始化或详细数据。默认值为“普通”。
维持行顺序	保持转换输入数据的行顺序。如果集成服务不应执行任何可能改变行顺序的优化，请选择此选项。默认值为 false。

重置

如果将不可重用序列生成器转换配置为使用重置属性，则集成服务会将原始起始值用于每次映射运行。否则，集成服务将对当前值递增并在下次映射运行时使用该值。

例如，配置一个序列生成器转换以创建从 1 到 1,000 的值，其增量为 1，起始值为 1，并选择“重置”。在第一个映射运行期间，集成服务将生成 1 到 234 的数字。以后每次运行映射时，集成服务都会从起始值 1 开始重新生成数字。

如果没有重置，则集成服务会在第一次运行结束时将当前值更新为 235。下次使用该序列生成器转换时，生成的第一个值为 235。

注意: 对于可重用的序列生成器转换，将禁用“重置”。

保持行顺序

保持转换输入数据的行顺序。如果集成服务不应执行任何可能改变行顺序的优化，请选择此选项。

集成服务执行优化时，可能会失去之前在映射中建立的顺序。您可以在具有已排序的平面文件源、已排序的关系源或排序器转换的映射中设置顺序。当您将转换配置为保持行顺序时，集成服务将在执行映射优化时考虑此配置。如果集成服务可以保持顺序，将为转换执行优化。如果优化会改变行顺序，集成服务将不会为转换执行优化。

序列数据对象

序列数据对象创建和维护数字值序列。序列生成器转换使用序列数据对象为转换生成值。

可以在多个序列生成器转换中使用可重用序列数据对象。如果使用同一个序列数据对象的所有序列生成器转换在相同集成服务中运行，则它们将使用相同的值序列。也可以在不可重用序列生成器转换中使用可重用序列数据对象。可以在不可重用序列生成器转换中使用不可重用序列数据对象。

例如，创建多个在关系表中写入相同主键字段的映射。每个映射使用相同的可重用序列生成器转换，而可重用序列生成器转换使用同一个可重用序列数据对象并在相同的集成服务中运行。每个映射将向主键字段写入唯一值。

创建序列数据对象

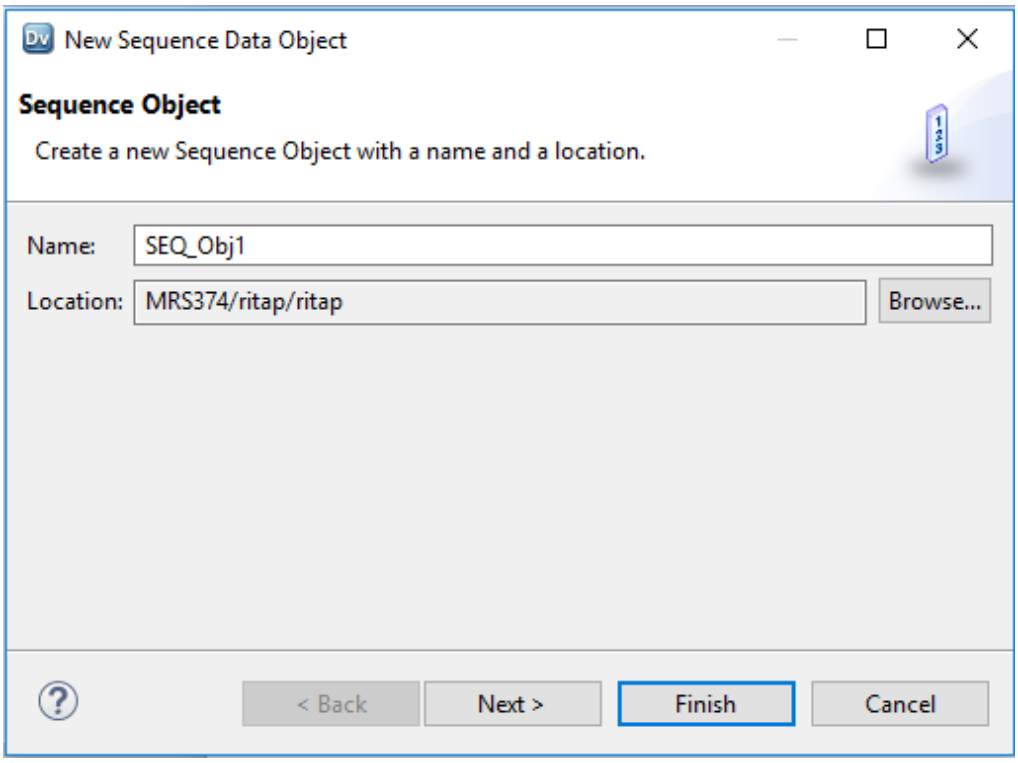
要使用序列数据对象创建序列生成器转换，请创建序列数据对象，配置对象属性，然后在“序列生成器转换”对话框中选择对象。

1. 在映射编辑器中，在映射选项板中向下滚动以找到该序列生成器转换，然后将其拖动到映射中。

   此时将打开**新建转换**向导。

2. 单击**新建序列数据对象**。

   此时将打开**新建数据对象**向导。

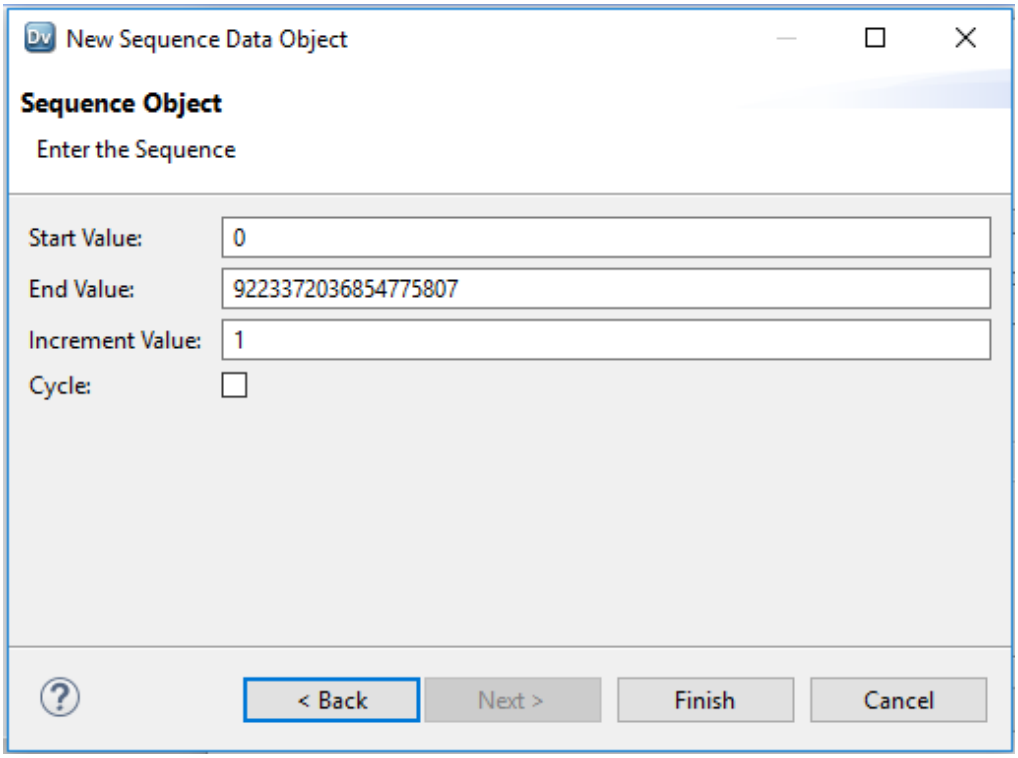


3. 输入序列数据对象的名称。

   序列数据对象的命名约定是 SEQ_<数据对象名称>。

4. 单击**下一步**以配置序列数据对象属性。

如果从对象创建序列生成器转换，转换将使用您为该数据对象输入的属性。下图显示了可以配置的属性：



5. 配置数据对象属性后，可以使用该序列数据对象创建序列生成器转换。创建转换时，请为序列生成器转换命名，然后单击**选择现有序列对象**。导航到该数据对象，然后单击**确定**。

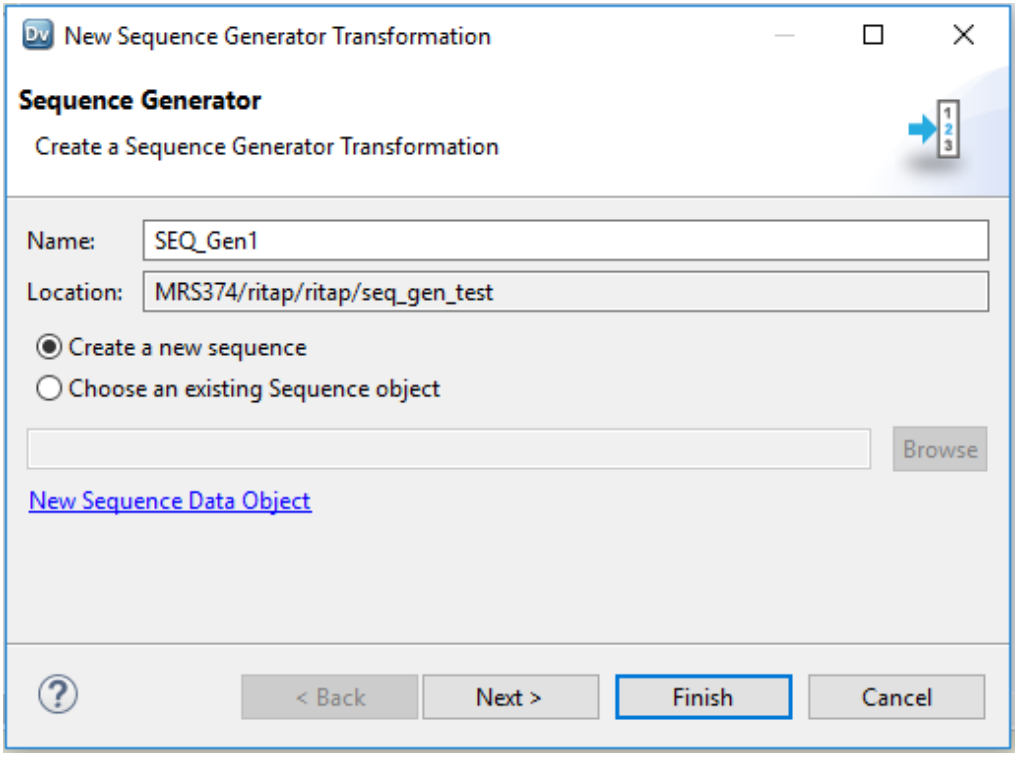
   该序列生成器转换将显示在具有 NEXTVAL 仅输出端口的映射编辑器中。可以将 NEXTVAL 端口连接到下游转换或目标来生成数字序列。

创建序列生成器转换

要在映射中使用序列生成器转换，请将其添加到映射中并配置转换属性，然后将 NEXTVAL 连接到一个或多个转换。

1. 在映射编辑器中，在映射选项板中向下滚动以找到该序列生成器转换，然后将其拖动到映射中。

   此时将打开**新建转换**向导。

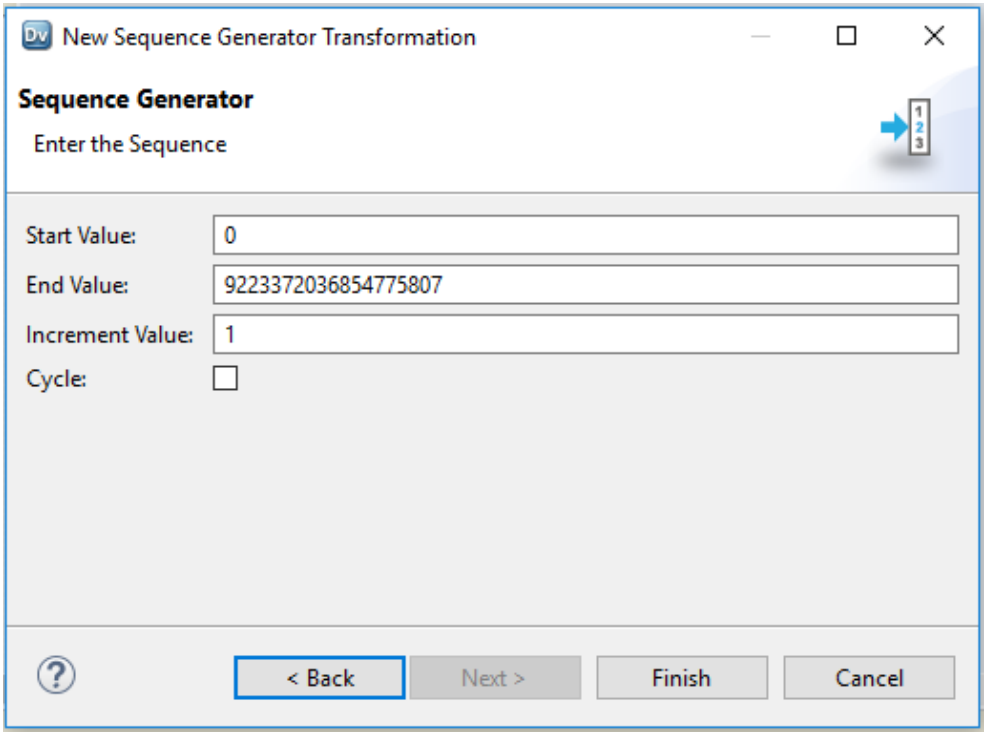


2. 输入序列生成器转换的名称。

   序列生成器转换的命名约定是 SEQ_<转换名称>。

3. 选择要创建新序列还是使用现有序列对象。

- 要创建新序列，请选择**创建新序列**。单击**下一步**以配置序列属性。下图显示了可以配置的属性：



- 要使用现有序列对象，请单击**选择现有序列对象**。导航到该序列对象，然后单击**确定**。

该序列生成器转换将显示在具有 NEXTVAL 仅输出端口的映射编辑器中。可以将 NEXTVAL 端口连接到下游转换或目标来生成数字序列。

常见问题

我是否可以更改不可重用序列生成器转换以将其设置为可重用？

您无法使该转换重新可用，但可以将该转换改为使用序列数据对象。无论有多少转换在使用序列数据对象，该对象都可以保持序列的完整性。

我是否可以在 Mapplet 中放入一个不可重用序列生成器转换？

不，您不能。Mapplet 是可重用对象，因此其中的所有对象也必须是可重用对象。可改用可重用序列生成器转换。

非本地环境下的序列生成器转换

非本地环境下的序列生成器转换处理取决于运行此转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。

- Databricks Spark 引擎。当域和 Databricks 群集位于 AWS 或 Azure Databricks 环境中的同一虚拟网络中时，支持此引擎。

Blaze 引擎上的序列生成器转换

当以下条件成立时，包含序列生成器转换的映射会消耗大量资源：

- 您可以配置转换以维持行顺序。
- 映射在单个分区上运行。

Spark 引擎上的序列生成器转换

序列生成器转换不会在输出数据中维持行顺序。如果在转换中启用了**维持行顺序**属性，数据集成服务将会忽略此属性。

注意: 如果数据集相对较小并且可以在单个分区上运行或强制运行，则配置了维持行顺序的 Spark 序列生成器可以正常工作。

第 42 章

排序器转换

本章包括以下主题：

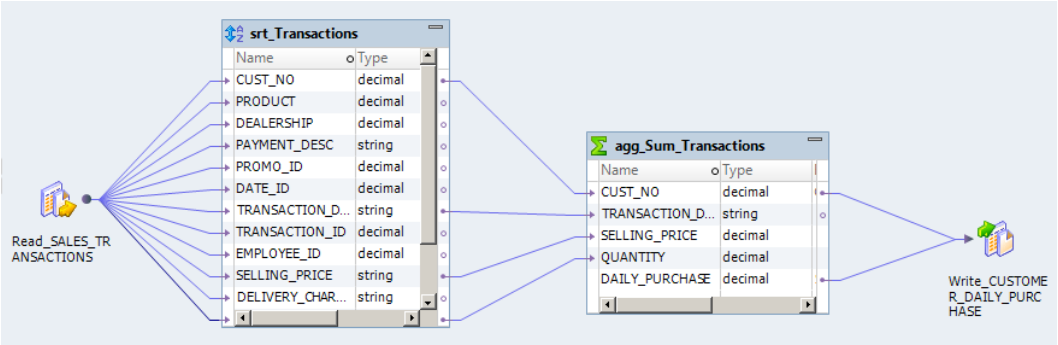
排序器转换概览

使用排序器转换可以根据指定的排序键按升序或降序对数据排序。 可以配置排序器转换，以便进行区分大小写的排序以及生成相异输出。 排序器转换是一种主动转换。

创建排序器转换时，可以指定端口作为排序键，并配置每个排序键，以便按升序或降序排序。 如果指定多个端口作为排序键，则数据集成服务将依次对每个端口进行排序。

例如，您需要为客户数据库中的总体客户销售额创建一张发票。 可以对客户销售额表使用排序器转换，根据客户编号按降序对数据排序。 然后，使用排序器转换的结果作为汇总器转换的输入。 您可以通过已排序输入选项提高汇总器转换的性能。

下图显示了该映射：



动态映射中的排序器转换

您可以在动态映射中使用排序器转换。 可以在转换中配置动态端口，并引用生成的端口。

您可以在排序器转换中引用动态端口或生成的端口。 但是，如果生成的端口在运行时不存在，则映射将会失败。

如果您使用动态端口作为排序键，数据集成服务将考虑动态端口中所有生成的端口，还将考虑生成的端口顺序。

您可以参数化排序键。 将排序列表参数用于排序键。

开发排序器转换

开发排序器转换时，需要考虑诸如排序键端口、相异输出行和区分大小写的排序条件等因素。

开发排序器转换时，请考虑以下因素：

- 要配置为排序键和排序方向的端口。
- 排序是否要区分大小写。
- 是否要将空值作为优先排序。
- 是否要相异输出行。
- 要设置的排序器缓存大小的值。

排序器转换端口

在排序器转换中创建端口时，默认创建输入和输出端口。 排序器转换返回与输入端口相同的输出端口。

可以在排序器转换中定义动态端口。 动态端口可以从映射中的上游转换接收不同的数据列。 这样，排序器转换便可对包含不同列的行进行排序。

在**属性**视图的**排序**选项卡中定义排序键。

排序选项卡

在**属性**视图中“排序器转换”的**排序**选项卡上定义排序键。 选择要用作排序条件的一个或多个端口。

数据集成服务按照“排序器”选项卡上的端口顺序对数据进行排序。 配置为按照升序或降序对数据排序。 默认设置为升序。

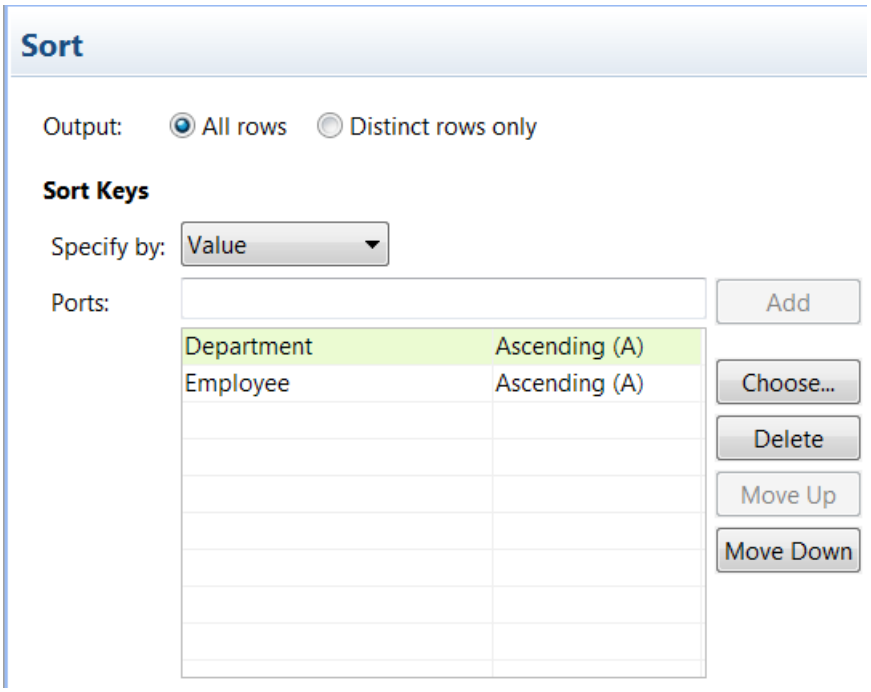
如果将排序器转换配置为生成相异的输出行，则 Developer tool 会将所有端口配置为排序键的一部分。 数据集成服务会丢弃在排序操作期间比较出的重复行。

配置排序键

在**属性**视图的**排序**选项卡中定义排序键。

这就是概念的开始。

下图显示了**排序**选项卡：



排序选项卡包含以下选项：

输出

选择是返回所有已排序行还是丢弃重复行。 重复行是指所有列值都相同的行。

指定依据

选择**值**或**参数**。 选择**值**可使用端口名称。 选择**参数**可使用排序列表参数。

添加

接受手动键入的端口名称。 单击**添加**之前必须键入有效的端口名称。

选择

单击**选择**可选择要添加到排序键中的端口。 Developer tool 提供一个来自转换的端口列表，您可从中进行选择。

“上移”和“下移”

可以更改组中的端口顺序。 选择端口名称，然后单击其中一个移动按钮，可在排序顺序中将其上移或下移。

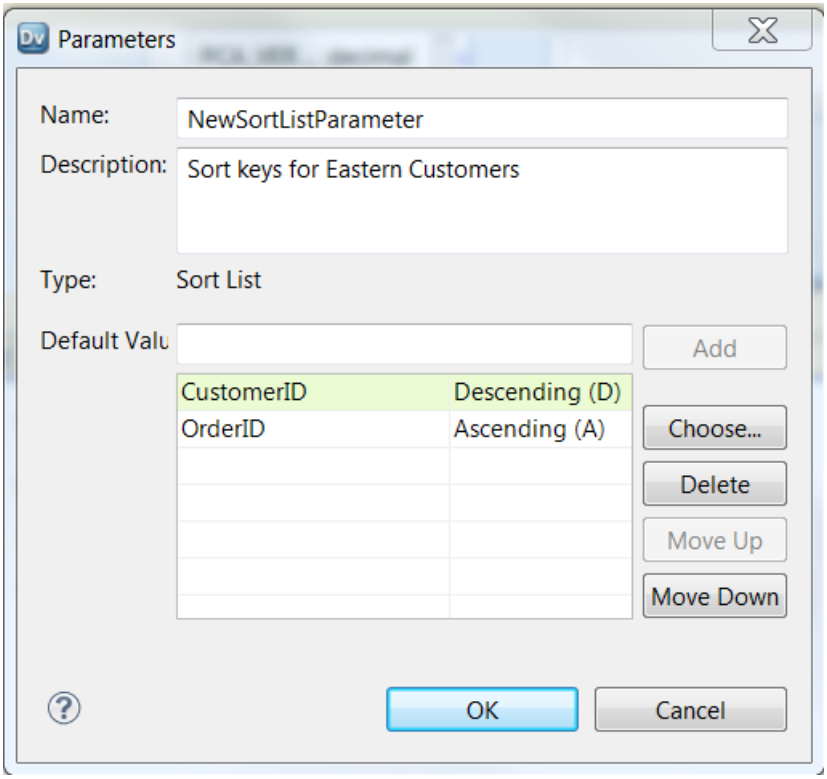
参数化排序键

可以为排序键创建一个包含端口列表的排序列表参数。

如果排序转换采用动态映射，则排序器转换可能包含生成的端口。 您可以参数化排序键。 创建一个排序列表参数，让其包含排序所依据的端口的列表。

在转换属性的**排序**选项卡上，选择**指定依据参数**。 单击**新建**创建一个参数。

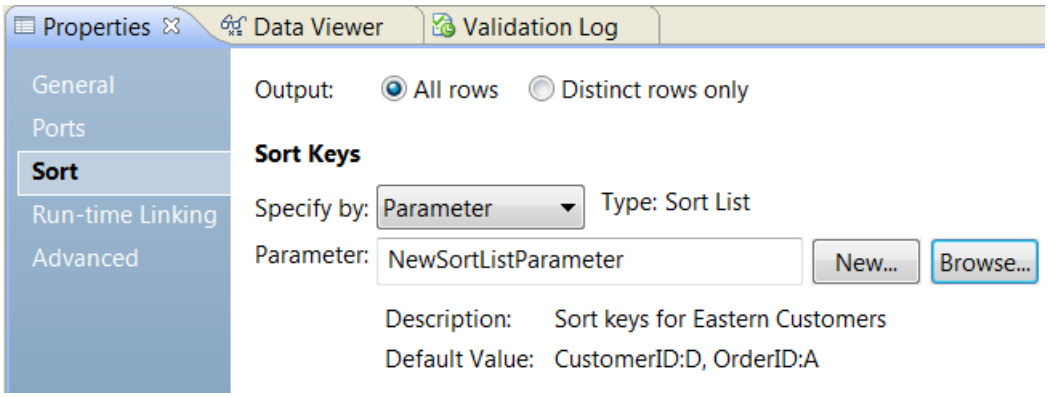
下图显示了**参数**对话框：



选择端口或生成的端口作为排序键。 您可以升序或降序排序类型。

可以手动输入端口名称。 在**默认值**字段中键入端口名称，然后单击**添加**。 Developer tool 会将此端口名称添加到排序列表中。

下图显示了您为排序键配置参数后的**排序**选项卡：



排序器转换高级属性

可以在排序器转换高级属性中指定其他的排序条件。 数据集成服务将这些属性应用于所有排序键端口。 排序器转换属性还可确定数据集成服务在对数据排序时分配的系统资源。

以下部分介绍了排序器转换的高级属性：

区分大小写

确定数据集成服务在对数据排序时是否考虑大小写。 如果启用了“区分大小写”属性，则数据集成服务会将大写字符排在小写字符之前。 默认情况下，Developer tool 会设置“区分大小写”。

空值视为低值

将空值视为比其他任何值都低。 如果希望数据集成服务在执行排序操作时将空值视为比其他任何值都低，请启用此属性。

排序器缓存大小

数据集成服务在映射运行开始时为执行排序操作所分配的内存量。数据集成服务在执行排序操作之前会将所有传入数据传递到排序器转换。 选择“自动”可让数据集成服务自动计算运行时的内存需求。在调整缓存大小时输入特定值（以字节为单位）。默认为“自动”。

工作目录

数据集成服务在传入数据量大于排序器缓存大小时用于临时存储数据的目录。数据集成服务对数据进行排序后，会删除临时文件。

输入多个以分号分隔的目录，以提高缓存分区期间的性能。缓存分区会为处理转换的每个分区创建一个单独的缓存。

默认值是 TempDir 系统参数。您可以在此字段中配置另一系统参数或用户定义的参数。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

相关主题：

- “缓存大小” 页面上 67

排序器缓存

数据集成服务在内存中创建缓存来运行排序器转换。数据集成服务在执行排序操作之前会将所有传入数据传递到排序器转换。如果数据集成服务需要的空间大于内存缓存中的可用空间，它会将数据临时存储在排序器转换的工作目录中。

如果您未将缓存大小配置为在内存中对所有数据进行排序，会话日志中将显示一条警告，指示数据集成服务对源数据执行了多次传递。当数据集成服务必须将信息分页到磁盘才能完成排序时，它会对数据进行多次传递。该消息将指出单次传递（即数据集成服务一次性读取数据并在内存中执行排序而不分页到磁盘）所需的内存量。要优化映射性能，请配置适当的缓存大小，以使数据集成服务对数据执行一次传递。

如果传入的数据量大于排序器缓存大小，数据集成服务会将数据临时存储在排序器转换的工作目录中。对工作目录中的数据进行排序时，数据集成服务所需的磁盘空间至少为传入数据量的两倍。

要达到最佳性能，请将排序器缓存大小配置为一个小于或等于运行映射的计算机的可用物理内存量的值。要使用排序器转换对数据进行排序，请至少分配 16 MB（16,777,216 个字节）物理内存。默认情况下，排序器缓存大小设置为“自动”。

优化排序器缓存

排序器缓存已经过优化，可使用可变长度来存储通过排序器转换的 binary 和 string 数据类型。

可变长度可以减少数据集成服务存储在排序器缓存中的数据量，并降低数据集成服务计算机上的磁盘空间占用量。

例如，您存储客户的相关数据。有些客户的姓名比其他客户要长。如果数据集成服务使用固定长度存储客户名称的相关数据，则可能会为每个姓名存储 20 个字符的数据。如果数据集成服务使用可变长度，则可能存储平均长度为 10 个字符的数据。

创建排序器转换

可以创建可重用或不可重用排序器转换。

创建可重用排序器转换

可创建可重用排序器转换以供在多个映射或 Mapplet 中使用。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择排序器转换。
4. 单击**下一步**。
5. 输入转换的名称。

6. 单击**完成**。

   此时，转换将显示在编辑器中。

7. 单击**新建**向转换添加端口。
8. 编辑端口以设置名称、数据类型和精度。
9. 在**排序**选项卡上，选择要排序的端口或者选择排序列表参数。
10. 单击**高级**视图并编辑转换属性。

创建不可重用排序器转换

可在映射或 Mapplet 中创建不可重用排序器转换。

1. 在映射或 Mapplet 中，将“转换”选项板中的排序器转换拖动到编辑器中。

   此时，转换将显示在编辑器中。

2. 在**属性**视图中，编辑转换名称和说明。
3. 在**端口**选项卡上，单击**新建**将端口添加到转换中。
4. 编辑端口以设置名称、数据类型和精度。
5. 选择**键**将端口指示为排序键。
6. 单击**高级**选项卡然后编辑转换属性。

排序器转换示例

假定您有一个数据库表 PRODUCT_ORDERS，其中包含客户下达的所有订单的信息。

ORDER_ID	ITEM_ID	ITEM	QUANTITY	PRICE
43	123456	ItemA	3	3.04
41	456789	ItemB	2	12.02
43	000246	ItemC	6	34.55
45	000468	ItemD	5	0.56
41	123456	ItemA	4	3.04
45	123456	ItemA	5	3.04
45	456789	ItemB	3	12.02

对 PRODUCT_ORDERS 使用排序器转换，并指定 ORDER_ID 作为排序键，且方向为降序。

对数据排序后，数据集成服务会从排序器转换中传递出以下行：

ORDER_ID	ITEM_ID	ITEM	QUANTITY	PRICE
45	000468	ItemD	5	0.56
45	123456	ItemA	5	3.04
45	456789	ItemB	3	12.02

ORDER_ID	ITEM_ID	ITEM	QUANTITY	PRICE
43	123456	ItemA	3	3.04
43	000246	ItemC	6	34.55
41	456789	ItemB	2	12.02
41	123456	ItemA	4	3.04

您需要了解每个订单的总金额和总商品数量。 可以使用排序器转换的结果作为汇总器转换的输入。 在汇总器转换中使用已排序输入可以提高性能。

如果不使用已排序输入，则数据集成服务将在读取时执行汇总计算。 数据集成服务将存储每个组的数据，直至读取整个源，以确保所有汇总计算准确无误。 如果使用已排序输入，但没有预先对数据进行正确排序，则会产生意外结果。

汇总器转换具有 ORDER_ID 分组依据端口，并选中了已排序输入选项。 将排序器转换的数据传递到汇总器转换后，汇总器转换会对 ORDER_ID 分组以计算每个订单的总金额。

ORDER_ID	SUM
45	54.06
43	216.42
41	36.2

非本地环境下的排序器转换

非本地环境下的排序器转换处理取决于运行此转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射和流映射中有限支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的排序器转换

Blaze 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

在下列情况下，映射验证会失败：

- 目标配置为维持行顺序，而未将排序器转换直接连接至平面文件目标。

并行排序

数据集成服务支持受以下限制的并行排序：

- 映射在排序器转换和目标之间不包含其他转换。

- 排序器转换和目标之间的排序键数据类型一致。
- 排序器转换中的每个排序键都必须链接至目标中的某一列。

全局排序

在下列情况下，Blaze 引擎可以执行全局排序：

- 排序器转换直接连接至平面文件目标。
- 目标配置为维持行顺序。
- 排序键不是二进制数据类型。

如果上述任何一个条件不成立，则 Blaze 引擎将会执行本地排序。

数据缓存优化

如果在汇总器转换或等级转换之前插入了一个排序器转换来优化汇总器或等级数据缓存，则排序器缓存的大小与汇总器转换或等级转换的数据缓存大小相同。要配置排序器缓存，必须配置汇总器转换或等级转换的数据缓存大小。

Spark 引擎上的排序器转换

Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

如果禁用区分大小写，则映射验证将会失败。

在下列情况下，数据集成服务会记录一条警告并忽略排序器转换：

- 目标和排序器转换之间的排序键类型不匹配。
- 转换中含有未连接至目标的排序键。
- 写入转换未配置为维持行顺序。
- 转换不直接位于写转换的上游。

空值

即使将转换配置为将空值处理为高，数据集成服务仍会将空值处理为低。

数据缓存优化

无法通过优化排序器缓存来使用可变长度存储数据。

流映射中的排序器转换

流映射具有不适用于批处理映射的其他处理规则。

映射验证

在下列情况下，映射验证会失败：

- 流映射中的排序器转换没有上游汇总器转换。

一般准则

请考虑以下一般准则：

- 如果映射包含排序器转换，则以完整输出模式运行。
- 要保持全局排序顺序，请确保目标是单分区。源可以是单分区也可以是多分区。

Databricks Spark 引擎上的排序器转换

Databricks Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

映射验证

如果禁用区分大小写，则映射验证将会失败。

在下列情况下，数据集成服务会记录一条警告并忽略排序器转换：

- 目标和排序器转换之间的排序键类型不匹配。
- 转换中含有未连接至目标的排序键。
- 写入转换未配置为维持行顺序。
- 转换不直接位于写转换的上游。

空值

即使将转换配置为将空值处理为高，数据集成服务仍会将空值处理为低。

数据缓存优化

无法通过优化排序器缓存来使用可变长度存储数据。

并行排序

数据集成服务支持受以下限制的并行排序：

- 映射在排序器转换和目标之间不包含其他转换。
- 排序器转换和目标之间的排序键数据类型一致。
- 排序器转换中的每个排序键都必须链接至目标中的某一列。

第 43 章

SQL 转换

本章包括以下主题：

SQL 转换概览

SQL 转换可处理映射中的 SQL 查询中游。 可以通过 SQL 转换运行 SQL 查询，或者可以配置 SQL 转换以通过数据库运行存储过程。

可以将输入端口值传递给查询或存储过程中的参数。 转换可以在数据库中插入、删除、更新和检索行。 可以运行 SQL DDL 语句，以便在映射中创建表或删除表中游。 SQL 转换是主动转换。该转换可以为每个输入行返回多个行。

可以将存储过程从数据库导入到 SQL 转换。 导入存储过程时，开发程序工具将创建与存储过程中的参数对应的转换端口。 开发程序工具还会为您创建存储过程调用。

要配置 SQL 转换以运行存储过程，请执行以下任务：

1. 定义转换属性，包括要连接到的数据库类型。
2. 导入存储过程以定义端口并创建存储过程调用。
3. 手动为需要运行的结果集或其他存储过程定义端口。
4. 在 SQL 编辑器中添加其他存储过程调用。

可以在转换 SQL 编辑器中配置 SQL 查询。 运行 SQL 转换时，该转换会处理查询，返回行，并返回任何数据库错误。

要配置 SQL 转换以运行查询，请执行以下任务：

1. 定义转换属性，包括要连接到的数据库类型。
2. 定义输入和输出端口。
3. 在 SQL 编辑器中创建 SQL 查询。

配置转换后，在映射中配置 SQL 转换，并连接上游端口。 预览数据以验证结果。

SQL 转换端口

创建 SQL 转换时，默认情况下 Developer 工具会创建 SQLError 端口。 可在**端口**视图中添加输入端口、输出端口和传递端口。

SQL 转换具有以下类型的端口：

输入

接收可用于 SQL 查询的源数据。

输出

返回 SQL SELECT 查询的数据库数据。

传递

输入-输出端口，用于将源数据不加更改地传递给转换。

SQLError

从数据库返回 SQL 错误。 如果未出现错误，则返回空值。

NumRowsAffected

为输入行返回受 INSERT、DELETE 和 UPDATE 查询语句影响的数据库总行数。 如果选择在输出行中包含更新统计信息，则 Developer 工具会创建此端口。

返回值

从存储过程接收返回值。

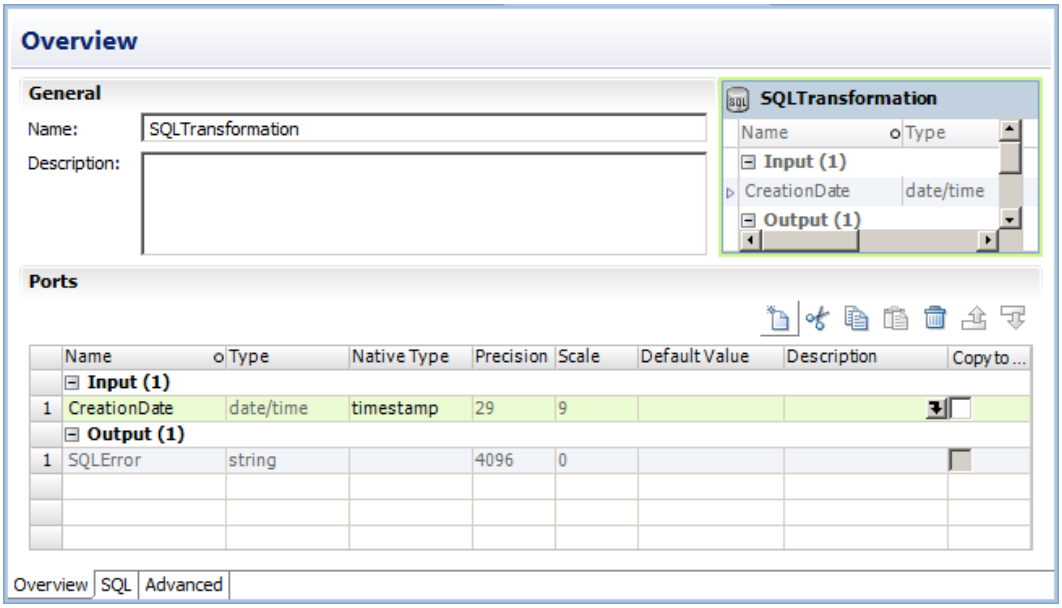
输入端口

您可以使用绑定在任意类型的 SQL 语句或存储过程中的参数引用 SQL 转换输入端口。 您可以在 SQL 转换中为不想传递至输出端口的数据创建输入端口。

如果配置具有输入参数的 SQL 查询，必须手动添加端口。 将存储过程导入到 SQL 转换时，SQL 转换将创建输入端口。 您可以添加传递端口以将数据传递到该转换而不对其进行更改。

您可以在**概览**视图中添加端口。 添加端口时，请输入该端口的本地数据类型。 本地数据类型是对将连接到的数据库有效的数据类型。 配置本地数据类型时，将显示一个转换数据类型。 如果将行拖至 SQL 转换中，则 Developer 工具会根据对将连接到的数据库有效的数据类型来设置本地数据类型。 验证在查询中使用的列数据类型是否与数据库中的列数据类型相同。

下图显示了可重用 SQL 转换中的 **CreationDate** 输入端口：



要添加输入端口，请在**端口**面板中单击**输入**。 单击**新建**。

注意: 如果为端口选择**复制到输出**，则输入端口将成为传递端口。 传递端口显示在**端口**视图的**输入**和**输出**部分。

输出端口

SQL 转换输出端口从查询语句或从存储过程返回值。

手动配置 SQL 转换时，必须定义输出端口。 为每个存储过程输出参数或者为 SELECT 语句返回的每个端口定义一个输出端口。

导入存储过程时，Developer 工具会为该过程所返回的每个输出参数创建一个输出端口。 如果该过程返回一个结果集，必须在结果集中手动定义输出端口。 存储过程可以返回结果集并可以返回不包含在同一个运行的结果集中的输出参数。 必须为结果集字段和输出参数定义输出端口。

配置输出端口时，请为该端口选择本地数据类型。 输出端口的本地数据类型必须与数据库中对应列的数据类型相匹配。 配置本地数据类型时，Developer 工具将定义该端口的转换数据类型。

例如，SQL 转换包含以下适用于 Oracle 数据库的 SQL 查询：

```
SELECT FirstName, LastName, Age FROM EMPLOYEES
```

可以在 SQL 转换中配置以下输出端口和本地数据类型：

输出端口	本地数据类型	转换数据类型
FirstNm	varchar2	string
LastNm	varchar2	string
Age	number	双精度型

输出端口的数量和输出端口的顺序必须与查询或存储过程所返回的列的数量和顺序相匹配。 如果输出端口数大于查询或存储过程中的列数，多出的端口将返回空值。 如果输出端口数小于 SQL 中的列数，数据集成服务将生成行错误。

如果更改转换连接到的数据库类型，则 Developer 工具将更改输出端口的本地类型。 Developer 工具可能不会为所有端口选择正确的数据类型。 如果更改数据库类型，请验证各个输出端口的本地数据类型是否与数据库中列的数据类型相同。 例如，Developer 工具可能会为 Oracle 数据库的列选择 nVarchar2。 您可能需要将数据类型更改为 varchar2。

在 SQL 转换的**概览**视图中配置输出端口。

传递端口

传递端口是输入输出端口，通过转换传递数据但不更改数据。 无论 SQL 查询是否返回行，SQL 转换都会返回传递端口中的数据。

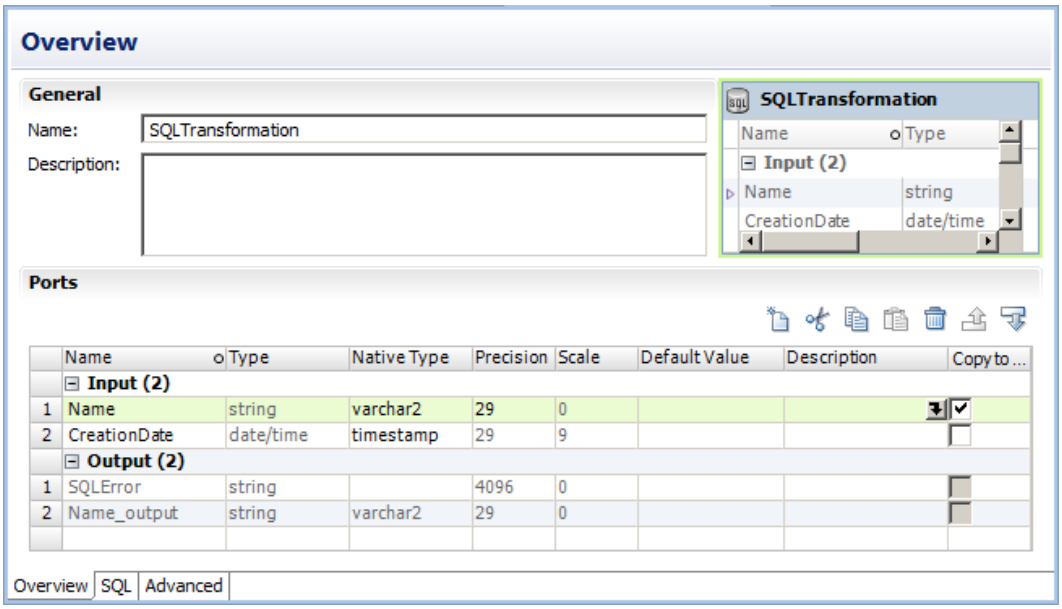
输入行包含 SELECT 查询语句时，SQL 转换会为它从数据库中返回的每行返回传递端口中的数据。 如果查询结果包含多行，SQL 转换会在每行中重复传递数据。

如果查询未返回行，SQL 转换将在输出列中返回带有空值的传递列数据。 例如，包含 INSERT、UPDATE 和 DELETE 语句的查询不返回任何行。 如果查询出现错误，SQL 转换将在输出端口中返回传递列数据、SQLError 消息和空值。

无法通过配置传递端口从 SELECT 查询中返回数据。

要创建传递端口，请创建输入端口并选择**复制到输出**。 Developer 工具会创建输出端口并向端口名添加“_output”后缀。 无法更改 Developer 工具为传递端口创建的输出端口。 无法创建带有“_output”后缀的输出端口。

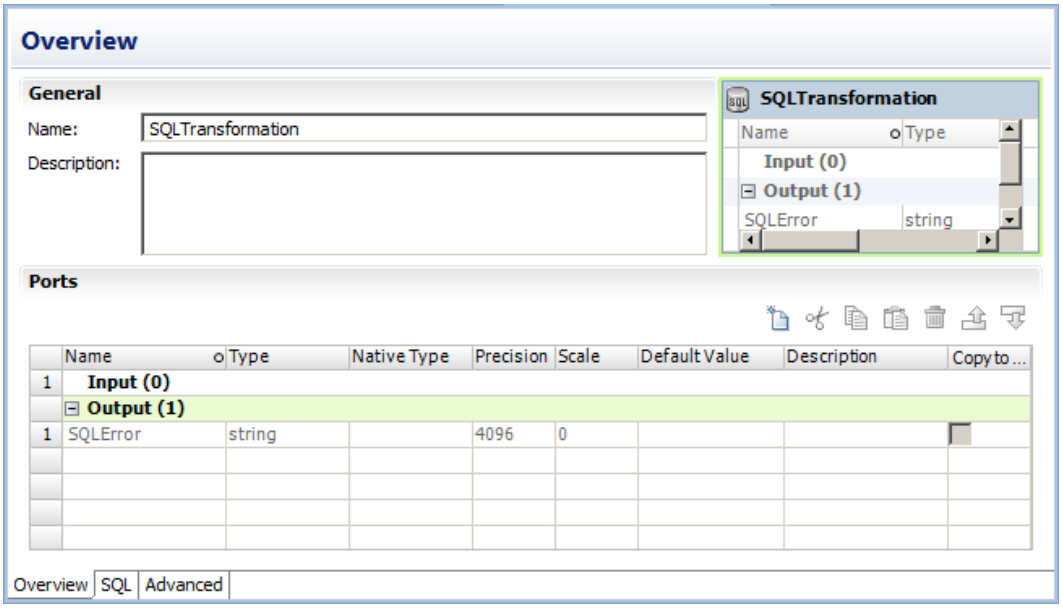
下图显示了可重用 SQL 转换中的“名称”传递端口：



SQLError 端口

SQLError 端口从存储过程或 SQL 查询返回数据库的 SQL 错误。

下图显示了一个可重用 SQL 转换中的 SQLError 端口：



当 SQL 查询包含语法错误时，SQLError 端口会包含数据库中的错误文本。 例如，以下 SQL 查询会从 Oracle 数据库生成一行错误：

```
SELECT Product_ID FROM Employees
```

Employees 表不包含 Product_ID。 数据集成服务将生成一行。 SQLError 端口在一行中包含以下错误文本：

```
ORA-0094: “Product_ID” : invalid identifier Database driver error... Function Name: Execute SQL Stmt:
SELECT Product_ID from Employees Oracle Fatal Error
```

可以在 SQL 查询中配置多个查询语句或者可以调用多个存储过程。 如果将 SQL 转换配置为在出现 SQL 错误时继续，则 SQL 转换可能会为一个查询语句返回行，而为另一个查询语句返回数据库错误。 SQL 转换会在单独的一行中返回任何数据库错误。

受影响的行数

启用 NumRowsAffected 输出端口可返回 INSERT、UPDATE 或 DELETE 查询语句针对每个输入行所更改的行数。 可以为 SQL 查询配置 NumRowsAffected 输出端口。

数据集成服务将为查询中的每个语句返回 NumRowsAffected。默认情况下，NumRowsAffected 处于禁用状态。

启用 NumRowsAffected 但 SQL 查询不包含 INSERT、UPDATE 或 DELETE 语句时，每个输出行中的 NumRowsAffected 均为零。

当 SQL 查询包含多个语句时，数据集成服务将针对每个语句返回 NumRowsAffected。 NumRowsAffected 包含 INSERT、UPDATE 和 DELETE 语句针对某个输入行更改的行数总和。

例如，查询可包含以下语句：

```
DELETE from Employees WHERE Employee_ID = ‘101’;
SELECT Employee_ID, LastName from Employees WHERE Employee_ID = ‘103’;
INSERT into Employees (Employee_ID, LastName, Address)VALUES ( ‘102’, 'Gein', '38 Beach Rd')
```

DELETE 语句影响一行。 SELECT 语句不影响任何行。 INSERT 语句影响一行。

数据集成服务将从 DELETE 语句返回一行。 NumRowsAffected 等于一。 数据集成服务将从 SELECT 语句返回一行，NumRowsAffected 为零。 数据集成服务将从 INSERT 语句返回一行，NumRowsAffected 等于一。

SQL 转换高级属性

您可以随时更改 SQL 转换属性。 默认数据库类型为 Oracle。 如果需要连接的数据库是其他数据库类型，请在将端口添加到转换之前更改数据库类型。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。如果将 SQL 转换跟踪级别配置为“详细数据”，数据集成服务会将准备的每条 SQL 查询写入映射日志。

连接类型

描述数据集成服务如何连接数据库。 连接类型是静态的。 数据集成服务将一次连接到数据库。 在 SQL 转换中选择数据库连接对象。 只读。

数据库类型

SQL 转换连接到的数据库的类型。 从列表中选择一种数据库类型。 可以选择 Oracle、Microsoft SQL Server、IBM DB2 或 ODBC。 数据库类型会影响可在**端口**选项卡上分配的数据类型。 更改数据库类型后，Developer tool 会更改输入、输出和传递端口的端口数据类型。

行内出现错误时继续

出现 SQL 错误后，继续处理查询中的剩余 SQL 语句。

包含统计信息作为输出

添加 NumRowsAffected 输出端口。 该端口为输入行返回 INSERT、DELETE 和 UPDATE 查询语句更新的数据库总行数。

最大输出行计数

定义 SQL 转换可以从 SELECT 查询输出的最大行数。要配置为无限多行，请将“最大输出行计数”设置为零。

查询说明

在转换中定义的 SQL 查询的说明。

SQL 模式

确定 SQL 查询是否为外部脚本或者是否在转换中定义查询。 SQL 模式是“查询”。 SQL 转换运行在 SQL 编辑器中定义的查询。 只读。

SQL 查询

显示在 SQL 编辑器中配置的 SQL 查询。

有副作用

指出 SQL 转换除了返回行之外还执行一个功能。 SQL 查询更新数据库时，SQL 转换会产生副作用。 如果 SQL 查询包含诸如 CREATE、DROP、INSERT、UPDATE、GRANT 或 REVOKE 语句，则启用**有副作用**。

此外，如果 SQL 转换针对未返回结果的 SELECT 语句返回 NULL 行，该转换也产生副作用。这些行可能包含传递端口值、SQL 错误信息或 NUMRowsAffected 字段。

要允许推入优化或早期选择优化，请禁用**有副作用**属性。 默认情况下启用该属性。

仅返回数据库输出

SQL 转换不会为返回 0 个结果的 SELECT 语句生成行，也不会为 INSERT、UPDATE、DELETE 或 COMMIT 之类的其他语句生成行或者空值行。

启用推送优化

使数据集成服务能够在映射中将筛选器转换中的逻辑推送到 SQL 转换中的 SQL 中。

维持行顺序

保持转换输入数据的行顺序。如果数据集成服务不应执行任何可能改变行顺序的优化，请选择此选项。

数据集成服务执行优化时，可能会失去之前在映射中建立的顺序。您可以在具有已排序的平面文件源、已排序的关系源或排序器转换的映射中设置顺序。当您将转换配置为维持行顺序时，数据集成服务将在执行映射优化时考虑此配置。如果数据集成服务可以保持顺序，将为转换执行优化。如果优化会改变行顺序，数据集成服务将不会为转换执行优化。

可分区

可以使用多个线程处理转换。如果希望数据集成服务使用一个线程来处理转换，请清除此选项。 数据集成服务可以使用多个线程处理余下的映射管道阶段。

当 SQL 查询要求使用一个线程处理转换时，为 SQL 转换禁用分区。或者，您可能会为了仅建立一个与数据库的连接而希望为 SQL 转换禁用分区。

SQL 转换查询

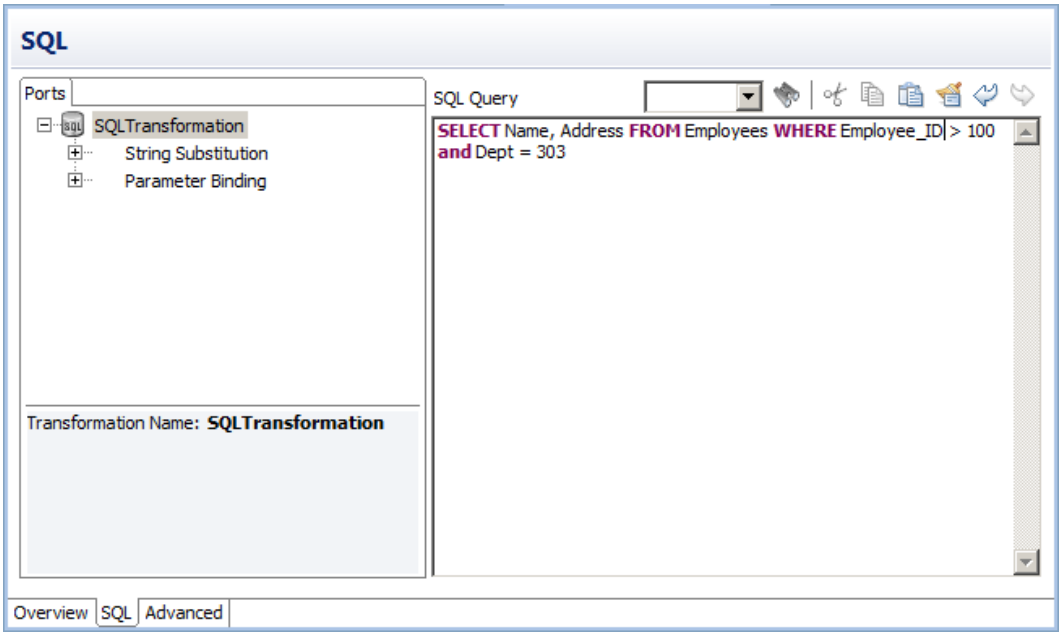
在 SQL 编辑器中创建 SQL 查询以从数据库检索行或更新数据库。

要创建查询，请在 SQL 编辑器的“SQL”视图中键入查询语句。可以在 SQL 查询语句中最多输入 32767 个字符。SQL 编辑器提供了可在查询中引用的转换端口列表。您可以右键单击某个端口名称以将其添加为查询参数。

创建查询时，SQL 编辑器会验证查询中的端口名称。它还会验证用于字符串替换的端口是否为字符串数据类型。SQL 编辑器不会验证 SQL 查询的语法。

可以在 SQL 查询中使用常量。将每个字符串都用单引号 (') 引上。

下图显示了一个示例 SQL 查询：



可以创建静态 SQL 查询。查询语句不会变，但可以加入参数以更改值。数据集成服务会为每个输入行运行此查询。

定义 SQL 查询

定义一个 SQL 查询，用于为每个输入行运行同一查询语句。 可以基于行中的输入端口值更改查询列或表。 还可以基于输入端口值更改 WHERE 子句中的值。

要为每个输入行更改 WHERE 子句中的数据值，请配置参数绑定。

要基于输入端口值更改查询列或更改表，请使用字符串替换。

参数绑定

要更改查询中的数据，请配置查询参数并将其绑定到转换中的输入端口。 将参数绑定到输入端口后，可按查询中的名称标识端口。 SQL 编辑器会用问号 (?) 将端口名称括起来。 查询数据将基于端口中数据的值进行更改。

以下查询使用了参数绑定：

```
DELETE FROM Employee WHERE Dept = ?Dept?
INSERT INTO Employee(Employee_ID, Dept) VALUES (?Employee_ID?, ?Dept?)
UPDATE Employee SET Dept = ?Dept? WHERE Employee_ID > 100
```

以下 SQL 查询包含绑定到 SQL 转换的 Employee_ID 和 Dept 输入端口的查询参数：

```
SELECT Name, Address FROM Employees WHERE Employee_Num =?Employee_ID? and Dept = ?Dept?
```

源可能包含以下行：

Employee_ID	Dept
100	Products
123	HR
130	Accounting

数据集成服务将从这些行中生成以下查询语句：

```
SELECT Name, Address FROM Employees WHERE Employee_ID =  ‘100’  and DEPT =  ‘Products’
SELECT Name, Address FROM Employees WHERE Employee_ID =  ‘123’  and DEPT =  ‘HR’
SELECT Name, Address FROM Employees WHERE Employee_ID =  ‘130’  and DEPT =  ‘Accounting’
```

字符串置换

使用字符串变量替换查询语句的组件。 例如，您可以使用字符串变量替换查询中的表名称。 或者，您可以置换 SELECT 语句中的列名称。

要置换表名称，请配置输入端口以接收每个输入行的表名称。 在 SQL 编辑器中，从端口的**字符串置换**列表中选择端口。 Developer 工具会根据查询中的名称标识输入端口，并使用颚化符 (~) 将该名称括起来。

以下查询包含字符串变量 ~Table_Port~：

```
SELECT Emp_ID, Address from ~Table_Port~ where Dept =  ‘HR’
```

源可能将以下值传递到 **Table_Port** 列：

Table_Port

Employees_USA

Employees_England

Employees_Australia

数据集成服务将 ~Table_Port~ 变量替换为输入端口中的表名称值：

```
SELECT Emp_ID, Address from Employees_USA where Dept =  ‘HR’
SELECT Emp_ID, Address from Employees_England where Dept =  ‘HR’
SELECT Emp_ID, Address from Employees_Australia where Dept =  ‘HR’
```

输入行至输出行基数

数据集成服务运行 SELECT 查询时，SQL 转换将为检索到的每一行返回一行。 如果查询未检索到数据，则 SQL 转换将为每一输入行返回零行或一行。

查询语句处理

SELECT 查询成功时，SQL 转换可能会检索到多行。 如果查询包含其他语句，则数据集成服务可能会生成包含 SQL 错误或受影响行数的行。

端口配置

NumRowsAffected 输出端口包含 UPDATE、INSERT 或 DELETE 语句针对每个输入行所更改的行数。 SQL 转换将针对查询中的每个语句返回受影响的行数。 如果 SQL 转换包含传递端口，则该转换将至少针对每个源行返回一次列数据。

最大行计数配置

“最大输出行计数”限制了 SQL 转换为 SELECT 查询返回的行数。

错误行

数据集成服务会在遇到连接错误或语法错误时返回行错误。 SQL 转换会将这些错误返回到 SQLError 端口。

出现 SQL 错误时继续

可以配置 SQL 转换以使其在 SQL 语句中存在错误时继续处理。 此时 SQL 转换不会生成行错误。

查询语句处理

SQL 查询类型确定 SQL 转换返回的行数。 SQL 转换可返回零行、一行或多行。 查询包含 SELECT 语句时，SQL 转换会将数据库中的每一列返回到输出端口。 此转换将返回所有满足条件的行。

下表列出了在查询模式下未发生错误时 SQL 转换针对不同类型的查询语句所生成的输出行：

查询语句	输出行
仅 UPDATE、INSERT、DELETE	针对查询中的每个语句都生成一行。
一个或多个 SELECT 语句	检索到的数据库行总数。
DDL 查询，例如 CREATE、DROP、TRUNCATE	针对查询中的每个语句都生成一行。

端口配置

启用“包含统计信息作为输出”时，Developer 工具将创建 NumRowsAffected 端口。 根据 SQL 查询中的语句，数据集成服务至少返回一行带有 NumRowsAffected 的值。

下表列出了启用 NumRowsAffected 时 SQL 语句将生成的输出行：

查询语句	输出行
仅 UPDATE、INSERT、DELETE	为语句中每个带有 NumRowsAffected 的语句生成一行
一个或多个 SELECT 语句	检索到的数据库行总数。 在每行中，NumRowsAffected 为零。
DDL 查询，例如 CREATE、DROP、TRUNCATE	在 NumRowsAffected 为零时生成一行。

最大输出行计数

可以限制 SQL 转换为 SELECT 查询返回的行数。 可以配置**最大输出行计数**属性来限制行数。 当查询中包含多个 SELECT 语句时，SQL 转换将限制所有 SELECT 语句的总行数。

例如，将**最大输出行计数**设置为 100。 查询中包含两个 SELECT 语句：

```
SELECT * FROM table1; SELECT * FROM table2;
```

如果第一个 SELECT 语句返回 200 行，第二个 SELECT 语句返回 50 行，则 SQL 转换将返回第一个 SELECT 语句中的 100 行。 SQL 转换将不会返回第二个语句中的任何行。

要将输出行数配置为不受限制，请将**最大输出行计数**设置为零。

错误行

数据集成服务遇到连接错误或语法错误时会返回行错误。 SQL 转换将 SQL 错误返回到 SQLError 端口。

配置传递端口或 NumRowsAffected 端口后，SQL 转换将针对每个源行至少返回一行。 查询未返回数据时，SQL 转换将返回传递数据和 NumRowsAffected 值，但在输出端口中将返回空值。 您可以利用筛选器转换传递输出行以删除带有空值的行。

下表介绍了 SQL 转换针对 UPDATE、INSERT 或 DELETE 查询语句生成的行：

已配置 NumRowsAffected 端口或传递端口	SQLError	行输出
两个端口均未配置	否	在 SQLError 端口中生成一个包含空值的行。
两个端口均未配置	是	在 SQLError 端口中生成一个包含错误的行。
配置了其中一个端口	否	针对每条查询语句均生成一个含有 NumRowsAffected 列数据或传递列数据的行。
配置了其中一个端口	是	在 SQLError 端口中生成一个包含错误且带有 NumRowsAffected 端口数据或传递端口数据的行。

下表介绍了 SQL 转换针对 SELECT 语句生成的输出行数：

已配置 NumRowsAffected 端口或传递端口	SQLError	行输出
两个端口均未配置	否	基于从每个 SELECT 语句返回的行生成一行或多行。
两个端口均未配置	是	生成一行（大于成功语句的总输出行数）。 在 SQLError 端口中，最后一行包含错误。
配置了其中一个端口	否	基于针对每个 SELECT 语句返回的行生成一行或多行： - 如果启用 NumRowsAffected，每行中将包含一个值为零的 NumRowsAffected 列。 - 如果配置了传递端口，每行中将包含传递列数据。 当查询返回多行时，每行中的传递列数据都是重复的。
配置了其中一个端口	是	基于针对每个 SELECT 语句返回的行生成一行或多行。 在 SQLError 端口中，最后一行包含错误： - 如果启用 NumRowsAffected，每行中将包含一个值为零的 NumRowsAffected 列。 - 如果配置了传递端口，每行中将包含传递列数据。 当查询返回多行时，每行中的传递列数据都是重复的。

下表介绍了 SQL 转换针对 DDL 查询（例如 CREATE、DROP 或 TRUNCATE）生成的输出行数：

已配置 NumRowsAffected 端口或传递端口	SQLError	行输出
两个端口均未配置	否	在 SQLError 端口中生成一个包含空值的行。
两个端口均未配置	是	在 SQLError 端口中生成一个包含错误的行。
配置了其中一个端口	否	生成一个包含值为零的 NumRowsAffected 列数据和传递列数据的行。
配置了其中一个端口	是	在 SQLError 端口中生成一个包含错误且带有值为零的 NumRowsAffected 列或传递列数据的行。

出现 SQL 错误时继续

您可以选择忽略出现在查询语句中的 SQL 错误。 启用**行内出现 SQL 错误时继续**。 数据集成服务将针对该行继续运行剩余的 SQL 语句。

数据集成服务不会生成行错误。 但是，SQLError 端口会包含失败的 SQL 语句和错误消息。

例如，查询可能包含以下语句：

```
DELETE FROM Persons WHERE FirstName = ‘Ed’;
INSERT INTO Persons (LastName, Address) VALUES ('Gein', '38 Beach Rd')
```

如果 DELETE 语句失败，SQL 转换将从数据库返回错误消息。 数据集成服务将继续处理 INSERT 语句。

禁用**出现 SQL 错误时继续**选项可以排除数据库错误，并将错误与导致错误的查询语句关联。

SQL 转换的筛选器优化

如果筛选条件仅引用传递端口，而 SQL 转换不会产生负面影响，则数据集成服务可以对该 SQL 转换应用筛选器优化。

在下列情况下，SQL 转换会产生负面影响：

- SQL 查询更新数据库。 SQL 查询包含如下语句：CREATE、DROP、INSERT、UPDATE、GRANT 或 REVOKE。
- 转换对未返回任何结果的 SELECT 语句返回空行。 这些行可能包含传递端口值、SQL 错误信息或 NUMRowsAffected 字段。

数据集成服务可对 SQL 转换应用早期选择和推入优化方法。

通过 SQL 转换执行早期选择优化

如果筛选条件仅引用传递端口，并且 SQL 转换没有副作用，则数据集成服务可以通过 SQL 转换执行早期选择优化。

在下列情况下，SQL 转换会产生负面影响：

- SQL 查询更新数据库。 SQL 查询包含如下语句：CREATE、DROP、INSERT、UPDATE、GRANT 或 REVOKE。
- 转换对未返回任何结果的 SELECT 语句返回空行。 这些行可能包含传递端口值、SQL 错误信息或 NUMRowsAffected 字段。

启用通过 SQL 转换执行早期选择优化

如果 SQL 转换没有副作用，则可以在 SQL 转换中启用早期选择优化。

1. 在 SQL 转换**高级属性**中启用**仅返回数据库输出**选项。
2. 清除 SQL 转换**高级属性**中的**有副作用**。
3. 如果转换具有 **NumAffectedRows** 端口，请删除该端口。

通过 SQL 转换执行推入优化

通过推入优化，数据集成服务可以将筛选器逻辑从映射中的筛选器转换推送到 SQL 转换中的查询。

启用通过 SQL 转换执行推入优化时，请遵循以下规则和准则：

- 转换 SQL 查询中必须仅包含 SELECT 语句。
- 转换 SQL 查询必须为有效的子查询。
- 筛选条件不能引用 SQL 错误或 NumRowsAffected 字段。
- 输出端口的名称必须与 SQL SELECT 语句中的列名称匹配。 在筛选条件中引用输出端口时，数据集成服务会将相应的端口名称推送到 SQL 查询。 如果查询中的列与输出端口名称不匹配，可以向 SQL 添加别名。 例如 `SELECT mycolname1 AS portname1, mycolname2 AS portname2`。
- 转换不能有副作用。

通过 SQL 转换执行推入优化示例

SQL 转换按客户 ID 检索订单。 在 SQL 转换之后显示的筛选器转换仅返回订购量大于 1000 的行。

数据集成服务将以下筛选器推送到 SQL 转换中的 SELECT 语句：

```
orderAmount > 1000
```

SQL 查询中的每个语句都变为包含该筛选器的 SELECT 语句的独立子查询。

以下查询语句会将原始查询语句显示为 SELECT 语句中的子查询：

```
SELECT <customerID>, <orderAmount>, ... FROM (original query statements) ALIAS WHERE <orderAmount> > 1000
```

如果 SQL 查询具有多个语句，每个语句将包含在一个独立的子查询中。 子查询具有相同的语法，包括 WHERE 子句。

端口 *customerID* 和 *orderAmount* 是指 SQL 转换中输出端口的名称。 子查询不包含传递端口、SQL 错误或 SQL 统计信息端口。 如果您将多个筛选器推送到 SQL 转换，WHERE 子句中将包含所有筛选器。

启用通过 SQL 转换执行推入优化

通过在 SQL 转换**高级属性**选项卡中配置属性，可以启用推入优化。

1. 清除**有副作用**。
2. 启用**仅返回数据库输出**。
3. 将**最大输出行计数**设置为零。
4. 启用推入优化。

使用 SQL 查询的 SQL 转换示例

您是 Hypostores 公司人力资源部门的开发人员。 Hypostores 采用与人力资源员工数据不同的数据库来维护员工薪资信息。 人力资源部门需要集中查询不同区域的员工和薪资信息。

您希望创建一个逻辑数据对象映射，该映射可通过一个员工逻辑数据对象集中显示员工数据和薪资数据。

可使用员工数据源创建一个逻辑数据对象映射， 并包含一个 SQL 转换，以便从薪资数据库中检索薪资和雇用日期。

逻辑数据对象映射

逻辑数据对象映射包含以下对象：

员工表

从人力资源数据库中输入员工数据的关系表。

薪资表

该表位于包含员工薪资和雇用日期的薪资数据库中。 该数据库是 Oracle 数据库。

SQL 转换

该转换可检索每个员工行的雇用日期和薪资。 该转换将连接到薪资数据库，并对该数据库中的薪资表运行 SQL 查询。

逻辑数据对象

包含员工和薪资数据的组合视图。 逻辑数据对象从 SQL 转换接收输出。

SQLErrors 文件

SQLErrors 文件是一个包含数据库中所有 SQL 错误的平面文件。 数据集成服务会在 SQLErrors 文件中为每个输入行至少写入一行。 如果没有出现 SQL 错误，则 SQLError 列将包含空值。 检查 SQLErrors 文件以排除错误。

薪资表

薪资表是薪资数据库中的关系表。 该表包含薪资部门维护的员工数据。 SQL 转换可从薪资表检索到雇用日期和员工薪资。

下表显示了薪资表中的部分行：

Employee_Num	HireDate	薪资
10	3-May-97	232000
11	11-Sep-01	444000
12	17-Oct-89	656000
13	13-Aug-07	332100

员工表

源表是人力资源数据库中的员工表。

下表显示了员工表中的示例行：

EmpID	LastName	FirstName	DeptId	Phone
10	Smith	Martha	FIN	(415) 552-1623
11	Jones	Cynthia	ENG	(415) 552-1744
12	Russell	Cissy	SLS	(415) 552-1656
13	Goyal	Girish	FIN	(415) 552-1656

SQL 转换

SQL 转换可从薪资数据库的薪资表中检索员工雇用日期和薪资。 薪资表位于 Oracle 数据库中。

执行以下步骤配置 SQL 转换：

1. 配置 SQL 转换属性。
2. 定义端口。
3. 创建 SQL 查询。
4. 为 SQL 转换配置数据库连接。

定义 SQL 转换属性

在**高级属性**视图中配置 SQL 转换属性。

配置以下属性：

数据库类型

数据库类型为 Oracle。 定义端口时，可以选择适用于 Oracle 的端口数据类型。

行内出现错误时继续

禁用。 如果在行中发生 SQL 错误，将停止处理。

包含统计信息作为输出

禁用。 不要创建 NumRowsAffected 输出端口。

定义端口

为员工源表中的每个列定义输入端口。 选择**复制到输出**为列将输入端口更改为传递端口。 选择**复制到输出**时，Developer 工具将为复制的每个端口创建相应的输出端口。

创建以下输入传递端口：

名称	类型	本地类型	精度	小数位数	复制到输出
EmpID	decimal	number(p,2)	4	0	x
LastName	string	varchar2	30	0	x
FirstName	string	varchar2	20	0	x
DeptID	string	varchar2	4	0	x
电话	string	varchar2	16	0	x

SQL 转换具有以下输出端口：

名称	类型	本地类型	精度	小数位数
EmpID	decimal	number(p,s)	4	0
LastName	string	varchar2	30	0
FirstName	string	varchar2	20	0

名称	类型	本地类型	精度	小数位数
DeptID	string	varchar2	4	0
电话	string	varchar2	16	0
HireDate	date/time	时间戳	29	0
薪资	decimal	number(p,s)	8	2

当选择**复制到输出**时，Developer 工具会将“_output”后缀添加到其创建的每个输出端口。

为雇用日期列和薪资列手动定义输出端口。 SQL 转换将在端口中返回薪资表中的雇用日期列和薪资列。

定义 SQL 查询

创建一条 SQL 查询以选择 Salary 表中每个员工的雇用日期和薪酬。

请在 SQL 转换的 SQL 视图中定义查询。

在 SQL 编辑器中输入以下查询：

```
select HIREDATE,SALARY,from Salary where EMPLOYEE_NUM =?EmpID?
```

Hiredate、Salary 和 Employee_Num 是 Salary 表中的列名称。

?EMPID? 是包含 EmpID 端口值的参数。

定义数据库连接

在**运行时**视图中，为 SQL 转换所连接的数据库选择数据库连接对象。 选择 Oracle 数据库连接对象。

输出

将 SQLError 端口和 EmpID_output 端口连接到 SQLErrors 平面文件。 除非出现 SQL 错误，否则 SQLError 端口将包含空值。

将 EmpID 和其他输出端口连接到逻辑数据对象。

SQL 转换将返回包含员工表中数据以及薪资表中雇用日期和薪资的行。

下表显示了逻辑数据对象中的部分行：

EmpID	LastName	FirstName	DeptId	电话	HireDate	薪资
10	Smith	Martha	FIN	(415) 552-1623	19970303 00:00:00	2320.00
11	Jones	Cynthia	ENG	(415) 552-1744	20010911 00:00:00	4440.00
12	Russell	Cissy	SLS	(415) 552-1656	19891017 00:00:00	6560.00
13	Goyal	Girish	FIN	(415) 552-1660	20070813 00:00:00	3210.00

存储过程

可以从 SQL 转换调用存储过程。可以使用存储过程在关系数据库中自动执行任务。存储过程接受用户定义的变量、条件语句和标准 SQL 语句所不支持的其他功能。

SQL 转换连接关系数据库以运行存储过程。SQL 转换可以从 Oracle、IBM DB2、Microsoft SQL Server、Sybase 和 ODBC 调用存储过程。存储过程保留在数据库中并在数据库中运行。

创建 ODBC 连接以便从 Sybase 数据库调用存储过程。要从非 Windows 操作系统中的 Microsoft SQL Server 数据库调用存储过程，也必须创建 ODBC 连接。

存储过程是预编译的 Transact-SQL、PL-SQL 或其他数据库过程语句的集合。存储过程的语法因数据库而异。

可以使用存储过程完成以下任务：

- 向目标数据库加载数据前检查该数据库的状态。
- 确定数据库空间是否充足。
- 执行专门的计算。
- 通过值检索数据。
- 删除然后重新创建索引。

可以使用存储过程执行要在转换中包含的查询或计算。例如，如果有一个用于计算销售税的经过测试的存储过程，可以使用该存储过程执行此计算，而不必在表达式转换中重新创建同样的计算。

存储过程可以接受输入，然后返回行的结果集。存储过程可以运行不需要输入因而也不返回输出的 DDL 任务。

可以将 SQL 转换配置为运行多个存储过程。对于配置的每个存储过程，将转换端口配置为与存储过程参数相匹配。每个存储过程都可以将数据传递回输出端口。

包含存储过程的数据库具有用户权限。必须具有权限才能在数据库中运行存储过程。

注意: 存储函数类似于存储过程，但函数只返回一个值。SQL 转换可以运行存储函数。

存储过程的 SQL 转换端口

SQL 转换输入和输出端口与存储过程中的输入和输出参数相对应。

导入存储过程时，Developer 工具将根据数据库连接确定数据库类型。 该工具将根据存储过程中的参数在 SQL 转换中生成输入和输出端口。 Developer 工具根据存储过程的参数确定每个端口的本地数据类型。

手动配置 SQL 转换时，需要在转换中配置输入和输出端口。 配置数据库类型时，SQL 转换将根据您输入的数据库类型更改每个端口的本地数据类型。

以下数据类型可以在 SQL 转换与存储过程之间传递：

输入和输出参数

SQL 转换将参数发送到存储过程，同时 SQL 转换将接收输入和输出端口中存储过程的参数。

返回值

如果存储过程传递返回值，Developer 工具将创建一个“返回值”端口。

SQL 错误

SQL 转换将在 SQLError 端口中从存储过程返回错误。

输入和输出参数

当从 SQL 转换调用存储过程时，调用语句引用的每个字段标识一个输入或输出端口。 导入存储过程时，Developer 工具会生成存储过程调用语句。 否则，您需要手动配置调用语句。

您可以在转换的 **SQL** 视图中编辑调用语句。

调用语句的格式如下：

```
?RETURN_VALUE? = call <stored proc name>(?Field1?, ?Field2?,. . . )
```

用问号将端口名称括起。 端口名称没有必要与存储过程中的参数名称匹配。 输出端口的顺序必须与 SELECT 查询中的参数顺序相同。

您可以使用包含 INOUT 参数的存储过程。 SQL 转换按输入端口名称标识 INOUT 参数。 输出端口的前缀为 output_。 数据集成服务将输入端口和输出端口绑定到同一参数。

您可以配置 SQL 转换返回结果集。 当存储过程返回结果集时，Developer 工具无法为结果集中的列创建输出端口。 导入存储过程时，必须手动输入端口并配置存储过程调用。

返回值

返回值是用于定义存储过程的状态的代码或文本字符串。 如果存储过程具有返回值，SQL 转换将具有**返回值**端口。

大多数数据库能够在运行存储过程后传递返回值。 返回值可以包含整数值，或者可以包含您在存储过程中定义的值。 例如，存储过程成功时，该过程可能会返回“Success”。

如果存储过程返回一个结果集，而非单个返回值，SQL 转换接收该过程的第一个返回值。

存储过程结果集

可以将 SQL 转换配置为从存储过程接收结果集。 存储过程在结果集中返回多个行。 SQL 转换可以将每个行返回到映射中。

以下存储过程返回一个结果集：

```
CREATE OR REPLACE FUNCTION fetchEMPinfo
(p_State  IN VARCHAR2 )
return types.cursortype
AS
my_cursor types.cursortype;
BEGIN
OPEN my_cursor FOR SELECT EMP_ID, NAME, CITY FROM EMP WHERE STATE = p_State ORDER BY EMP_ID;
RETURN my_cursor;
END;
```

导入存储过程时，Developer 工具会创建一个类似以下语法的存储过程调用语句：

```
call FETCHEMPINFO (?P_STATE?)
```

输入参数为 p_state。 Developer 工具不为您创建输出端口。 必须手动创建数据类型与存储过程参数相同的输出端口。

例如，resultSet 包含 EMP_ID、EMPNAME 和 CITY 列。为这些列创建输出端口。

还必须使用以下语法用输出列手动更新 SQL 调用：

```
(?EMP_ID?,?EMPNAME?,?CITY?) = call FETCHEMPINFO (?P_STATE?)
```

不同数据库的结果集

配置存储过程以根据数据库类型使用不同的语法返回结果集。

Oracle

Oracle 存储函数返回包含游标的结果：

```
create or replace function sp_ListEmp return types.cursortype
as
    l_cursor    types.cursorType;
begin
    open l_cursor for select ename, empno from emp order by ename;
    return l_cursor;
end;
```

Oracle 还将游标接受为输入参数。不能通过 SQL 转换将游标配置为输入参数。

Microsoft SQL Server

Microsoft SQL Server 存储过程返回在过程体中包含 Select 语句或者将返回类型显式声明为表的结果集存储过程。

```
Create  PROCEDURE InOut(
@inout varchar(100) OUT
)
AS
BEGIN
set @inout = concat(@inout,'__')
select * from mytable;
END
```

IBM DB2

IBM DB2 存储过程可以返回包含打开游标的结果集。其返回的结果集数量在 RESULT SET 子句中声明。存储过程打开游标并将其返回。以下示例返回 2 个打开游标。

```
CREATE PROCEDURE TESTMULTIRS
 (IN i_cmacct CHARACTER(5))
RESULT SETS 2
LANGUAGE SQL
BEGIN

DECLARE csnum INTEGER;

--Declare serial cursors to consume less resources
--You do not need a rollable cursor.

DECLARE getDeptNo CHAR(50); --Be careful with the estimated length.
DECLARE getDeptName CHAR(200);
DECLARE c1 CURSOR WITH RETURN FOR s1;
SET getDeptNo = 'SELECT DEPTNO FROM DEPT';
SET getDeptName = 'SELECT DEPTNAME FROM DEPT';

PREPARE s1 FROM getDeptNo;
OPEN c1;

END;
```

Sybase

Sybase 存储过程返回在过程体中包含 Select 语句或者将返回类型显式声明为表的结果集存储过程。

```
CREATE PROCEDURE FETCHEMPINFO
(
@p_State VARCHAR(5),
@e_id INT OUTPUT,
@e_name VARCHAR(50) OUTPUT
```

```
)
AS
BEGIN
SET NOCOUNT ON
SELECT EMP_ID, NAME FROM EMP WHERE STATE = @p_State ORDER BY EMP_ID
SET NOCOUNT OFF
SELECT @e_id AS EMP_ID, @e_name AS NAME
RETURN
END
GO

--Configure the following variables to execute the procedure.

DECLARE @p_State VARCHAR(5)
DECLARE @EMPID int
DECLARE @EMPNAME varchar(50)

SET @p_State = 'CA'
exec FETCHEMPINFO @p_State, @e_id = @EMPID, @e_name = @EMPNAME
GO
```

结果集行

除了结果集行，某些存储过程还返回输出参数。 SQL 转换在最后一行返回输出参数。 它未在结果集行中包括单次出现的输出参数。

例如，您编写这样一个存储过程：接收员工 ID 并在输出参数 1 中返回员工姓名，在输出参数 2 中返回部门。 存储过程还为员工一年中所休的每个病假日返回一行。 该行包括日期、小时数以及缺勤原因。

每位员工的结果集包含的行数不同。 结果集中的每行包含空的员工姓名和部门。 SQL 转换在结果集之后返回员工姓名和部门。 员工姓名和部门出现在最后一行。

存储过程示例

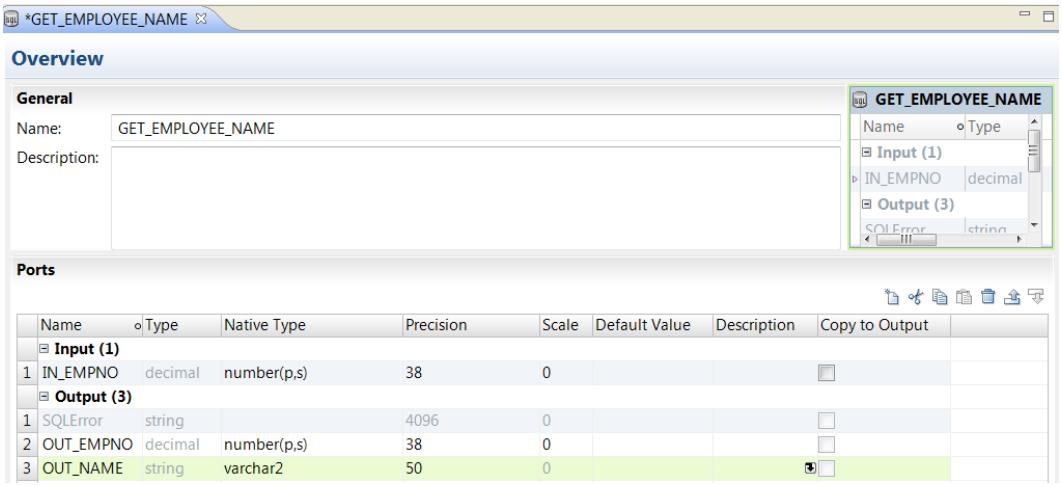
可以调用一个将数据返回到 SQL 转换时使用的存储过程。

以下存储过程将接收员工号，并返回一个包含员工号和员工姓名的行：

```
CREATE OR REPLACE PROCEDURE SP_GETNAME
(IN_EMPNO IN NUMBER, OUT_EMPNO NUMBER, OUT_NAME OUT STRING)
AS
BEGIN
SELECT EMP_KEY,EMP_NAME into OUT_EMPNO , OUT_NAME from EMP_TABLE where EMP_KEY=IN_EMPNO;
END;/"
```

要创建 SQL 转换，请导入存储过程。 Developer 工具将创建输入端口和输出端口。 端口名称与存储过程中的参数名称相同。

下图显示了 SQL 转换的端口：



Developer 工具将创建以下存储过程调用以检索员工号：

```
call SP_GETNAME (?IN_EMPNO?,?OUT_EMPNO?,?OUT_NAME?)
```

可以在 SQL 编辑器中查看存储过程调用。 所有 SQL 错误都在 SQLError 报告中显示。

SQL 转换连接

可以在转换运行时属性中配置 SQL 转换连接。如果在创建转换时未指定连接，则您可能需要配置运行时连接。

您可以定义与选择用于创建 SQL 转换的连接不同的连接名称。您必须选择与 SQL 转换**高级**属性中的数据库类型相同的数据库类型连接。

可为 SQL 转换连接名称配置一个参数。您必须先在映射的**参数**视图中定义参数，然后才能将该参数分配给运行时连接。

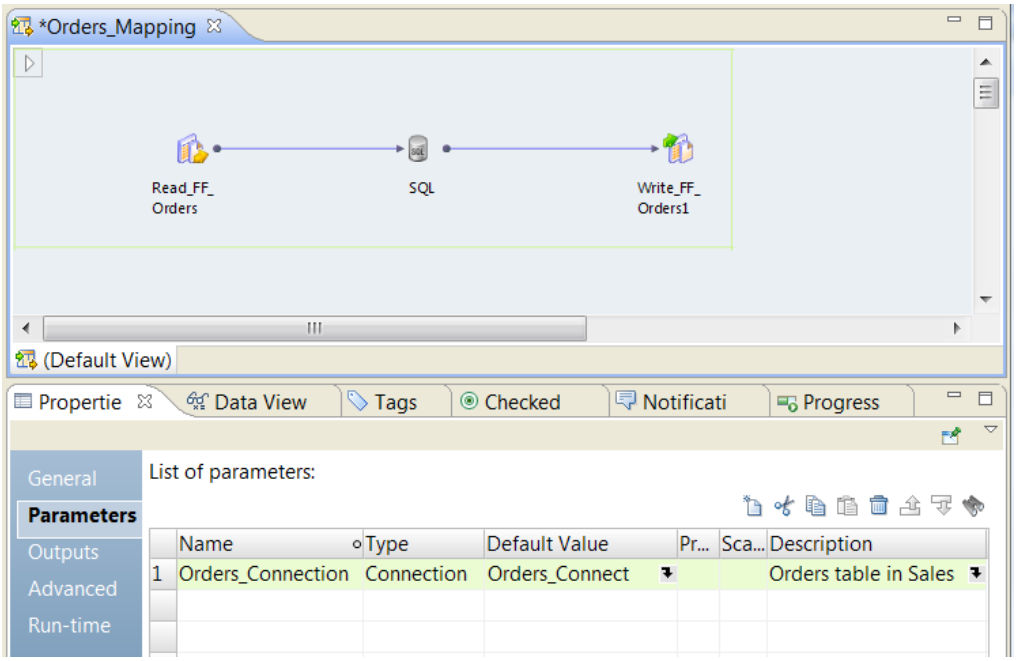
创建连接名称参数

可以在 SQL 转换的运行时连接名称中指定用户定义的参数。 Developer tool 将为该连接创建映射参数而非转换参数。

1. 创建包含 SQL 转换的映射。
2. 单击 SQL 转换的**运行时**选项卡。
3. 在**连接名称**中，单击选择箭头并选择**分配参数**。
4. 在**分配参数**对话框中，单击**新建**。
5. 在**参数**对话框中，输入连接参数名称和参数说明。 参数类型默认为连接。
6. 依次单击“参数”对话框和**分配参数**对话框中的**确定**。

   Developer tool 即会创建映射参数并将其分配给“连接名称”。 参数名称显示在“运行时”属性中。

7. 要查看映射参数的列表，请在编辑器内单击，然后单击映射的**参数**选项卡。



手动创建 SQL 转换

您可以手动创建 SQL 转换。 配置运行 SQL 查询的转换时，请手动创建该转换。 存储过程对导入不可用时，您也可能会手动创建一个调用该过程的转换。 手动创建转换时，请配置输入和输出端口，然后在 SQL 编辑器中键入 SQL 语句。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择 SQL 转换。
4. 单击**下一步**。
5. 选择**创建为空**。
6. 输入转换的名称，然后输入转换的存储库位置。
7. 单击**完成**。
8. 单击**概览**视图，将端口添加到转换中。
9. 要添加输入端口，请单击**端口**面板中的**输入**，以指示要添加端口的位置。 单击**新建**按钮，然后输入端口名称、本地类型和精度。

   默认数据库类型为 Oracle。除非您在**高级**视图中更改了数据库类型，否则 Developer 工具将显示 Oracle 数据库的本地类型。
10. 要添加输出端口，请先单击**端口**面板中的**输出**，然后再添加端口。 单击**新建**按钮，然后输入端口名称、本地类型和精度。

    默认情况下，**SQLError** 端口是第一个输出端口。

11. 在**高级**视图中，选择 SQL 转换连接到的数据库类型。 配置用于错误处理的其他高级属性以及其他可选属性。

    选择数据库类型时，Developer 工具将在**概览**视图中更改端口的本地数据类型。

12. 在 **SQL** 视图中键入 SQL 查询或存储过程调用。 在 **SQL 编辑器**中为参数绑定或字符串置换选择端口。

    如果存储过程返回一个结果集，则必须使用类似于以下语法的语法输入存储过程调用：(?Field1?,?Field2?,?Field3?) = **call** Stored_Procedure_Name (?Input_Parm?)。

相关主题：

- “定义 SQL 查询” 页面上 551

通过存储过程创建 SQL 转换

可以通过从数据库连接导入存储过程来配置 SQL 转换。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择 SQL 转换。
4. 单击**下一步**。
5. 选择**通过现有存储过程创建**。
6. 浏览并选择数据库连接。
7. 浏览并选择要导入的存储过程。
8. 输入转换的名称和位置。
9. 单击**完成**。

   开发程序工具将创建端口和存储过程调用。
10. 如果存储过程返回一个结果集，必须手动添加输出端口，然后重新配置存储过程调用。
    a. 在**概览**视图中，单击**端口**面板中的**输出**。 单击**新建**按钮并输入输出端口名称、本地类型和精度。
    b. 在 **SQL** 视图中，更改存储过程调用以使用以下语法：(?Field1?,?Field2?,?Field3?) = **call** Stored_Procedure_Name (?Input_Parm?)

       可以在 SQL 编辑器中从端口的**参数绑定**列表选择输入和输出参数。

第 44 章

标准创建器转换

本章包括以下主题：

标准创建器转换概览

标准创建器转换是一种被动转换，用于检查输入字符串并创建这些字符串的标准化版本。

标准创建器转换创建包含输入字符串标准化版本的列。 创建这些列时，该转换可以替换或删除输入数据中的字符串。

例如，您可以使用标准创建器转换来检查包含 Street、St. 和 STR 字符串的地址数据列。 您可以使用字符串 St 替换这些字符串的所有实例。

在标准创建器转换中，您可以创建多个标准化策略。 每个策略可以包含多个标准化操作。 标准创建器转换提供一个向导，用于创建策略。

标准化策略

使用标准化策略，通过标准化版本的输入字符串创建列。

配置标准化策略时，需要添加一个或多个操作。 每个操作均实现一个特定的标准化任务。

可向标准化策略添加以下类型的操作：

将引用表匹配项替换为有效值

将与引用表值匹配的字符串替换为引用表中的“有效”值。

将引用表匹配项替换为自定义字符串

将与引用表值匹配的字符串替换为用户定义的替换字符串。

删除引用表匹配项

删除与引用表值匹配的字符串。

替换自定义字符串

将用户定义的字符串替换为用户定义的替换字符串。

删除自定义字符串

删除用户定义的字符串。

重要说明: 可以更改操作的顺序。 操作的顺序会更改策略的输出，因为每个操作读取的是前一次操作的结果。

标准化属性

要配置标准化策略和操作的属性，请在标准创建器转换中选择**策略**视图。

策略属性

策略属性适用于策略中的所有操作。 您可以配置以下策略属性：

删除多个空格

用一个空格替换多个连续空格。

删除尾随空格和前导空格

删除数据字符串开头和末尾的空格。

分隔符

确定用于定义搜索标志的分隔符。 例如，如果选择“分号”，则标准创建器转换会搜索字符串“oranges;apples;”，然后找到字符串“oranges”和“apples”。 默认情况下，该转换使用空格作为分隔符。

操作属性

可以配置以下类型的标准化操作属性。

引用表操作

引用表操作包含以下属性：

- **引用表**。 确定用于标准化数据的引用表。 单击**浏览**以选择引用表。
- **区分大小写**。 确定输入字符串是否必须与引用表条目的大小写相匹配。
- **替换为**。 使用提供的文本替换与引用表条目匹配的输入字符串。 仅适用于替换操作。
- **作用域**。 指定包含引用表值的部分输入字符串。

自定义字符串操作

自定义字符串操作包含以下属性：

- **标志匹配项**。 定义要在输入数据中查找的搜索字符串。
- **替换为**。 替换与指定的搜索字符串匹配的输入字符串。 仅适用于替换操作。
- **作用域**。 指定要搜索的部分输入字符串。

配置标准化策略

要配置标准化策略，请在标准创建器转换的**策略**视图中编辑相应设置。

1. 选择**策略**视图。
2. 单击**新建**。

   此时将打开 **新建策略** 向导。
3. 单击**输入**字段为策略选择端口。
4. 配置策略属性，然后单击**下一步**。
5. 选择操作，然后单击**下一步**。
6. 配置操作属性。
7. 或者，单击**下一步**向策略中添加更多操作。
8. 将所有操作添加到策略后，单击**完成**。
9. 或者，向转换中添加更多策略。
10. 或者，更改转换处理策略或操作的顺序。 选择某个策略或操作，然后单击**上移**或**下移**。

标准创建器转换高级属性

配置属性以帮助确定数据集成服务如何处理标准创建器转换的数据。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境中的标准创建器转换

非本地环境中的标准创建器转换处理取决于运行转换的引擎。

请注意是否支持以下非本地运行时引擎：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射和流映射中无限支持。
- Databricks Spark 引擎。无限支持。

第 45 章

联合转换

本章包括以下主题：

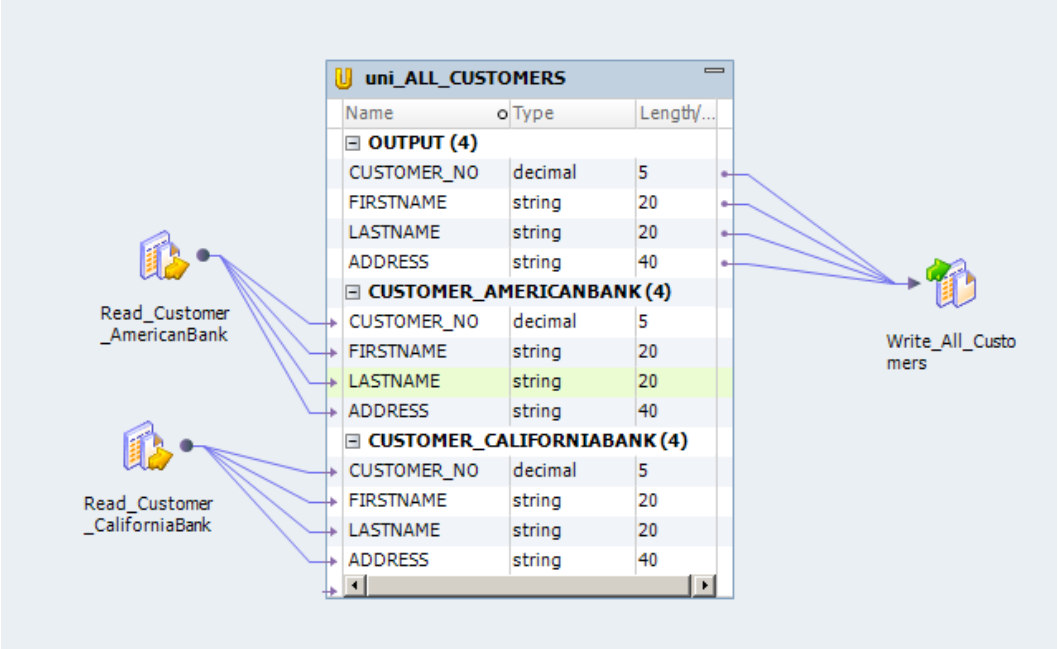
联合转换概览

使用联合转换可以将多个管道或多个管道分支中的数据合并到一个管道分支。

联合转换是一种主动转换，其中包含多个输入组和一个输出组。它将源与匹配的端口合并，并将数据传递给与输入组具有相同端口结构的输出组。在 Developer 工具中使用联合转换可以合并多个源中的数据，而无需删除重复行。

例如，如果要将美国银行和加利福尼亚银行的客户帐户数据合并，则可以创建包含联合转换的映射来合并源对象的数据，并将其写入目标对象。

下图显示了包含联合转换的映射：



组和端口

一个联合转换具有多个输入组和一个输出组。 您可以创建一个或多个输入组。 您可以使用 Developer 工具创建一个输出组。 您不能创建、编辑或删除输出组。 每个组必须具有匹配的端口。

要创建端口，可以从转换中复制端口，也可以手动创建端口。 创建端口时，Developer 工具会在每个输入组中创建输入端口，在输出组中创建输出端口。 Developer 工具使用您为每个输入端口和输出端口指定的输出端口名。 Developer 工具还为每个端口使用相同的元数据（例如数据类型、精度和小数位数）。

您可以从一个管道中的不同分支或不同源管道连接输入组。 向某个映射添加联合转换时，必须确认在所有输入组中连接了相同的端口。 如果在一个输入组中连接了一个端口，但在另一个输入组中未连接同一端口，则数据集成服务会向未连接的端口传递空值。

联合转换高级属性

配置属性以帮助确定数据集成服务如何显示联合转换的日志详细信息。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

联合转换处理

使用联合转换可以将多个管道或多个管道分支中的数据合并到一个管道分支。 数据集成服务将并行处理所有输入组。 同时，它还将读取连接到联合转换的源，并将数据块传递给联合转换的输入组。 联合转换会基于从数据集成服务接收数据块的顺序处理这些数据块。 联合转换不会阻止输入组上的输入数据。

创建联合转换

可以创建可重用或不可重用联合转换。

创建可重用联合转换

可创建可重用联合转换以供在多个映射或 Mapplet 中使用。

1. 在**对象浏览器**视图中选择一个项目或文件夹。
2. 单击**文件** > **新建** > **转换**。

   此时将显示**新建**对话框。
3. 选择联合转换。
4. 单击**下一步**。
5. 输入转换的名称。
6. 单击**完成**。

   此时，转换将显示在编辑器中。
7. 单击**新建**按钮向转换添加端口。
8. 编辑端口以设置名称、数据类型和精度。
9. 选择**组**视图。
10. 单击**新建**按钮以添加输入组。
11. 单击**高级**视图并编辑转换属性。

创建不可重用联合转换

可在映射或 Mapplet 中创建不可重用联合转换。

1. 在映射或 Mapplet 中，将“转换”选项板中的联合转换拖动到编辑器中。

   此时，转换将显示在编辑器中。
2. 在**常规**选项卡中，编辑转换名称和说明。
3. 从上游转换中选择所有端口，然后将其拖动到联合转换。这些端口将显示为联合转换的输入组和输出组中的端口。
4. 在**属性**视图中，单击**组**选项卡上的**新建**以添加输入组。

   此时将显示具有与现有输入组相似端口的另一个输入组。
5. 选择联合转换的输出组中的端口，然后将其拖动到映射中的下游转换。

非本地环境下的联合转换

非本地环境中的联合转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中无限支持。在流映射中有限支持。
- Databricks Spark 引擎。无限支持。

流映射中的联合转换

流映射具有映射验证限制。

映射验证

在下列情况下，映射验证会失败：

- 转换配置为合并流管道和非流管道的数据。

Databricks Spark 引擎上的联合转换

第 46 章

更新策略转换

本章包括以下主题：

更新策略转换概览

更新策略转换是一种主动转换，用于将某行标记为插入、更新、删除或拒绝。使用更新策略转换可基于您应用的条件控制对目标中现有行的更改。

作为一种主动转换，更新策略转换可能会更改传递给它的行的数量。更新策略转换会测试每个行，以检查其是否满足特定条件，然后相应地标记这些行。该转换将标记为要插入、更新或删除的行传递到下一个转换。可以将该转换配置为将标记为要拒绝的行传递到下一个转换，或删除标记为要拒绝的行。

例如，当邮寄地址更改时，可以使用更新策略转换将所有客户行标记为要进行更新。或者，可以将不再就职于组织的人员的所有员工行标记为要拒绝。

映射在 Spark 引擎上运行时，可以使用“更新策略”转换将结果写入关系数据库目标。映射使用 JDBC 连接字符串。

设置更新策略

要定义更新策略，请完成以下步骤：

1. 要控制映射中的行标记为插入、更新、删除或拒绝的方式，请向该映射添加更新策略转换。使用更新策略转换可为针对同一目标的行标记不同的数据库操作，或拒绝相应行。
2. 配置映射时为各个目标定义插入、更新和删除选项。按目标配置可以允许或禁止对标记为要插入或删除的所有行执行插入和删除操作。可以选择不同的方式处理对标记为要更新的行执行的更新操作。
3. 如果在 Spark 运行时引擎上运行映射，可以在“更新策略”转换高级属性中选择“使用 Hive 合并”。当查询使用 MERGE 语句而非 INSERT、UPDATE 或 DELETE 语句时，处理通常更高效。

在动态映射中更新策略转换

可以在动态映射中使用更新策略转换。 可以在转换中配置动态端口，并引用生成的端口。

可以在更新策略转换中引用动态端口或生成的端口。 但是，如果生成的端口在运行时不存在，则映射将会失败。

如果在更新策略表达式中使用某个动态端口，则该动态端口最多只能有一个生成的端口。

您可以参数化更新策略表达式。 使用表达式类型参数并输入整个表达式作为默认参数值。

在映射中标记行

将更新策略转换添加到映射中以将行标记为插入、更新、删除或拒绝。

定义更新策略表达式，以测试每一行，确保其符合特定条件。然后，为每一行分配一个数字代码，以将行标记为特定数据库操作。

下表列出了用于每个数据库操作的常量及其等效数值：

操作	常量	数值
插入	DD_INSERT	0
更新	DD_UPDATE	1
删除	DD_DELETE	2
拒绝	DD_REJECT	3

数据集成服务会将任何其他值视为插入。

更新策略表达式

在表达式编辑器中输入更新策略表达式。

更新策略表达式使用转换语言中的 IIF 或 DECODE 函数来测试每一行。例如，如果条目日期在应用日期之后，以下 IIF 语句将设置一行的拒绝标志。否则，该语句将设置该行的更新标志：`IIF( ( ENTRY_DATE > APPLY_DATE), DD_REJECT, DD_UPDATE)`

可在更新策略表达式中配置参数。在表达式编辑器中创建参数或浏览参数。

如果转换采用动态映射，则转换中生成的字段可能会发生变化。您可以参数化整个更新策略表达式。如果使用参数来定义表达式，则 Developer tool 无法验证表达式。表达式参数不能包含其他参数。

更新策略转换高级属性

配置高级属性以帮助确定数据集成服务如何处理更新策略转换的数据。

可以在“高级”选项卡中为更新策略转换定义以下高级属性：

转发拒绝的行

确定更新策略转换是将拒绝的行传递给下一个转换，还是将其删除。默认情况下，数据集成服务会将拒绝的行转发到下一个转换。数据集成服务会标记要拒绝的行，并将其写入拒绝文件。如果未选择“转发拒绝的行”，则数据集成服务将删除拒绝的行，并将其写入映射日志文件。

使用 Hive 合并

确定映射在 Spark 引擎上运行时更新策略转换是否使用 Hive MERGE 在 Hive 目标上执行更新。查询使用 MERGE 语句，而不是 INSERT、UPDATE 或 DELETE 语句时，处理更有效率。

在以下情况下，映射会忽略 Hive MERGE 选项：

- 映射在 Databricks Spark 引擎上运行。
- 映射在 Blaze 引擎上运行。
- 在特定 Hadoop 发行版上 Hive 实现限制使用 MERGE。

在 Databricks Spark 引擎上运行的映射使用 Databricks“MERGE INTO”SQL 命令，而不是 Hive MERGE。在 Blaze 引擎上运行的映射中和限制使用 MERGE 的情况下，运行时引擎使用 INSERT、UPDATE 和 DELETE 来执行操作。

映射日志包含操作的结果，其中包括限制是否影响结果。

更新影响对列进行分区或分桶时，会省略列的更新。

注意: Developer tool 和数据集成服务不验证此限制。如果更新策略表达式违反了这些限制，则映射可能会产生意外结果。

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

汇总器和更新策略转换

将汇总器和更新策略转换连接起来作为同一管道的一部分时，请将汇总器置于更新策略转换之前。 数据集成服务将按照此顺序执行汇总计算，然后将包含此计算结果的行标记为插入、更新、删除或拒绝。

如果将更新策略置于汇总器转换之前，则必须考虑汇总器转换如何处理标记为不同操作的行。 数据集成服务将按照此顺序将行标记为插入、更新、删除或拒绝，然后再执行汇总计算。 行的标记方式将决定汇总器转换如何处理该行中用于计算的值。 例如，如果将某行标记为删除，然后使用该行计算总和，则数据集成服务将减去此行中的值。 如果将某行标记为拒绝，然后使用该行计算总和，则数据集成服务不会包括此行中的值。 如果将某行标记为插入或更新，然后使用该行计算总和，则数据集成服务会将此行中的值添加到总和中。

为各个目标指定更新选项

使用更新策略转换为每个行标记特定的数据库操作后，为映射中的每个目标定义插入、更新和删除选项。可以禁止对标记为要插入或删除的行执行插入或删除操作。可以选择不同的方式处理对标记为要更新的行执行的更新操作。

在映射中的目标数据对象的高级属性中定义更新策略选项。可以设置以下更新策略选项：

插入

将标记为要插入的所有行插入到目标。默认情况下启用该属性。

删除

将标记为要删除的所有行从目标中删除。默认情况下启用该属性。

更新策略

更新现有行的策略。选择以下策略之一：

- **更新为更新。**更新标记为要更新的所有行。这是默认值。
- **更新为插入。**插入标记为要更新的所有行。
- **更新否则插入。**对于标记为要更新的所有行，如果这些行在目标中存在，则更新这些行，然后插入标记为插入的剩余行。

截断表

在加载数据之前，截断目标。默认为“已禁用”。

非本地环境下的更新策略转换

非本地环境中的更新策略转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。有限支持。
- Spark 引擎。在批处理映射中有限支持。在流映射中不受支持。
- Databricks Spark 引擎。有限支持。

Blaze 引擎上的更新策略转换

Blaze 引擎的某些处理规则与数据集成服务的处理规则不同。

一般限制

可以在支持 Hive ACID 的 Hadoop 发行版上使用“更新策略”转换。

如果更新策略转换收到同一主键值的多个更新行，则转换会随机选择某一行来更新目标。

如果多个更新策略转换写入同一目标的不同实例，则目标数据可能无法预测。

Blaze 引擎按以下顺序执行操作：删除、更新、插入。不按更新策略转换收到行的顺序来处理这些行。Hive 目标始终执行“更新为更新”操作。Hive 目标不支持“更新 Else 插入”或“更新为插入”。

映射验证和编译验证

在下列情况下，映射验证会失败：

- 更新策略转换连接到多个目标。
- 更新策略转换不直接位于目标之前。
- 更新策略目标不是 Hive 目标。
- 更新策略转换目标是外部 ACID 表。
- 目标不包含主键。
- 启用了在运行时截断目标表的 Hive 目标属性。
- 启用了在运行时创建或替换目标表的 Hive 目标属性。

在下列情况下，映射会失败：

- 目标未进行 ORC 分桶。
- 将 Hive 目标修改为行数少于实际表。

以下情况下会出现编译验证错误，并停止执行映射：

- 没有为事务启用目标表。
- Hive 版本早于 0.14。

使用 Hive 目标表

要将 Hive 目标表与更新策略转换配合使用，必须使用 Hive 数据定义语言的以下子句：TBLPROPERTIES ("transactional"="true") 创建 Hive 目标表。

要将更新策略转换与 Hive 目标配合使用，请确认在与 Hadoop 连接关联的 hive-site.xml 配置集中是否配置了以下属性：

```
hive.support.concurrency         true
hive.enforce.bucketing           true
hive.exec.dynamic.partition.mode nonstrict
hive.txn.manager         org.apache.hadoop.hive.ql.lockmgr.DbTxnManager
hive.compactor.initiator.on      true
hive.compactor.worker.threads    1
```

Spark 引擎上的更新策略转换

Spark 引擎的某些处理规则与数据集成服务的处理规则不同。

Hive 目标的一般限制

可以在支持 Hive ACID 的 Hadoop 发行版上使用“更新策略”转换。

还可以使用更新策略转换将映射结果写入与 JDBC 兼容的关系目标。

目标是 Hive 表或兼容 JDBC 的表时，更新策略转换不会将被拒绝的行转发给下一个转换。

如果更新策略转换收到同一主键值的多个更新行，则转换会随机选择某一行来更新目标。

如果多个更新策略转换写入同一目标的不同实例，则目标数据可能无法预测。

如果映射在 Spark 引擎上运行，则可以选择“使用 Hive 合并”选项。该选项具有以下限制：

- 用于删除或更新的单个行不能与目标中的多个行匹配。映射违反此限制时，映射将失败并显示运行时错误。
- 如果配置更新策略表达式来更新分区列或分桶列，则映射会忽略“Hive 合并”选项，并且不会更新列。

注意: Developer tool 和数据集成服务不会验证这些限制。如果表达式或映射违反这些限制，映射可能会运行，但不会按预期提供结果。

Hive 目标始终执行“更新为更新”操作。Hive 目标不支持“更新 Else 插入”或“更新为插入”。

在更新策略表达式中使用层次结构数据

可将表达式配置为在复杂数据字段中使用原始类型。

例如，您有一个由以下语句定义的结构数据类型嵌套数组：

```
addresses array<struct<city:varchar(15), apartment:smallint, pincode:int>>
```

您可以在更新策略表达式中使用城市列，因为其为使用以下表达式的原始类型：

```
IIF( (addresses[0].city = ' NEW YORK' ), DD_INSERT, DD_UPDATE)
```

映射验证

更新策略转换输出字段必须与目标输入字段完全匹配。

在下列情况下，映射验证会失败。

- 更新策略转换连接到多个目标。
- 更新策略转换不直接位于目标之前。
- 更新策略转换目标是外部 ACID 表。
- 目标不包含连接的主键。
- 选择了运行时启用截断目标表的属性。
- 运行时为目标表选择了以下目标策略之一：
  - 创建或替换目标表
  - 应用新列
  - 应用新架构
  - 失败

目标是 Hive 目标时，在下列情况下映射会失败：

- 没有为事务启用目标表。
- 目标未进行 ORC 分桶。

使用 Hive 目标表

要将 Hive 目标表与更新策略转换配合使用，必须使用 Hive 数据定义语言的以下子句：`TBLPROPERTIES ("transactional"="true")` 创建 Hive 目标表。

要将更新策略转换与 Hive 目标配合使用，请确认在与 Hadoop 连接关联的 hive-site.xml 配置集中是否配置了以下属性：

```
hive.support.concurrency         true
hive.enforce.bucketing           true
hive.exec.dynamic.partition.mode nonstrict
hive.txn.manager         org.apache.hadoop.hive.ql.lockmgr.DbTxnManager
hive.compactor.initiator.on      true
hive.compactor.worker.threads    1
```

Databricks Spark 引擎上的更新策略转换

您可以使用更新策略转换来执行插入、更新和其他操作。还可以在选择的 Delta Lake 列中执行插入和更新操作。

一般限制

当使用 Delta Lake 表目标时，只能在 Databricks Spark 引擎上使用更新策略转换。

更新策略转换仅在支持 Delta ACID 的 Databricks 发行版上受支持。

映射验证

在下列情况下，映射验证会失败。

- 更新策略转换连接到多个目标。
- 更新策略转换不直接位于目标之前。
- 更新策略转换目标是外部 ACID 表。
- 目标不包含连接的主键。
- 选择了运行时启用截断目标表的属性。
- 运行时为目标表选择了以下目标策略之一：
  - 创建或替换目标表

- 应用新列
- 应用新架构
- 失败

第 47 章

Web 服务使用者转换

本章包括以下主题：

Web 服务使用者转换概览

Web 服务使用者转换作为 Web 服务客户端连接到 Web 服务以访问或转换数据。Web 服务使用者转换属于多组转换。

Web 服务使用开放标准，例如 SOAP、WSDL 和 XML。 SOAP 是 Web 服务所使用的通信协议。 Web 服务客户端请求和 Web 服务响应都属于 SOAP 消息。 WSDL 是一种 XML 架构，用于描述 Web 服务操作的协议、格式和签名。

Web 服务操作包括对信息的请求、更新数据的请求或执行任务的请求。 例如，Web 服务使用者转换发送一个 SOAP 请求，要求运行名为 getCustomerOrders 的 Web 服务操作。 该转换会在请求中传递客户 ID。 Web 服务将检索客户信息和订单信息。 Web 服务会在 SOAP 响应中将信息返回给该转换。

Web 服务使用者转换使用在 WSDL、Web 服务连接或端点 URL 输入端口中定义的端点 URL 连接到 Web 服务。可以在 Web 服务连接中为 Web 服务启用安全性。

SOAP 消息

Web 服务使用者转换使用简单对象访问协议 (SOAP) 与 Web 服务提供程序交换信息并请求 Web 服务。SOAP 可定义 Web 服务请求和响应消息的格式。

使用 Web 服务使用者转换转换数据时，该转换将生成 SOAP 请求并连接到 Web 服务。该转换使用在 WSDL 对象、Web 服务连接或端点 URL 输入端口中定义的端点 URL 连接到 Web 服务。SOAP 请求包含 Web 服务运行请求操作所需的信息。Web 服务操作会在 SOAP 响应中将数据返回给转换。该转换会映射 SOAP 响应中的数据，然后在输出端口中返回数据。

Web 服务使用者转换对 ISO-8859-1 中的 SOAP 消息头进行编码。

该转换可以使用文档/文字编码处理 SOAP 消息。文档/文字样式需要一个 XML 架构来描述 SOAP 消息。SOAP 消息由 XML 构成。如果 SOAP 消息包含多次出现的元素，则元素组将形成 XML 层次结构中的级别。如果一个级别嵌套在另一个级别内，则这些组都相互关联。

SOAP 请求消息可以包含层次结构数据。例如，如果 Web 服务使用者转换向销售数据库发送一个添加客户订单的请求， 则该转换会在 SOAP 请求消息中传递两组数据。一组包含客户 ID 和名称，另一组包含订单信息。订单信息将出现多次。

SOAP 响应消息可以包含层次结构数据。例如，如果 Web 服务使用者转换生成一条关于客户订单的 SOAP 请求，则 Web 服务将在 SOAP 响应中返回订单表头和多次出现的订单详细信息元素。

WSDL 文件

WSDL 文件包含对要传递给 Web 服务的数据的说明，以便发件人和收件人了解要交换的数据。 必须将 WSDL 文件导入存储库，才能创建 Web 服务使用者转换。

WSDL 描述了对数据执行的操作以及对协议或传输的绑定，以便 Web 服务使用者可以按照正确的格式发送请求消息。 WSDL 描述了连接到 Web 服务的网络地址。

WSDL 包含有关如何对 SOAP 请求消息和响应消息进行编码的信息。 SOAP 编码方式决定 SOAP 消息正文的格式。 它描述了 Web 服务用于和 Web 服务使用者进行通信的请求消息和响应消息的格式。 Web 服务开发人员可以使用各种工具包创建 Web 服务。 工具包支持不同方式的 SOAP 编码消息。

Web 服务使用者转换支持文档/文字 SOAP 编码样式。 可以对 Web 服务使用者转换使用 WSDL 1.1。 不能使用 WSDL 附件，例如 MIME、DIME 和 MTOM 消息。

操作

对于自身支持的每个行为，Web 服务都包含一项操作。

例如，Web 服务可以包含名为 getcustomerid 的操作，用于接收客户名称并使用客户详细信息进行响应。 操作输入包含客户名称的元素。 操作输出包含基于客户名称的客户详细信息的元素。

配置 Web 服务使用者转换时，请定义转换如何将数据映射到操作输入，以及转换如何从操作输出映射数据。 在转换中配置以下信息：

输入映射

定义如何将转换输入端口映射到 Web 服务操作输入节点。 操作输入为操作定义了 SOAP 请求中的元素。

输出映射

定义如何将 Web 服务操作输出节点映射到转换输出端口。 操作输出为操作定义了 SOAP 响应中的元素。

Web 服务安全

可以在 Web 服务连接中为 Web 服务启用安全性。 可以配置以下安全类型：

Web 服务安全

数据集成服务向 Web 服务提供程序发送 SOAP 请求时，可以包含 Web 服务安全表头。 Web 服务安全表头包含身份验证信息，因此 Web 服务提供程序可以对数据集成服务进行身份验证。

Web 服务使用者转换提供用户名标志。 数据集成服务可在 SOAP 请求中创建单独的安全 SOAP 表头，然后将该请求传递给 Web 服务提供程序。

可以在 Web 服务连接中使用以下类型的 Web 服务安全：

- PasswordText。数据集成服务不更改 WS-Security SOAP 表头中的密码。

- PasswordDigest。数据集成服务将密码与临时值和时间戳结合。 数据集成服务对密码应用 SHA 哈希，采用 base64 编码方式进行编码，并在 SOAP 表头中使用编码后的密码。

传输层安全

使用安全套接字层 (SSL) 实现 TCP/IP 的传输层（TCP 层）上的安全性。 Web 服务将 SSL 上的超文本传输协议 (HTTPS) 用作安全消息传输的 Web 地址。 Web 服务使用者转换可以使用 TLS 1.2、TLS 1.1 或 TLS 1.0。可以将以下身份验证与传输层安全结合使用：HTTP 身份验证、代理服务器身份验证和 SSL 证书。

SSL 身份验证

通过 HTTPS 协议进行连接时，可以使用 SSL 身份验证。

可以使用以下类型的 SSL 身份验证：

- 单向 SSL 身份验证
- 双向 SSL 身份验证

HTTP 身份验证

通过 HTTP 协议进行连接时，可以使用 HTTP 身份验证。

可以使用以下 HTTP 身份验证方法：

- 基本身份验证
- 摘要身份验证
- NT LAN Manager (NTLM) 身份验证

WSDL 选择

创建 Web 服务使用者转换之前，必须将 WSDL 文件导入模型存储库中。只能导入使用 document/literal 的 SOAP 绑定样式定义的 WSDL 文件。

WSDL 定义了要运行的 Web 服务的操作签名。导入 WSDL 时，Developer tool 将创建一个物理数据对象，您可以将其重用于其他转换。

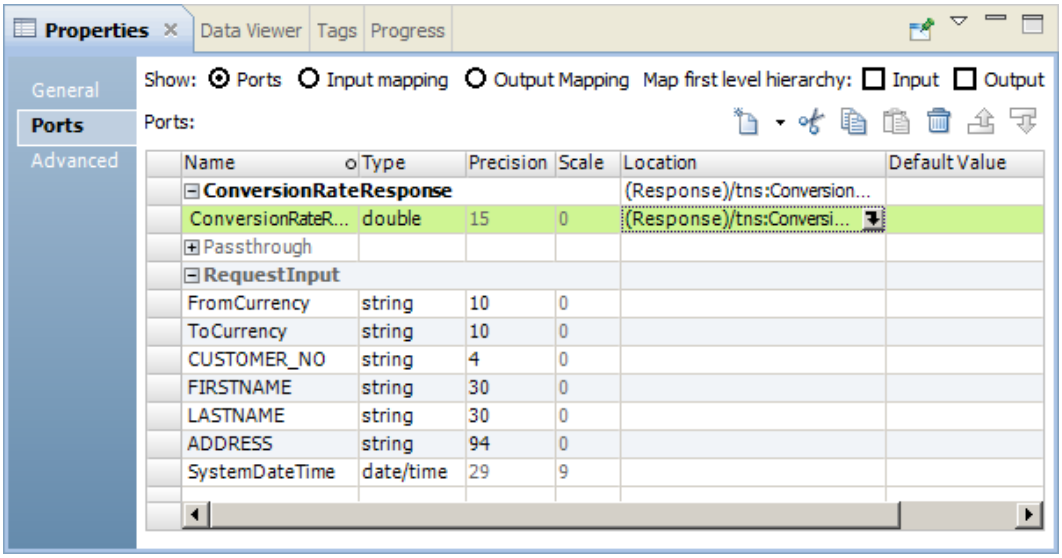
一个 WSDL 可以定义多个操作。创建 Web 服务使用者转换时，需要选择要运行的操作。可以查看 Web 服务使用者转换中操作输入和操作输出的层次结构。这些层次结构将定义 SOAP 请求消息和 SOAP 响应消息的结构。

还可以通过单向输入操作导入 WSDL。在通过单向输入操作导入 WSDL 时，必须创建虚拟输出端口。

Web 服务使用者转换端口

查看转换端口时，如果不需要查看操作层次结构，则可以显示端口。 显示端口时，可以定义组，定义端口，以及将操作输出中的节点映射到输出端口。

下图显示了一个不可重用的 Web 服务使用者转换的端口：



一个 Web 服务使用者转换可以包含多个输入组和多个输出组。创建端口后，可以创建组，然后将这些端口添加到组中。请基于操作输入层次结构或操作输出层次结构的结构在组层次结构中定义端口。通过添加键，可以将子组与父组关联。在层次结构中，除了最下面的组外，其他所有组都必须具有主键。在层次结构中，除了根组以外，其他所有组都必须具有外键。

该转换包含一个名为 RequestInput 的根输入组。必须在这个根输入组中添加主键。主键必须为 String、Bigint 或 Integer 类型。

可以向该根输入组添加其他传递端口。这些传递端口会将数据不加修改地传递给转换。传递端口可以在输入数据中出现一次。可以向任何输出组添加传递端口。将输出端口关联到输入端口。传递给 SOAP 请求的输入值会在 SOAP 响应的输出行中重复出现。

还可以向根输入组添加 HTTP 表头、Cookie 端口、动态 URL 端口和用于 Web 服务安全身份验证的端口。根组中的数据只出现一次。

要将操作输出节点映射到输出端口，请单击**位置**列中的字段并展开**选择位置**对话框中的层次结构。然后，从层次结构中选择一个节点。

HTTP 表头输入端口

Web 服务可能需要其他 HTTP 表头。 可以在根输入组中创建输入端口，以将其他表头信息传递到 Web 服务提供程序。

要添加 HTTP 表头和 HTTP 端口，请选择根输入组，然后单击**新建**按钮旁边的箭头。 然后，单击 **HTTP 表头**。输入表头名称和端口名称。

可以创建多个 HTTP 表头。

其他输入端口

可以将预定义的输入端口添加到 Web 服务使用者转换。

可以添加以下的预定义输入端口：

Cookie 端口

可以配置 Web 服务使用者转换以使用 Cookie 身份验证。 远程 Web 服务服务器会根据 cookie 跟踪 Web 服务使用者用户。 某个映射多次调用一个 Web 服务时，可以提高性能。

将 cookie 端口映射到 Web 服务请求消息时，Web 服务提供程序将在响应消息中返回 cookie 值。 您可以将该 cookie 值传递到映射中的其他转换下游，或者也可以将该 cookie 值保存在文件中。 将 cookie 值保存在文件中时，可以将该 cookie 配置为 Web 服务使用者转换的输入。

您可以将 cookie 输出端口传递到任意 Web 服务使用者转换输出组。

端点 URL 端口

Web 服务使用者转换使用端点 URL 连接到 Web 服务。 可以在 WSDL 文件、Web 服务连接或端点 URL 输入端口中定义端点 URL。 当转换动态接收端口中的 URL 后，数据集成服务将替代在 WSDL 文件或 Web 服务连接中定义的 URL。

对于每个 Web 服务请求，Web 服务使用者转换都可以有一个 URL 端口值。 将端点 URL 端口添加到根输入组。

WS-Security 端口

在 Web 服务连接中启用 Web 服务安全。 启用 Web 服务安全时，必须在 Web 服务连接或 WS-Security 输入端口中定义用户名和密码。

添加 WS-Security 端口时，可以将用户名和密码传递到转换中的输入端口。 当转换动态接收端口中的用户名和密码后，数据集成服务将替代在 Web 服务连接中定义的值。

注意：Web 服务连接有一个用于 HTTP 和 WS-Security 身份验证的用户名和密码。

要添加预定义的输入端口，请在**端口**区域中单击根输入组。 单击**新建**按钮旁边的箭头，然后单击**其他端口**。 选择要添加的端口。

Web 服务使用者转换输入映射

查看转换端口时，系统将显示输入映射，从中可以查看操作输入层次结构。 显示输入映射时，可以定义输入组，定义输入端口，以及将输入端口映射到操作输入节点。

输入映射包含以下区域：

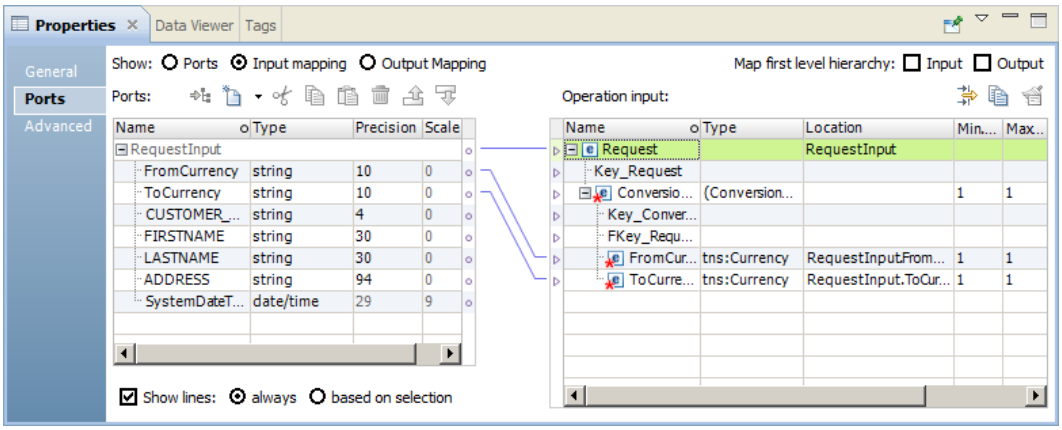
端口

可在**端口**区域中创建转换输入组和输入端口。

操作输入

操作输入区域显示了 Web 服务使用者转换发送到 Web 服务的 SOAP 请求消息中的节点。 用于创建转换的 WSDL 数据对象定义了操作输入层次结构。

下图显示了一个不可重用的 Web 服务使用者转换的输入映射：



创建输入端口后，请将**端口**区域中的输入端口映射到**操作输入**区域中的节点。将输入端口映射到操作输入中的节点后，该端口的位置将显示在**操作输入**区域的**位置**列中。

如果选择映射操作输入层次结构的第一级，则 Developer 工具会将操作输入第一级中的节点映射到输入端口。开发程序工具还会创建端口以执行映射。如果层次结构的第一级包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer 工具不会映射该层次结构的第一级。

可以将一个字符串或文本输入端口中的 XML 数据映射到整个 SOAP 请求消息。如果将 XML 数据映射到整个 SOAP 请求，则无法将端口映射到操作输入中的节点。

可以选择查看将输入端口与操作输入中的节点连接在一起的线。

相关主题：

- “生成 Web 服务 SOAP 消息概览” 页面上 600

将输入端口映射到节点的规则和准则

将输入端口映射到操作输入层次结构中的节点时，请遵循以下规则：

- 可以将一个输入端口映射到层次结构中的一个节点。 可以将同一端口映射到层次结构中任意数量的键。
- 输入端口和节点的数据类型必须兼容。
- 可以将一个输入组中的端口映射到操作输入中的多个层次结构级别。
- 必须将输入端口映射到操作输入中的键。 映射到键的任何端口都必须为 String、Integer 或 Bigint 数据类型。 将数据映射到操作输入中位于 SOAP 消息所含层次结构级别之上的所有级别中的键。 对于映射级别及之上的所有级别，需要包括外键。

  注意: 如果只打算映射操作输入层次结构的最低级别，则无需将输入端口映射到键。
- 可以将多个 String、Bigint 或 Integer 输入端口映射到**操作输入**区域中的键以创建复合键。 为复合键单击**位置**字段时，可以重新对输入端口进行排序，或者删除其中一个端口。

自定义视图选项

您可以更改操作输入层次结构以便在**操作输入**区域中显示键。 此外，还可以显示分组构造，用于定义节点的排序方式。

单击**自定义视图**按钮 ( )，该按钮位于**操作输入**区域中。 启用以下任意选项：

“序列”、“选项”和“全部”

显示一行，指示元素定义是“序列”、“选项”还是“全部”。

“全部”组中的节点都必须包含在 SOAP 消息中。

“序列”组中的节点必须符合 WSDL 中指定的顺序。

“选项”组中必须至少有一个节点显示在 SOAP 消息中。

键

在**操作输出**区域中查看键。 **操作输入**区域包含每个组的键。 可以在**端口**区域中将键添加到输入端口。

将输入端口映射到操作输入

显示转换输入映射时，您可以定义输入组、定义输入端口并将输入端口映射到操作输入节点。

1. 打开 Web 服务使用者转换。
2. 要显示转换输入映射，请使用以下方法之一：
   - 对于可重用转换，请单击**概览**视图。 选择显示输入映射。
   - 对于不可重用转换，请单击**属性**视图中的**端口**选项卡。 选择显示输入映射。
3. 为根输入组定义主键。
4. 要将输入组或端口添加到**端口**区域，请使用以下方法之一：

选项	描述
拖动节点	将**操作输入**区域中的组节点或子节点拖动到**端口**区域的空列中。 如果该节点是组节点，则 Developer 工具将添加组，而不添加端口。
手动添加组或端口	要添加组，请单击**新建**按钮旁边的箭头，然后单击**组**。 要添加端口，请单击**新建**按钮旁边的箭头，然后单击**字段**。
从其他转换中拖动端口	在编辑器中，将其他转换中的端口拖动到 Web 服务使用者转换。
复制端口	从其他转换中选择端口，并将其复制到**端口**区域。 要复制端口，可以使用键盘快捷键或 Developer 工具中的复制和粘贴按钮。
选择**映射第一级层次结构**	选择**映射第一级层次结构**。 Developer 工具会将第一级操作输入中的节点映射到输入端口和组。 Developer 工具还会创建输入端口和组以执行映射。

5. 如果手动创建端口或从其他转换中复制端口，请单击**操作输入**区域中的**位置**列，然后从列表中选择端口。
6. 要将输入端口映射为复合键，请使用以下方法之一：

选项	描述
拖动输入端口	选择两个或多个输入端口，然后将它们拖动到操作输入层次结构的键中。
从**选择位置**对话框中选择输入端口	在操作输入层次结构中单击键的**位置**列，然后选择输入端口。

7. 要清除节点的位置，请使用以下方法之一：

选项	描述
单击**清除**	在**操作输入**区域中选择一个或多个节点，然后单击**清除**。
删除将端口连接到节点的行	在操作输入中选择将输入端口连接到节点的一个或多个行，然后按**删除**。

8. 如果关联的 WSDL 数据对象包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在**操作输入**区域中选择对象。 在节点的**类型**列中，单击**选择**，然后从列表中选择一个或多个类型、元素或属性。
9. 要将字符串或文本输入端口中的 XML 数据映射到完整的 SOAP 请求，请在端口中右键单击，然后选择**映射为 XML**。

Web 服务使用者转换输出映射

查看转换端口时，系统将显示输出映射，从中可以查看操作输出层次结构。 显示输出映射时，可以定义输出组，定义输出端口，以及将操作输出节点映射到输出端口。

输出映射包含以下区域：

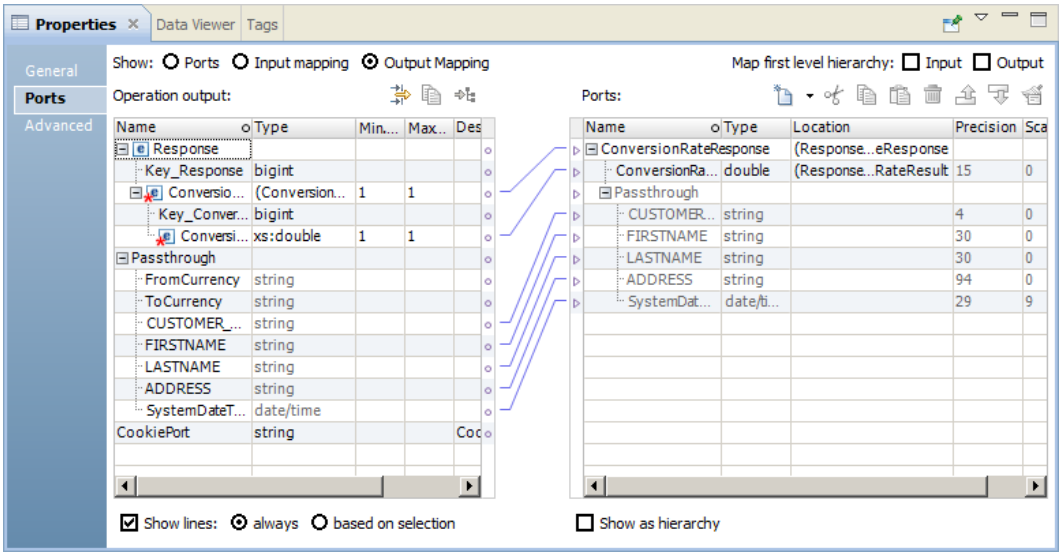
操作输出

操作输出区域显示了 Web 服务返回给 Web 服务使用者转换的 SOAP 响应消息中的节点。 用于创建转换的 WSDL 数据对象定义了操作输出层次结构。

端口

可在**端口**区域中创建转换输出组和端口。

下图显示了一个不可重用的 Web 服务使用者转换的输出映射：



创建输出端口后，将**操作输出**区域中的节点映射到**端口**区域中的端口。将操作输出中的节点映射到输出端口时，节点的位置将显示在**端口**区域的**位置**列中。

选择映射操作输出层次结构的第一级时，Developer 工具会将其中的节点映射到输出端口。开发程序工具还会创建端口以执行映射。如果层次结构的第一级包含一个多次出现的父节点，并包含一个或多个多次出现的子节点，则 Developer 工具不会映射该层次结构的第一级。

可以选择在层次结构中显示输出端口。每个子组都显示在父组下方。还可以选择查看将操作输出中的节点连接到输出端口的行。

如果从存储库中删除了关联的 WSDL 数据对象，则 Developer 工具会将操作节点的位置保留在输出映射中。显示输出映射时，**端口**区域仍将在输出端口的**位置**列中显示操作节点的位置。如果对转换关联了其他 WSDL，则 Developer 工具会检查每个位置是否有效。如果操作节点位置不再有效，则 Developer 工具将在输出映射的**端口**区域中清除该位置。

相关主题：

- “解析 Web 服务 SOAP 消息概览” 页面上 593

将节点映射到输出端口的规则和准则

将操作输出层次结构中的节点映射到输出端口时，请查看以下规则：

- 操作输出节点和输出端口必须使用兼容的数据类型。
- 不能将一个节点映射到组中的多个输出端口。
- 除了传递端口，每个输出端口都必须具有有效位置。
- 如果要将多次出现的子节点拖动到空的输出端口，则必须将该组关联到其他输出组。 选择组后，Developer 工具将创建用于关联组的键。
- 将多次出现的元素拖动到包含父元素的组时，可以配置要包含的子元素的出现次数。 也可以将父组替换为转换输出中多次出现的子组。

将 SOAP 消息映射为 XML

可以将完整的 SOAP 消息映射为 XML，而不是将数据返回到不同的输出端口。

将 SOAP 消息映射为 XML 后，数据集成服务会在一个端口中返回完整的 SOAP 消息。 不要创建输出端口。

要映射完整的消息，请右键单击**操作输出**区域中的根组。 选择**映射为 XML**。

Developer 工具将创建一个字符串输出端口。 精度为 65535 个字节。

自定义视图选项

可以通过更改操作输出层次结构，在**操作输出**区域中显示 Cookie 端口、传递端口和键。 此外，还可以显示分组构造，用于定义节点的排序方式。

单击**自定义视图**按钮，该按钮位于**操作输出**区域中。 启用以下任意选项：

“Sequence”、“Choice”和“all”

显示一行，指示元素定义是“sequence”、“choice”还是“all”。

“all”组中的节点都必须包含在 SOAP 消息中。

“sequence”组中的节点必须符合 WSDL 中指定的顺序。

“choice”组中必须至少有一个节点显示在 SOAP 消息中。

键

在**操作输出**区域中查看键。 **操作输出**区域包含每个组的键。 可以在**端口**区域中将键添加到输出端口。

传递端口

操作输出区域显示传递端口。 传递端口是指在转换中传递数据而不更改数据的端口。 可以将操作输出中的传递端口映射到任何 Web 服务使用者转换输出组。 传递端口会一次性接收数据，因此，该端口处于 SOAP 消息中的根级别。

Cookie 端口

显示 Cookie 端口。 如果配置 Cookie 身份验证，则远程 Web 服务服务器将根据 Cookie 跟踪 Web 服务使用者用户。 如果映射请求消息中的 Web 服务 Cookie，则 Web 服务将在响应消息中返回 Cookie。 可以将操作输出中的 Cookie 映射到任何 Web 服务使用者转换输出组。

将操作输出映射到输出端口

显示转换输出映射时，您可以定义输出组、定义输出端口并将操作输出节点映射到输出端口。

1. 打开 Web 服务使用者转换。
2. 要显示转换输出映射，请使用以下方法之一：
   - 对于可重用转换，请单击**概览**视图。 选择显示输出映射。
   - 对于不可重用转换，请单击**属性**视图中的**端口**选项卡。 选择显示输出映射。
3. 要将输出组或端口添加到**端口**区域，请使用以下方法之一：

选项	描述
拖动节点	将**操作输出**区域中的组节点或子节点拖动到**端口**区域的空列中。 如果该节点是组节点，则 Developer 工具将添加组，而不添加端口。
手动添加组或端口	要添加组，请单击**新建**按钮旁边的箭头，然后单击**组**。 要添加端口，请单击**新建**按钮旁边的箭头，然后单击**字段**。
从其他转换中拖动端口	在编辑器中，将其他转换中的端口拖动到 Web 服务使用者转换。
复制端口	从其他转换中选择端口，并将其复制到**端口**区域。 要复制端口，可以使用键盘快捷键或 Developer 工具中的复制和粘贴按钮。
选择**映射第一级层次结构**	选择**映射第一级层次结构**。 Developer 工具会将第一级操作输出中的节点映射到输出端口和组。 Developer 工具还会创建输出端口和组以执行映射。

4. 如果手动创建端口或从其他转换中复制端口，请单击**端口**区域中的**位置**列，然后从列表中选择节点。
5. 要清除端口的位置，请使用以下方法之一：

选项	描述
单击**清除**	在**端口**区域中选择一个或多个端口，然后单击**清除**。
删除将节点连接到端口的行	在操作输出中选择将节点连接到输出端口的一个或多个行，然后按**删除**。

6. 如果关联的 WSDL 数据对象包含 anyType 元素、any 元素、anyAttribute 属性、派生类型元素或置换组，请在**操作输出**区域中选择对象。 在节点的**类型**列中，单击**选择**，然后从列表中选择一个或多个类型、元素或属性。
7. 要将完整的 SOAP 响应消息映射为 XML，请在**操作输出**区域中右键单击根组，然后选择**映射为 XML**。

Web 服务使用者转换高级属性

Web 服务使用者转换高级属性包括跟踪级别、通用故障端口、Web 服务连接和并发 Web 服务请求消息。

可以在“高级”选项卡中为 Web 服务使用者转换定义以下高级属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

SOAP 操作

对于 Web 服务使用者转换，需使用常量值替代 WSDL 中定义的 SOAP 操作值。

启用通用 SOAP 故障处理

返回 WSDL 中未定义的故障消息。 在 GenericFault 输出组中创建输出端口以处理故障代码和消息。

下表介绍了 SOAP 1.1 和 SOAP 1.2 的故障输出端口：

SOAP 1.1 的故障输出端口	SOAP 1.2 的故障输出端口	说明
Fault Code	Code*	返回故障标识代码。
Fault String	Reason*	在故障消息中返回错误说明。
Fault Detail	详细端口	在通用故障消息中返回 Web 服务提供程序传递给 Web 服务使用者转换的自定义信息。
Fault Actor	Role	返回导致故障发生的对象的相关信息。
-	Node	返回生成故障的 SOAP 节点的 URI。
* Code 和 Reason 输出端口是分层次的。		

注意: 可以展开 Code 故障输出端口，以便将 SubCode 故障输出端口提升一个级别。

启用 HTTP 错误处理

返回来自 Web 服务的任何 HTTP 错误。 在 GenericFault 输出组中创建 HTTP 错误输出端口。

将故障视为错误

将故障消息添加到映射日志。 出现故障时，数据集成服务将增加映射的错误计数。 禁用此属性才能允许早期选择优化和推入优化。 默认情况下启用该属性。

连接

标识连接到 Web 服务的 Web 服务连接对象。 在 Developer 工具中创建 Web 服务连接。 在 Developer 工具或 Administrator 工具中编辑 Web 服务连接。 配置 Web 服务连接时，请配置端点 URL、Web 服务所需的安全类型和连接超时时限。

Web 服务使用者转换使用端点 URL 连接到 Web 服务。 可以在 WSDL 文件、Web 服务连接或端点 URL 输入端口中定义端点 URL。

使用以下准则确定何时配置 Web 服务连接：

- 如果使用的端点 URL 与 WSDL 文件中的 URL 不同，且不打算使用端点 URL 输入端口，请配置一个该连接。
- 如果连接到的 Web 服务需要 Web 服务安全、HTTP 身份验证或 SSL 证书，请配置一个该连接。

- 如果要更改默认的连接超时时限，请配置一个该连接。

注意: 可以将存储库中的 WSDL 数据对象与 Web 服务连接关联。 关联的连接会成为从该 WSDL 创建的每个 Web 服务使用者转换的默认连接。

启用压缩

使用 GZIP 压缩方法启用 SOAP 请求编码，然后使用 GZIP 或 deflate 启用 SOAP 响应解码。

XML 架构验证

在运行时验证 SOAP 响应消息。 选择 **XML 无效错误**或**不验证**。

已排序输入

允许数据集成服务在未处理所有输入数据的情况下生成输出。 当输入数据按操作输入层次结构中的键排序时，请启用已排序输入。

推入优化

启用推入优化。 单击**推入优化**属性中的**打开**按钮以选择用于接收筛选值的筛选器端口。 对于每个筛选器端口，请选择在 Web 服务响应中包含已筛选列的输出端口。

有副作用

该复选框指示 Web 服务除返回行之外还执行其他功能。 如果 Web 服务除返回行之外还修改对象或与其他对象或功能进行交互，则 Web 服务使用者转换会产生副作用。 该 Web 服务可能会修改数据库、计入总数、引发异常、编写电子邮件或调用其他具有副作用的 Web 服务。 要允许推入优化或早期选择优化，请禁用**有副作用**属性。 默认情况下启用该属性。

启用并发

允许 Web 服务使用者转换创建多个到同一 Web 服务的并发连接，从而并行发送多个 Web 服务请求。 允许 Web 服务使用者转换创建与 Web 服务的多个并发连接时，可以设置总内存消耗限制和并发连接限制数。

下表介绍了这些选项：

选项	说明
启用并发	创建到同一 Web 服务的多个并发连接。
并发连接限制	并发 Web 服务连接数。 默认值为 20。
总并发内存限制(以 MB 为单位)	所有并发连接的总内存分配限制。 默认值为 100 MB。

Web 服务错误处理

可以将 Web 服务使用者转换配置为将 SOAP 故障和 HTTP 错误传递给映射中的下游。 出现故障时，可以增加错误计数。 可以在转换高级属性中配置 Web 服务错误处理方式。

Web 服务返回响应消息或故障。 故障属于错误。 Web 服务可能会基于出现的错误生成不同的故障。

Web 服务使用者转换可以返回以下类型的故障：

SOAP 故障

WSDL 所定义的 SOAP 故障。 配置可在 Web 服务响应消息中返回故障的输出错误端口。 对于 SOAP 1.1 绑定，数据集成服务将返回该故障的故障消息、故障代码、故障字符串以及故障触发者等元素。 对于 SOAP 1.2 绑定，数据集成服务将返回该故障的故障消息、代码、原因、节点和角色等元素。

通用 SOAP 故障

Web 服务将在运行时生成通用 SOAP 故障。 SOAP 1.1 绑定和 SOAP 1.2 绑定的故障元素不同。 WSDL 未定义通用 SOAP 故障。 通用 SOAP 故障包括身份验证失败和 SOAP 请求错误。

HTTP 错误

如果在转换中启用了 HTTP 错误处理，则 Developer 工具将添加 HTTP 故障输出端口。 数据集成服务将通过一个字符串端口返回来自 Web 服务的 HTTP 错误。 每个 HTTP 错误都包含一个错误代码和一条消息。

如果来自 Web 服务的 SOAP 响应包含无效的 XML 数据，则 Web 服务使用者转换将返回错误。

可以配置是否将 SOAP 故障视为错误。 如果启用了“将故障视为错误”，并且发生 SOAP 故障，则数据集成服务会使映射中的错误计数递增。 该故障将显示在消息日志中。

消息压缩

如果启用了 SOAP 消息压缩，则 Web 服务使用者转换会压缩 Web 服务请求消息，并接收经过压缩的 Web 服务响应消息。

Web 服务使用者转换将使用 GZip 压缩对 SOAP 请求进行编码。 该转换将接受使用 GZip 或 deflate 压缩进行编码的响应消息。

数据集成服务接收来自 Web 服务的响应时，将检查 SOAP 消息中的内容编码 HTTP 表头并对该消息进行解码。

默认设置为无压缩编码。 Web 服务不会压缩 SOAP 响应。

下表显示了在打开或关闭压缩时请求消息和响应消息的表头：

压缩	表头
打开	Content-Encoding 表头：GZip Accept-Encoding 表头：GZip、deflate
关闭	空的 Content-Encoding 表头 空的 Accept-Encoding 表头

有时，Web 服务会使用默认压缩对响应消息进行编码。 如果该消息是通过 GZip 或 deflate 编码的，则 Web 服务使用者转换可对该消息进行解码。 如果 Web 服务是以意外的方式对响应消息进行编码的，则 Web 服务使用者转换会将该消息记录到映射日志中。

可在转换高级属性中启用压缩。

并发

您可以启用 Web 服务使用者转换来创建指向 Web 服务的多个并发连接，以便同时发送多个 Web 服务请求。

例如，在查询银行信息时，可将 Web 服务使用者转换配置为并发，以便同时发送多行。 如果存在 20 个输入行，则可以同时发送 20 个请求，从而加快处理速度。

在 Web 服务使用者转换中启用并发后，可配置总内存消耗限制。

在 Web 服务使用者转换中启用并发后，可配置并发的 Web 服务连接数。

针对并发的规则和准则

使用并发时，请使用以下规则和准则：

- 并发支持将已排序的输入行作为与 Web 服务的多个并发连接： 不支持已排序的输出行。

- 如果数据集超过 100 行，请使用并发。
- 建议不要增加并发 Web 服务连接的数量。 并发 Web 服务连接的数量将链接到操作系统使用的套接字数量。 增加套接字数量的成本较高。
- 使用并发功能时，请使用具有多内核处理器并具有至少 100 MB RAM 的系统，以实现最佳性能。
- 并发内存限制表示调用 Web 服务时并发工作流使用的内存。
- 启用 Web 服务使用者转换中的并发时，可以配置内存使用限制。 请确保内存使用不超过服务器上的物理 RAM。

并发的最佳实践

要在使用并发时获得最佳性能，请遵循以下最佳实践：

- 避免更改总并发内存限制的默认值和并发连接限制的默认值。
- 避免对少于 100 行的数据集使用并发。
- 使用并发时，避免在映射中使用传递端口。

筛选器优化

筛选器优化通过减少传递映射的行数来提高性能。 数据集成服务可以应用早期选择优化或推入优化。

数据集成服务应用筛选器优化方法时，会将筛选器移至尽可能靠近映射中的源的位置。 如果数据集成服务无法将筛选器移至映射中的转换之前，则可以将筛选器逻辑推入转换中。

启用通过 Web 服务使用者转换执行早期选择优化

如果 Web 服务使用者转换没有副作用，并且不会将故障视为错误，请为该转换启用早期选择优化。

1. 打开 Web 服务使用者转换**高级属性**视图。
2. 清除**将故障视为错误**。
3. 清除**有副作用**。

通过 Web 服务使用者转换执行推入优化

在 SQL 数据服务中，当 Web 服务使用者转换位于虚拟表中时，您可以通过该转换配置推入优化。

映射调用 Web 服务以根据最终用户 SQL 查询中的语句检索数据集或数据子集。 最终用户 SQL 查询中包含一个可选筛选条件。

通过推入优化，Web 服务使用者转换在筛选器端口中接收筛选器值。 筛选器端口是指未连接的输入端口，您在配置推入优化时将该端口标识为筛选器端口。 筛选器端口具有默认值，该值可确保最终用户查询中不包含筛选器时 Web 服务返回所有行。 筛选器端口不是传递端口。

注意: 筛选器字段必须属于 Web 服务请求中根组的一部分。

配置筛选器端口时，应将 Web 服务使用者转换中负责从 Web 服务响应中接收列数据的输出端口标识为输出端口。 例如，如果筛选器端口是指名为 EmployeeID 的输入端口，响应中的输出端口可能是名为 EmployeeNum 的端口。 Developer 工具需要与输入筛选器端口和输出端口关联，以便将筛选器逻辑从读取的虚拟表推送到 Web 服务使用者请求。 Web 服务请求的输入端口通常与 Web 服务响应中的输出端口不同。

筛选器字段不能是传递端口。 配置筛选器端口时，端口的默认值更改为筛选条件的值，因而传递输出端口值会发生变化。 基于输出传递端口的筛选器返回意外结果。

您可以将多个筛选器表达式推送到 Web 服务使用者转换。 每个筛选条件的格式都必须如下所示：

```
<Field> = <Constant>
```

筛选条件必须通过 AND 联接。 不能通过 OR 联接这些条件。

通过 Web 服务使用者转换执行推入优化示例

SQL 数据服务返回所有客户的订单，或者根据收到的来自用户的 SQL 查询返回特定客户的订单。

数据服务中包含具有以下组件的逻辑数据对象：

客户表

包含客户信息的 Oracle 数据库表。

Web 服务使用者转换

调用 Web 服务的转换可检索客户的最新订单。 Web 服务使用者转换包含 customerID 和 orderNum 对应的输入端口。 该转换具有包含接收的来自客户表的客户数据的传递端口。 orderNum 端口为筛选器端口，处于未连接状态。 orderNum 的默认值为“*”。 Web 服务在 Web 服务请求中收到此值后，将返回所有订单。

订单虚拟表

从 Web 服务接收客户和订单数据的虚拟表。 最终用户会查询此虚拟表。 订单中包含客户列、订单 ID 列以及客户和订单数据。

最终用户将以下 SQL 查询传递到 SQL 数据服务：

```
SELECT * from OrdersID where customer = 23 and orderID = 56
```

数据集成服务将拆分该查询以优化映射。 数据集成服务使用早期选择优化并将筛选器逻辑 `customer = 23` 移至读取的客户表。 数据集成服务使用推入优化并将筛选器逻辑 `orderID = 56` 推送到 Web 服务使用者转换筛选器端口。 Web 服务使用者转换检索客户 23 的订单 ID 56。

启用通过 Web 服务使用者转换执行推入优化

如果 Web 服务使用者转换没有副作用，并且不会将故障视为错误，请为该转换启用推入优化。

1. 打开 Web 服务使用者转换**高级属性**视图。
2. 清除**将故障视为错误**。
3. 清除**有副作用**。
4. 单击**推入优化**属性中的**打开**按钮。
5. 在“已优化输入”对话框中选择筛选器端口名称。
   您可以选择多个筛选器端口。
6. 单击**输出**列。
7. 对于每个筛选器端口，请选择在 Web 服务响应中包含已筛选列的输出端口。
8. 输入每个筛选器端口的默认值。

   注意: 不能为 Web 服务使用者端口配置默认值，除非该端口为筛选器端口。

创建 Web 服务使用者转换

可以创建可重用或不可重用的 Web 服务使用者转换。 可重用转换可存在于多个映射中。 不可重用转换存在于单个映射内。

可以从单个 WSDL 对象创建 Web 服务使用者转换以用于 SOAP 1.1 绑定和 SOAP 1.2 绑定。

1. 要创建转换，请使用以下方法之一：

选项	说明
可重用	在**对象浏览器**视图中选择一个项目或文件夹。 单击**文件** > **新建** > **转换**。 选择 Web 服务使用者转换，然后单击**下一步**。
不可重用	在映射或 Mapplet 中，将“转换”选项板中的 Web 服务使用者转换拖动到编辑器中。

   此时将显示**新建 Web 服务使用者转换**对话框。

2. 浏览并选择 WSDL 数据对象以定义 Web 服务请求和响应消息。

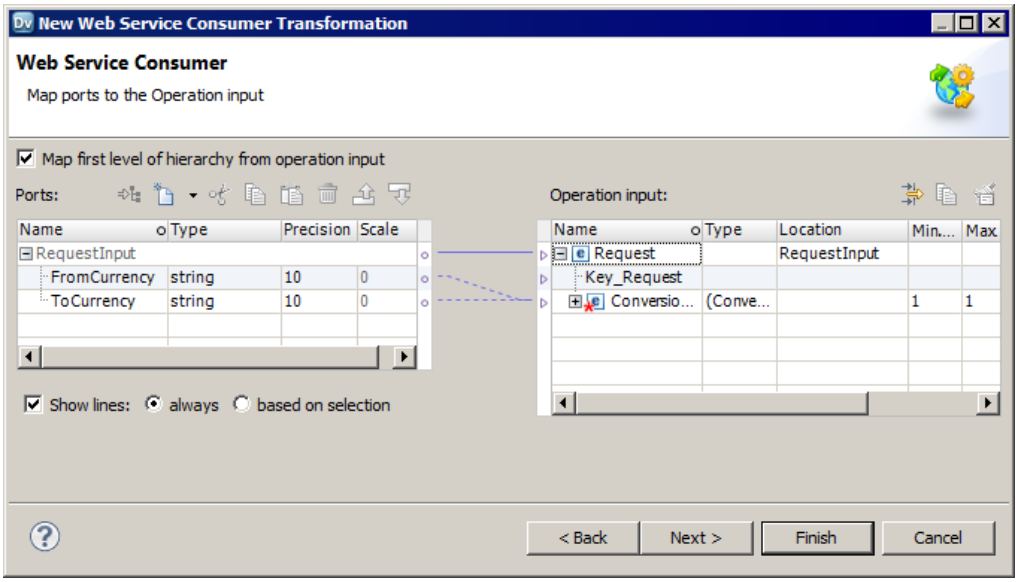
   如果 WSDL 不在存储库中，可以从“新建 Web 服务使用者转换”对话框中导入 WSDL。

3. 浏览并从 WSDL 中选择一项操作。

   可以选择包含 SOAP 1.1 绑定或 SOAP 1.2 绑定的操作。

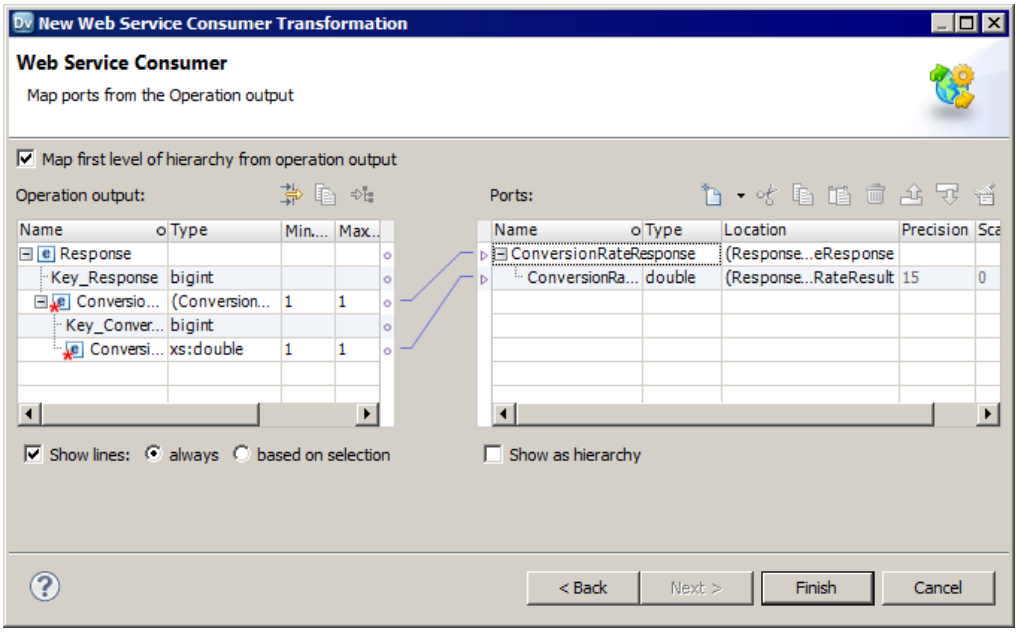
4. 单击**下一步**。

   此时将显示**将端口映射到操作输入**屏幕。 **端口**区域显示转换输入组和输入端口。 **操作输入**区域显示请求消息层次结构。



5. 定义输入组和输入端口，并将输入端口映射到操作输入节点。
6. 单击**下一步**。

此时将显示**从操作输出中映射端口**屏幕。 **操作输出**区域显示响应消息层次结构。 **端口**区域显示转换输出组和输出端口。



7. 定义输出组和输出端口，并将操作输出节点映射到后者。
8. 单击**完成**。
9. 单击**高级**视图以配置转换属性和 Web 服务连接。

相关主题：

- “将输入端口映射到操作输入” 页面上 579
- “将操作输出映射到输出端口” 页面上 582
- “Web 服务使用者转换高级属性” 页面上 583

Web 服务使用者转换示例

您的组织需要向销售组织公开 RT100 产品线的订单信息。 销售团队需要每日查询订单摘要和订单详细信息。

创建一个逻辑数据对象，该对象可公开虚拟表中的每日订单信息。 读取映射包含一个 Web 服务使用者转换，该转换可返回最新的 RT100 订单。 该 Web 服务使用者转换将使用一个 Web 服务返回 RT100 产品线的每日订单摘要和订单详细信息。

输入文件

输入文件是包含产品线编号的平面文件。

请创建物理数据对象以定义输入文件。 该文件包含一个字段，即 Product_Line。 字段值为 RT100。 在**运行时属性**视图中定义物理数据对象的位置。

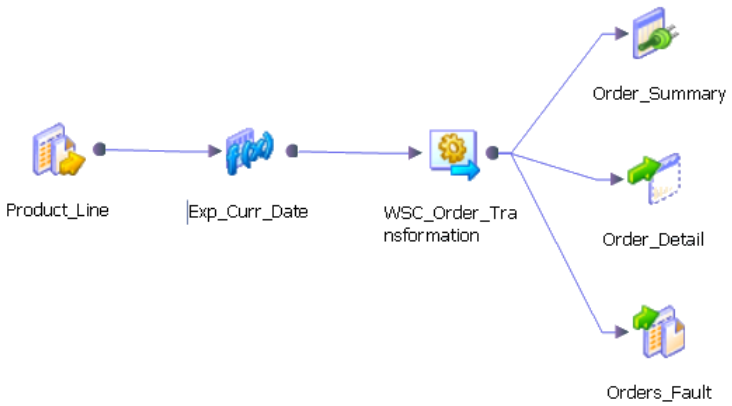
逻辑数据对象模型

组织中的业务分析将创建用于描述顺序摘要和顺序详细表结构的逻辑数据模型。逻辑数据模型包含 Order_Summary 和 Order_Detail 逻辑数据对象。

该分析通过定义逻辑数据模型的建模工具创建架构。您可以从架构中导入逻辑数据模型，并创建 Order_Summary 和 Order_Detail 逻辑数据对象。

逻辑数据对象映射

逻辑数据对象映射介绍如何通过逻辑数据对象访问数据。



读取映射包含以下对象：

Product_Line

包含产品线编号的输入平面文件。

Exp_Curr_Date 转换

用于返回当前日期和 Web 服务使用者转换根级别输入组主键的表达式转换。

WSC_Order 转换

使用 Web 服务检索顺序信息的 Web 服务使用者转换。 该转换会将产品线和当前日期传递给请求消息中的 Web 服务。 该转换将通过响应消息中的 Web 服务接收顺序信息。

Order_Summary 表

包含顺序信息的逻辑数据对象，例如 Order_No、Customer_Id、Qty 和 Order_Date。

Order_Detail 表

包含详细顺序信息的逻辑数据对象，例如 Order_No、Customer_Id、Qty 和 Status。

Orders_Fault

用于接收通用故障消息的输出平面文件。

Web 服务使用者转换

Web 服务使用者转换会将产品线、日期和序列号作为输入来接收。 转换将使用 Get_Order_Info Web 服务操作以检索订单信息。

创建 Web 服务使用者转换时，请选择描述请求并响应 Web 服务消息的 WSDL 数据对象。 Web 服务消息包含 XML 元素的层次结构组。 一个元素可能包含其他元素。 某些元素可能出现多次。 从存储库中的 Order_Info WSDL 对象创建转换。

配置转换输入端口并将这些端口映射到操作输入层次结构。 将操作输出层次结构的节点映射到输出端口。 定义 Web 服务连接和运行时属性。

转换输入映射

在**端口**视图中显示输入映射时，您可以定义输入端口并将其映射到操作输入中的节点。

转换**端口**区域具有一个根组和一个订单组。 根组为请求输入组。 将一个端口添加到请求输入组来表示主键。

订单组具有 **Select_Date** 和 **Select_Product_Line** 输入端口。

将输入端口映射到**操作输入**区域的 **Order_Date** 和 **Product_Line** 节点。

操作输入区域定义 Web 服务使用者转换传递到 Web 服务的请求消息。 默认情况下，这些节点将出现在**操作输入**区域。

转换输出映射

在**端口**视图上显示输出映射时，您可以通过将操作输出的节点映射到转换输出组来定义输出端口。

Web 服务将在 Web 服务响应消息中返回以下层次结构：

```
Response
   Orders
      Order
         Key_Order
         Order_ID
         Order_Date
         Customer_ID
         Total_Qty
         Order_Details
            Order_Detail
               Product_ID
               Description
               Qty
               Status
```

Web 服务返回多个订单。 Order 为 Orders 级别中多次出现的节点。 对于每个订单，Web 服务可以返回多个订单明细。 Order_Detail 为 Order_Details 级别中多次出现的节点。

注意: Developer 工具将 Key_Order 节点添加至用户界面中。 您可以将键映射到输出组以定义组之间的关系。 例如，Order_ID 是 Order 中的主键，同时也是 Order_Details 中的外键。

在**端口**区域创建以下输出组：

```
Order
    Order_ID
    Order_Date
    Customer_ID
    Total_Qty

Order_Detail
    Order_ID
    Product_ID
    Description
    Qty
    Status
```

每当 Order_ID 的值更改时，数据集成服务就会从 Order 组写入一行。

每当 Order_ID 和 Product_ID 的值更改时，数据集成服务就会从 Order_Detail 组写入一行。

转换高级属性

请为 Web 服务使用者转换配置以下高级属性：

启用通用 SOAP 故障处理

添加用于接收 SOAP 故障消息的输出端口。

连接

选择一个 Web 服务连接以访问 Web 服务。

启用压缩

Web 服务使用者转换将使用 GZIP 压缩 Web 消息。

第 48 章

解析 Web 服务 SOAP 消息

本章包括以下主题：

解析 Web 服务 SOAP 消息概览

数据集成服务在解析 Web 服务转换中的 SOAP 消息时生成行数据。

Web 服务输入转换和 Web 服务使用者转换是解析 SOAP 消息的 Web 服务转换。

要将转换配置为解析 SOAP 消息，可以采用类似于 SOAP 消息层次结构的结构创建输出端口。 将 SOAP 消息层次结构中的节点映射到端口。

可以配置规范化输出端口组、非规范化组和透视端口组。 如果 SOAP 消息包含派生类型、anyType 元素或置换组，可以根据 SOAP 消息实例中可能出现的类型配置不同的输出组。

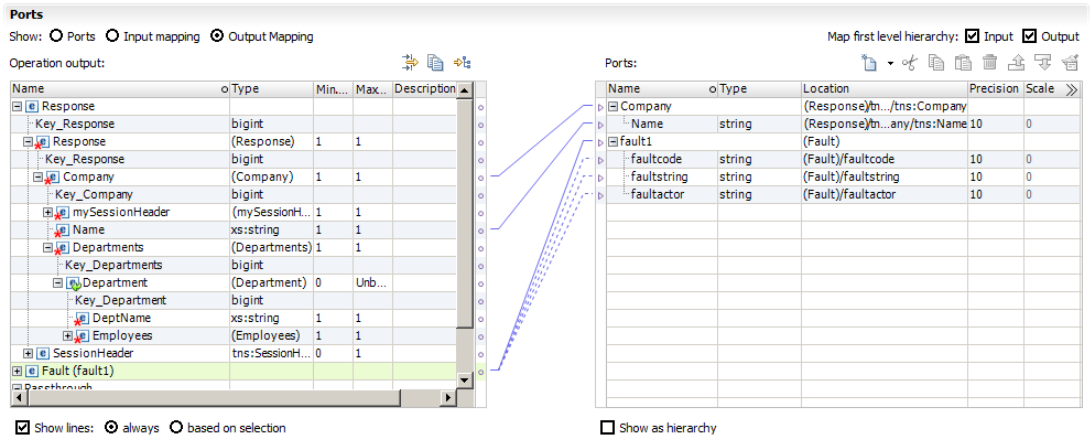
相关主题：

- “Web 服务使用者转换输出映射” 页面上 580

转换用户界面

Web 服务使用者转换和 Web 服务输入转换提供了一个用户界面，可用来将数据从 SOAP 消息映射到转换输出端口。

下图显示了 SOAP 1.1 消息节点与 Web 服务使用者转换中的输出端口之间的映射：



操作区域

操作区域包含 SOAP 消息层次结构。 复杂节点或多次出现节点定义结构中的层次结构级别。 Developer 在级别中添加键，用以定义这些级别之间的父子关系。

在上图中，SOAP 消息层次结构具有以下级别：

响应或请求

该级别表示响应或请求消息的根。

公司

请求数据的最高级别。

部门

公司内的多次出现部门。

员工

员工是部门内的一个复杂元素。

故障组

接收错误消息的故障消息组。

端口区域

可以将数据从 SOAP 消息级别映射到输出端口。 每个输出端口组均可通过主外键关系与其他输出组关联。

在上图中，转换有多个与 SOAP 消息中的节点组相对应的输出端口组。

多次出现输出配置

当输入转换或 Web 服务使用者转换返回多次出现数据时，可以将输出端口配置为不同的配置。

可以配置规范化输出数据、透视输出数据或非规范化输出数据。

例如，SOAP 消息包含部门和员工复杂元素。 每个部门包含多个员工。 部门是员工的父级别。

SOAP 消息包含以下元素层次结构：

```
Departments
      Department_ID
      Department_Name
      Employees
```

```
Employee_ID
Employee_Name
```

规范化的关系输出

创建规范化输出数据时，数据值不会在输出组中重复。在 SOAP 消息中的层次结构级别和输出端口组之间创建一对一关系。

如果 SOAP 消息包含部门父级层次结构级别和员工子级层次结构级别，可创建以下端口组：

```
Departments
    Department_Key
    Department_ID
    Department_Name


Employees
    Department_Key
    Employee_ID
    Employee_Name
```

Department_Key 是一个将 Employees 输出组与 Department 组相关联的生成键。

生成的键

添加输出组时，Developer 工具通过生成的键将输出组与其他输出组关联。 Developer 工具在父级组和子级组中添加长整型键。 数据集成服务在运行时创建生成的键的键值。

示例

SOAP 层次结构含有以下节点：

```
Departments
   Dept_Key
   Dept_Num
   Dept_Name

   Employees
      Dept_FK
      Employee_Num
      Employee_Name
```

为部门创建输出端口组时，将部门节点映射到端口区域中的一个空字段。 Developer 工具将创建以下输出组：

```
Departments
   Dept_Num
   Dept_Name
```

如果将员工节点映射到端口区域中的一个空字段，Developer 工具会提示您将员工组与部门组关联。 可以将员工组与多个组关联。 Developer 工具在每个组中添加一个键。

Developer 工具创建以下组和生成的键：

```
Departments
   Key_Departments
   Dept_Num
   Dept_Name

   Employees
      Key_Departments
      Employee_Num
      Employee_Name
```

注意: 不必将节点映射到生成的键。 数据集成服务在运行时创建键值。

Developer 工具可以在一个输出组中创建多个级别的生成的键。 员工组可包含以下端口：

```
Employees
   Key_Employees
   Key_Departments
   Key_Managers
   Employee_Num
   Employee_Name
```

Key_Departments 和 Key_Managers 是指向父级组的生成的键。 Key_Employees 是针对员工组的生成的键。将子级组关联到员工组时，将显示 Key_Employees。

非规范化的关系输出

可以对关系输出进行非规范化处理。对输出数据进行非规范化处理时，父级组中的元素值会对每个子元素重复。

要对输出数据进行非规范化，请将父级层次结构中的节点映射至子级输出端口组。

以下示例显示了 Employees 输出组中的 Department_ID 和 Department_Name：

```
Employees
    Department_ID
    Department_Name
    Employee_ID
    Employee_Name
```

对于部门中的每名员工，Department_ID 和 Department_Name 均重复：

Department_ID	Department_Name	Employee_ID	Employee_Name
100	Accounting	56500	Kathy Jones
100	Accounting	56501	Tom Lyons
100	Accounting	56509	Bob Smith

已转换的关系输出

您可以在一个输出组中包含特定数量的多次出现的元素。

要转换多次出现的元素，请将多次出现的子元素映射至父级输出端口组。Developer 工具会提示您定义要包含在父级输出端口组中的子元素数量。

下例显示了部门父级组中的两个 Employee_ID 实例：

```
Departments
    Department_ID
    Department_Name
    Employee_ID1
    Employee_ID2
```

解析 anyType 元素

anyType 元素代表 WSDL 或架构中所有全局类型的一个选项。 将节点映射到 Developer 工具中的端口时，选择要在 SOAP 消息中为 anyType 元素显示的类型。 必须将 SOAP 消息中的 anyType 元素替换为复杂类型或 xs:string。 为所选的每个类型创建端口组。

必须选择类型以将数据映射到输出端口。 如果 WSDL 或架构不包含全局类型，Developer 工具会将 anyType 元素替换为 xs:string。

要在操作区域中选择元素类型，请在 anyType 元素的**类型**列中单击**选择**。 此时将显示一列可用的复杂类型和 xs:string。

将 anyType 元素替换为派生类型时，数据集成服务一次为一个类型填充元素。 SOAP 消息不同时包含基本类型和派生类型的数据。

派生类型示例

WSDL 包含 anyType 元素。 将该元素替换为 AddressType 和称为 USAddressType 的派生类型。 SOAP 消息层次结构包含以下组：

```
Address:AddressType (base type)
  Address: AddressType
    Street
    City

  Address:USAddressType   (derived type)
     Street
     City
     State
     ZipCode
```

SOAP 消息包含以下数据：

```
<address xsi: type ="AddressType">
<street>1002 Mission St.</street>
<city>san jose</city>
</address>

<address xsi:type="USAddressType">
<street>234 Fremont Blvd</street>
<city>Fremont</city>
<zip>94556</zip>
<state>CA</state>
</address>
```

数据集成服务为 xsi:AddressType 返回一个行：

街道	城市
1002 Mission St.	San Jose

数据集成服务为派生类型 xsi:USAddressType 返回一个行：

街道	城市	省/自治区/直辖市	邮政编码
234 Fremont Blvd.	Sunnyvale	CA	94556

如果类型为 xsi:USAddressType，数据集成服务将不填充 AddressType。

解析派生类型

可以解析包含派生类型的 SOAP 消息。 定义从 SOAP 消息接收数据的端口时，选择可在 SOAP 消息中出现的类型。 所选类型的元素决定了需要创建的端口。

例如，WSDL 可包含 AddressType 和称为 USAddressType 的派生类型。 可以在 Developer 工具操作区域创建以下组：

```
Address
  Address: AddressType
    Street
    City
```

```
Address:USAddressType
   Street
   City
   State
   ZipCode
```

SOAP 消息可包含以下数据：

```
<address>
<street>1002 Mission St.</street>
<city>san jose</city>
</address>

<address xsi:type="USAddressType">
<street>234 Fremont Blvd</street>
<city>Fremont</city>
<zip>94556</zip>
<state>CA</state>
</address>

<address xsi:type="USAddressType">
<street>100 Cardinal Way</street>
<city>Redwood City</city>
<zip>94536</zip>
<state>CA</state>
</address>

<address>
<street>100 El Camino Real</street>
<city>Sunnyvale</city>
</address>
```

对于基本类型 Address，数据集成服务返回以下行：

街道	城市
1002 Mission St.	San Jose
234 Fremont Blvd	Sunnyvale
100 Cardinal Way	Redwood City
100 El Camino Real	Sunnyvale

对于派生类型 USAddress，数据集成服务返回以下行：

街道	城市	省/自治区/直辖市	邮政编码
234 Fremont Blvd.	Sunnyvale	CA	94556
100 Cardinal Way	Redwood City	CA	94536

数据集成服务返回所有基本类型的地址。 数据集成服务返回派生类型的 US 地址。 派生类型包括 USAddressType 从基本类型继承的街道和城市元素。

解析 QName 元素

数据集成服务解析 SOAP 消息中的 QName 元素时，其更新属于架构命名空间的 QName 值，以使用在架构中定义的命名空间前缀。 其他情况下数据集成服务不更新元素值。

例如，架构为命名空间 "http://user/test" 定义了命名空间前缀 tns。 SOAP 消息为同一个命名空间定义了命名空间前缀 mytns。 当数据集成服务解析 Qname 值 mytns:myelement 时，其将该值改为 tns:myElement。

数据集成服务在 SOAP 消息中生成 QName 元素时，其不更新元素的值。

解析置换组

置换组将一个元素替换为同一组中的其他元素。 置换组类似于派生类型，但每个元素定义都包含一个置换组名称。

可以配置从置换组中的特定类型接收元素的输出端口组。 可以创建从置换组中的其他类型接收元素的不同的输出端口组。

解析 SOAP 消息中的 XML 构造

SOAP 消息可能包含选项、列表和联合元素等 XML 构造。

Web 服务转换可以解析包含这些构造的 SOAP 消息，但受到一些限制。

选项元素

选项元素将子元素限制为 <choice> 声明中的一个元素。

下面的文字显示了一个身份为员工或承包商的人员元素：

```
<xs:element name="person">
  <xs:complexType>
    <xs:choice>
      <xs:element name="employee" type="employee"/>
      <xs:element name="contractor" type="contractor"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
```

可以使用以下方法映射选项元素：

- 为输出组中的每个选项元素创建输出端口。 部分元素在输出行中为空值。
- 为每个选项创建输出组。 对于上例，创建员工组和承包商组。 数据集成服务根据在 SOAP 消息中显示的元素生成行。

列表元素

列表是指可以包含多个简单类型值的 XML 元素，例如“Monday Tuesday Wednesday”。

数据集成服务可以返回列表作为字符串值。 SOAP 消息中包含列表时，无法将项目从列表映射到独立的输出行。 如果需要在映射中隔离，可以配置一个表达式转换以隔离列表中的元素。

联合元素

联合元素是一种由多种类型联合组成的简单类型。

以下文本显示了由两种简单类型（size_no 和 size_string）联合组成的元素 Size：

```
<xs:element name="Size">
  <xs:simpleType>
    <xs:union memberTypes="size_no size_string" />
  </xs:simpleType>
</xs:element>
```

要将 size 映射到输出端口，请为 size 创建一个端口。 将输出端口配置为字符串。 可以在映射中配置另一个转换，以将数据转换为其他类型。

第 49 章

生成 Web 服务 SOAP 消息

本章包括以下主题：

生成 Web 服务 SOAP 消息概览

数据集成服务在生成 SOAP 消息时从几组输入数据生成 XML 数据。 创建 Web 服务使用者转换、Web 服务输出转换或故障转换时，您将配置要映射到 SOAP 消息层次结构的输入端口。

要配置转换以生成 SOAP 消息，请创建几组输入端口，然后将每个组映射到 SOAP 消息层次结构中的某个组。 WSDL 或架构定义 SOAP 消息的结构。

您可以在非规范化输入数据的 SOAP 消息中配置几组数据。 还可以将多次出现的输入数据转换为 SOAP 消息中多次出现的节点。

可以将数据映射到 SOAP 消息中的派生类型、anyType 元素或置换组。 定义转换时，必须选择可以出现在 SOAP 消息中的类型。 所选类型决定需要创建的输入端口。

在 Developer 工具中查看 SOAP 消息层次结构时，该层次结构中包含键。 这些键不会出现在 SOAP 消息中。 数据集成服务使用键定义 SOAP 消息中各个组之间的父子关系。 要配置键值，请将输入数据映射到 SOAP 消息中的键。

相关主题：

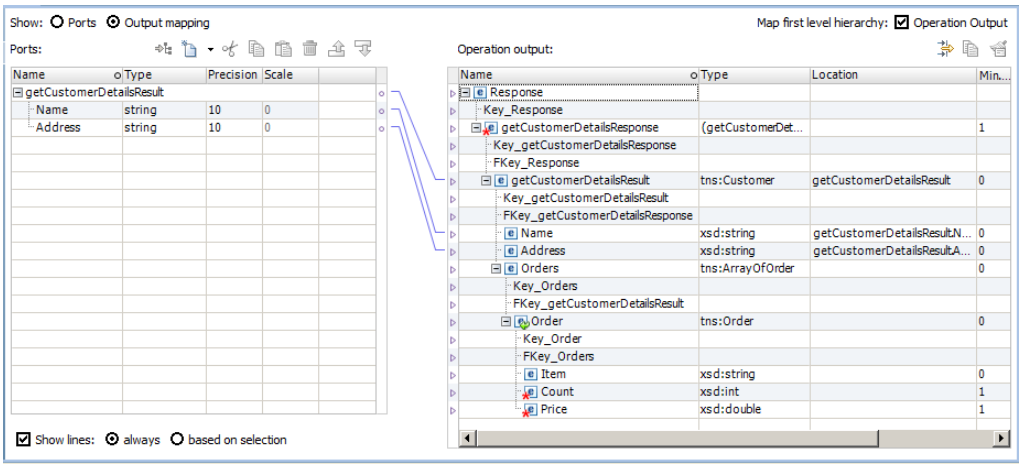
- “Web 服务使用者转换输入映射” 页面上 577

转换用户界面

Web 服务输出转换、故障转换和 Web 服务使用者转换中包含用于配置 SOAP 消息的用户界面。

要配置转换以生成 SOAP 消息，请在与 SOAP 消息层次结构相似的结构中创建输入端口。 WSDL 或架构决定层次结构的结构。 将每个输入端口映射到 SOAP 消息中的节点。

下图显示了输入端口与 Web 服务输出转换中的 SOAP 消息节点之间的映射：



输入端口区域

在**输入端口**区域中创建几组输入端口。 将需要映射的 SOAP 消息层次结构中每个级别的输入端口包括在内。

必须创建用于接收数据的响应或请求输入组和子组。

创建输入端口组时，请在每个父组中定义一个主键。 在每个子组中定义一个外键。 外键将子组与父组关联。

除非要在 WSDL 根级别传递数据，否则不需要为响应级别或 WSDL 根级别定义键。 例如，根级别可能包含 HTTP 表头。

您可能会创建与以下客户的组和顺序相似的端口组：

```
Response
   Response_Key

   Customer_Details_Root
      Key_Cust_Det
      FK_Response_Key

      Customer
        Customer_ID
        FK_Cust_Det
        Name
        Address

          Orders
            Order_Num
            FK_Cust_ID
```

```
Order_Items
  Order_Num
  Item
  Count
  Price
```

操作区域

操作区域显示 SOAP 消息层次结构中由 WSDL 或架构定义的元素。 SOAP 消息不需要包含 WSDL 或架构中的所有元素。 该消息中包含您从输入端口映射的数据。

多次出现的节点和复杂节点定义 SOAP 消息结构中的层次结构级别。 Developer 工具将键添加到这些级别，以在各个级别之间创建父子关系。 层次结构中的所有级别（叶级别除外）都有一个主键。 每个子级别都有一个指向父级别的外键。 SOAP 消息层次结构中出现的键不会出现在 SOAP 消息实例中。 数据集成服务要求在生成 SOAP 消息时键中的值与数据的级别相关联。

位置列中包含组名称和输入端口，该端口中包含 SOAP 消息中的元素的数据。 将输入端口映射到节点之前，**位置**列始终为空。

在上图中，SOAP 消息包含客户详细信息和顺序的单个实例。 “顺序”组包含多次出现的元素（名为 Order）。 SOAP 消息层次结构中包含具有关联键的以下级别：

```
Response
   GetCustomerDetailsResponse
      GetCustomerDetailsResult
         Orders
            Order
```

响应级别表示响应消息的根。 数据集成服务要求此级别将表头连接到 SOAP 消息。

GetCustomerDetailsResponse 级别是消息的根。

端口和层次结构级别关系

将输入端口映射到 SOAP 消息层次结构时，需要维护输入组与 SOAP 消息层次结构级别之间的关系。 例如，您可能有两个输入组，即“部门”和“员工”。

“部门”输入组接收以下各行：

Dept_num	名称	位置
101	HR	纽约
102	产品	加利福尼亚

“员工”输入组接收以下各行：

Dept_num	员工
101	Alice
101	Bob
102	Carol
102	Dave

映射“员工”组中的部门数量作为外键，用于在“部门”与“员工”组之间建立关系。 部门数量在部门层次结构级别出现，不在员工级别出现。

SOAP 消息中包含以下 XML 结构：

```
<department>
   <dept_num>101</dept_num>
```

```
   <name>HR</name>
   <location>New York</location>

   <employee>
      <name>Alice</name>
   </employee>

   <employee>
      <name>Bob</name>
   </employee>
</department>

<department>
   <dept_num>102</dept_num>
   <name>Product</name>
   <location>California</location>

   <employee>
      <name>Carol</name>
   </employee>

   <employee>
      <name>Dave</name>
   </employee>

</department>
```

键

SOAP 消息层次结构包括键。 数据集成服务需要键值才能在 SOAP 消息中构建 XML 层次结构。

必须将输入端口数据映射到 SOAP 消息层次结构中的键。 将数据映射到要提供数据的每个级别中的键。 具有多次出现的节点时，您需要将该节点与父节点相关联。

键出现在 SOAP 消息中，不带类型。 映射到键的任何端口都必须为 String、Integer 或 Bigint 数据类型。父组中的主键和每个子组中的外键必须具有相同的数据类型、精度和小数位数。 可以将所生成的键映射到 SOAP 消息键。

可以将端口映射到同一层次结构级别中的节点和键。 例如，您将 Employee_ID 映射到 SOAP 消息中的节点，然后将其映射到“员工”级别中的键。

如果层次结构中的两个组节点具有父子关系，请完成以下任务：

- 将端口映射到父节点组中的主键。
- 将端口映射到子节点组中的外键。

还必须将主键映射到输入端口，以删除带有空主键或带有重复主键的各个行。

可以通过将多个端口映射到同一个键，在 SOAP 消息中创建一个复合键。需要对数据进行非规范化和维护部分多次出现的值组合的唯一键时，请使用复合键。可以创建包含字符串值、长整型值或整数值的复合键。

注意: 可以在操作映射中包含表达式转换以生成键值。

复合键示例

从以下端口组配置唯一的 division-department 键：

```
Company
   Company_Num
   Company_Name

   Division
      Company_Num
```

```
Divison_Num
Division_Name

Department
   Division_Num
   Dept_Num
   Dept_Name
   Location
```

Dept_Num 在某个分支机构中是唯一的，但在公司下属的所有分支机构中不唯一。

您可能会配置一个包含分支机构和部门信息的“部门”组。配置分支机构编号和部门编号作为复合键的一部分：

```
Department
   Division_Num + Dept_Num (key)
   Dept_Name
   Location
```

映射端口的顺序决定键值。

映射端口

创建输入端口后，将每个输入端口映射到 SOAP 消息层次结构。 端口的位置在**操作**区域中的节点旁边显示。

可以将端口映射到以下类型的节点：

基本节点

没有子元素或子属性且不可见的简单元素或属性。

多次出现的基本节点

在层次结构中的同一位置多次出现的简单元素或属性。

复杂节点

包含其他元素的元素。

如果父节点没有位置，则会接收输入组名称作为位置。 父节点有位置时，层次结构级别中的每个节点都必须具有同一位置中的输出位置。

可以将输入组名称映射到层次结构级别中的父节点。 Developer 工具将更新层次结构中父节点的位置字段。 Developer 工具不更新属于层次结构中的组的子节点。 将输入端口映射到子节点时，每个输入端口位置都必须与位置父节点的位置相同。

将输入组映射到层次结构级别后，可以稍后进行更改。 可以单击**清除**，也可以删除“端口”与“操作”区域之间的行。 要删除行，请拖动行的指针以选择行。 单击**删除**。

映射端口

将端口映射到 SOAP 消息中的节点时，Developer 工具将提供不同的结果，具体取决于端口要映射到的节点类型。

下表介绍了将单个端口映射到**操作**区域中的不同目标节点时产生的结果：

目标节点	结果
基本节点	将单个端口映射到某个节点，但父节点没有位置时，该节点将接收端口的位置。 父节点位置将接收单个端口的输入组的位置。 将单个端口映射到某个节点，并且父节点已有位置时，您可以更改父节点的位置并清除同一级别中其他子节点的位置。 层次结构级别位置将更改为端口的组名称。
多次出现的基本节点或多次出现的基本节点的主键	将单个端口映射到多次出现的基本节点时，Developer 工具会将基本节点的位置设置为所选端口的组。
复杂节点	将单个端口映射到复杂节点时，Developer 工具会将复杂节点的位置设置为包含该端口的组的位置。 Developer 工具将提示您输入要向其分配端口的单次出现的基本节点。 如果单次出现的基本节点有位置，您将无法映射复杂节点。

映射组

将输入组映射到 SOAP 消息中的节点时，Developer 工具将提供不同的结果，具体取决于组要映射到的节点类型。

下表介绍了将组映射到**操作**区域中的某个节点时产生的结果：

目标节点	结果
基本节点	不能将组映射到基本节点。
多次出现的基本节点	系统将提示您选择输入组中的端口，以更新节点和主键的位置。
多次出现的复杂节点	Developer 工具会将复杂节点的位置设置为组的位置。

映射多个端口

将多个端口映射到 SOAP 消息中的节点时，Developer 工具将提供不同的结果，具体取决于端口要映射到的节点类型。 如果从相同组进行映射，则可以同时映射多个端口。

下表介绍了将多个端口映射到节点时产生的节点结果：

目标节点	结果
单个基本节点	将多个端口映射到单个节点时，请在**操作**区域中更新多个基本节点的位置。 如果层次结构在级别中要更新的节点不足，Developer 工具将仅对可用节点映射端口。

目标节点	结果
多次出现的基本节点	将多个端口映射到多次出现的基本节点时，端口将转换为多次出现的节点。 Developer 工具将根据映射的端口数量创建节点的实例。 此时将显示一条消息，指出计划的端口数量。
多次出现的复杂节点	将多个端口映射到一个复杂节点时，必须选择要更新的单次出现的基本节点。 将端口转换为多次出现的节点。 Developer 工具将根据映射的端口数量创建节点的实例。

转换多次出现的端口

可以将多个输入端口映射到 SOAP 消息中一个多次出现的节点。 Developer 工具会将输入数据转换为 SOAP 消息中的多个节点。

要更改要转换的元素数量，请选择**映射选项**对话框中的**替换现有转换**。

如果您从**端口**区域中删除了转换的其中一个端口实例，Developer 工具将从**操作**区域中删除所有实例。

转换示例

输入组可能具有以下行：

序号	名称	位置	emp_name1	emp_name2	emp_name3
101	HR	纽约	Alice	Tom	Bob
102	产品	加利福尼亚	Carol	TIm	Dave

每个行都包含一个部门编号和三个员工姓名。

“员工”是 SOAP 消息层次结构中多次出现的节点。 可以将“员工”的所有实例从输入行映射到 SOAP 消息层次结构。 选择出现的所有员工。 单击**映射**。 **映射选项**对话框将提示您从列表中选择一个节点。

Developer 工具将更改“员工”节点，以包含 SOAP 消息层次结构中的多个姓名节点：

```
Department
   num
   name
   location
   Employee     (unbounded)
       emp_name1
       emp_name2
       emp_name3
```

SOAP 消息返回以下层次结构：

```
<department>
   <num>101</num>
   <name>HR</name>
   <location>New York</location>
   <employee>
      <emp_name>Alice</name>
   </employee>
   <employee>
      <emp_name>Tom</name>
   </employee>
   <employee>
      <emp_name>Bob</name>
   </employee>
</department>

<department>
   <num>102</num>
   <name>Product</name>
```

```
<location>California</location>
<employee>
   <emp_name>Carol</name>
</employee>
<employee>
   <emp_name>Tim</name>
</employee>
<employee>
   <emp_name>Dave</name>
</employee>
</department>
```

映射非规范化数据

您可以映射非规范化数据并将其传递到 SOAP 消息中的规范化节点。

映射非规范化数据时，您将数据从一个输入组传递到 SOAP 消息层次结构中的多个节点。 您可以在 SOAP 消息中创建类似于以下类型的关系的组关系：

线性节点关系

节点 A 是节点 B 的父节点。节点 B 是节点 C 的父节点。节点 C 是节点 D 的父节点。

层次结构节点关系

节点 A 是节点 B 的父节点。节点 A 同时是节点 C 的父节点。节点 B 与节点 C 无关。

下表显示了包含非规范化分支机构和部门数据的输入行：

分支机构	Dept_Num	Dept_Name	电话	Employee_Num	Employee_Name
01	100	Accounting	3580	2110	Amir
01	100	Accounting	3580	2113	Robert
01	101	工程部	3582	2114	Stan
01	101	工程部	3582	2115	Jim
02	102	设施部	3583	2116	Jose

输入数据包含唯一的员工编号和姓名。 对于相同部门和分支机构中的每名员工，分支机构和部门数据都会重复。

线性组关系

配置端口时，可以为分支机构、部门和员工分别配置一个组。 分支结构是部门的父节点，部门是员工的父节点。 您可以按照以下线性结构配置组：

```
Division
   Division_Key
   Division_Num
   Division Name

      Department
        Department_Key
        Division_FKey
        Dept_Num
        Dept_Name
        Phone

        Employee
           Department_Fkey
           Employee_Num
           Employee_Name
```

尽管 Division_Num 和 Dept_Num 在输入数据中重复出现，但 SOAP 消息中包含唯一的分支机构和部门实例。在“分支机构”组中定义 Division_Num 作为主键。 在“部门”组中定义 Dept_Num 作为主键。

层次结构组关系

您可以创建一个组层次机构，其中包含“分支机构”父组以及“部门”和“员工”子组。 “部门”与“员工”之间不存在主键-外键关系。 “部门”和“员工”是“分支机构”的子组。 您可以按照以下结构配置组：

```
Division
   Division_Key
   Division_Num
   Division_Name

   Department
      Division_FKey
      Dept_Num
      Dept_Name

   Employee
      Division_FKey
      Employee_Num
      Employee_Name
```

派生类型和元素置换

可以将输入端口映射到 SOAP 消息中的派生复杂类型、anyType 元素和置换组。 SOAP 消息可以包含基本类型和派生类型的元素。

在类型关系中，基本类型是指派生了其他类型的类型。 派生类型继承基本类型中的元素。 扩展复杂类型是指继承了基本类型中的元素并包含其他元素的派生类型。 受限复杂类型是指限制基本类型中的部分元素的派生类型。

生成派生类型

如果 WSDL 或架构中包括派生类型，您必须选择要包含在 SOAP 消息中的类型。

例如，WSDL 定义了一个基本类型 AddressType。 WSDL 中还包含属于派生 AddressTypes 的 USAddressType 和 UKAddressType。

每种类型都包含以下元素：

- AddressType 返回一个行：street, city
- USAddressType（扩展 AddressType）：state, zipCode
- UKAddressType（扩展 AddressType）：postalCode, country

在“操作”区域中选择 USAddressType 时，Developer 工具将在 SOAP 消息中为 USAddressType 元素创建一个组。 该组包括 USAddress 的基本地址中的街道和城市以及州和 zipCode。扩展了基本类型的派生类型始终包括基本类型中的元素。

如果您为 SOAP 消息选择了所有可用的派生类型，Developer 工具将在 SOAP 层次结构中创建类似于以下组的组：

```
Address
  Address: Address
    Street
    City

  Address:USAddressType
     Street
     City
     State
```

```
      ZipCode

   Address: UKAddressType
      Street
      City
      PostalCode
      Country
```

您需要为 Address、USAddress 和 UKAddress 定义输入端口组。

生成 anyType 元素和属性

部分架构元素和属性允许在 SOAP 消息中使用任何类型的数据。

anytype 元素表示选择所有全局已知类型。 将端口映射到 SOAP 消息中的 anyType 元素之前，请选择可用的复杂类型或 xs:string。 如果 WSDL 或架构中不包含复杂类型，Developer 工具会将 anyType 元素类型替换为 xs:string。

要在操作区域中选择元素类型，请在 anyType 元素的**类型**列中单击**选择**。 此时将显示一列可用的复杂类型和 xs:string。

以下元素和属性允许使用任意类型的数据：

anyType 元素

允许关联 XML 文件中的元素属于任意数据类型。

anySimpleType 元素

允许关联 XML 文件中的元素属于任意 simpleType。

ANY 内容元素

允许元素属于在架构中定义的任意全局元素。

anyAttribute 属性

允许元素属于已在架构中定义的任意属性。

生成置换组

使用置换组将一个元素替换为 SOAP 消息中的另一个元素。 置换组的工作方式与派生类型相似，不同的是元素定义中包含一个置换组名称。

例如，您可能有一个基本类型 Address 以及派生类型 USAddress 和 UKAddress：

```
xs:element name="Address" type="xs:string"/>
<xs:element name="USAddress" substitutionGroup="Address"/>
<xs:element name="UKAddress" substitutionGroup="Address"/>
```

配置 SOAP 消息层次结构时，可以为 SOAP 消息中的 Address 选择要替换的元素。

生成 SOAP 消息中的 XML 构造

WSDL 或架构可以包含选项、列表或联合元素。 Web 服务转换可以生成包含这些元素的 SOAP 消息。

选项元素

选项元素将子元素限制为 <choice> 声明中的一个元素。

要将端口映射到包含选项元素的 SOAP 消息，请创建一个包含选项构造中的所有元素的输入组。 例如，项目说明可以是维度或权重：

```
item: description, choice {dimension, weight}
```

如果说明是维度，则说明属于包含长度、宽度和高度的复杂类型。

如果说明是权重，则说明属于简单字符串类型。

输入数据具有以下列和行：

说明	长度	宽度	高度	权重
框	20cm	18cm	15cm	NULL
coffee	NULL	NULL	NULL	500g

SOAP 消息包含的项目组中包含维度或权重说明：

```
Item
   Description
       Dimension
          Length
          Width
          Height
       Weight
```

输入数据中的 NULL 值成为 XML 输出中缺失的元素。

SOAP 消息包含以下数据：

```
<item>
   <desc>box</desc>
      <dimension>
        <length>20cm</length>
        <width>18cm</width>
        <height>15cm</height>
      </dimension>
</item>

<item>
   <desc>coffee</desc>
      <weight>500g</weight>
</item>
```

列表元素

列表是指在相同元素或属性中包含多个简单类型值的 XML 元素。 如果输入数据中的列表的表示形式为合并的数据字符串，则数据集成服务可以处理该列表。

如果该列表中的每个项目都是一个独立的元素，例如 ClassDates1、ClassDates2 和 ClassDates3，则数据集成服务将无法以列表方式处理这些项目。 如果需要在 SOAP 消息中返回一个列表，可以使用表达式转换将这些项目合并成一个字符串。

以下输入行中包含一个名为 ClassDates 的列表元素，其中包含周中的某一天：

CourseID	名称	ClassDates
Math 1	Beginning Algebra	Mon Wed Fri
History 1	World History	Tue Thu

数据集成服务可以返回具有以下 XML 结构的 SOAP 消息：

```
<class>
   <courseId>Math 1</courseId>
```

```
   <name>Beginning Algebra</name>
   <classDates>Mon Wed Fri</classDates>
</class>
<class>
   <courseId>History 1</courseId>
   <name>World History</name>
   <classDates>Tue Thu</classDates>
</class>
```

联合元素

联合元素是一种由多种类型联合组成的简单类型。 SOAP 消息中包含联合元素时，必须映射包含字符串中的数据的单个输入端口。

例如，SOAP 消息中包含一个名为 size 的元素。大小由整数和字符串组成：

```
<xs:element name="size">
  <xs:simpleType>
    <xs:union memberTypes="size_no size_string" />
  </xs:simpleType>
</xs:element>
```

输入行中包含具有说明和大小的项目。项目可以具有数值大小，例如 42。或者，项目可以具有字符串值类型的大小，例如 large、medium 或 small。

下表显示了具有数值大小和字符串大小的输入行：

说明	大小
shoes	42
shirt	large

为项目大小创建一个端口。 以字符串形式映射端口。 SOAP 消息中包含以下元素：

```
<item>
   <desc>shoes</desc>
   <size>42</size>
</item>

<item>
   <desc>shirt</desc>
   <size>large</size>
</item>
```

第 50 章

加权平均值转换

本章包括以下主题：

加权平均值转换概览

加权平均值转换是一种被动转换，用于读取多个匹配操作的匹配得分并生成单个匹配得分。

可以将数值权重应用于进入加权平均值转换的每个得分。 权重是介于零和 1 之间的值。 可以编辑应用于每个输入得分的权重，以增加或减少其对输出得分的贡献。 在重复分析中应用用于反映每个数据列的相对重要性的权重。

将比较转换添加到映射或 Mapplet 时使用加权平均值转换。

注意: 也可以在匹配转换中分配权重。 使用匹配转换可配置匹配策略以及分配单个转换中的权重。 您可以在匹配转换中嵌入匹配 Mapplet。

配置加权平均值转换

使用加权平均值转换可以调整映射针对一系列匹配分析操作生成的整体匹配得分。 编辑每个输入端口的相关权重以反映为源数据集定义的数据比较优先级。 加权平均值转换的每个输入端口表示来自比较转换策略的匹配得分输出。

以下步骤说明了在 Mapplet 或使用比较转换的映射中配置不可重用的加权平均值转换的流程。

1. 打开匹配分析 Mapplet 或映射，然后在比较转换下游添加加权平均值转换。
2. 将来自比较转换的得分输出连接到加权平均值输入端口。

   为 Mapplet 或映射中的其他比较转换重复执行该步骤。
3. 在加权平均值转换上选择**端口**选项卡。
4. 双击每个输入的**权重**字段，然后输入一个介于 0.001 和 1 之间的权重值。与转换中的其他输入相比较时，权重值应反映输入得分的相对重要性。
5. 保存 Mapplet 或映射。

加权匹配得分示例

创建匹配分析映射以确定客户数据库中重复的客户名称数。 添加两个比较转换以生成数据集中邮政编码和姓氏列的匹配得分。

许多记录具有匹配的邮政编码，但是具有匹配姓氏的记录要少得多。 在计算这些匹配得分的平均值时，需要强调更唯一匹配的重要性。

要强调姓氏匹配得分的重要性，请为姓氏匹配得分应用更高权重。

例如，将姓氏得分输入的**权重**值设置为 0.8，将邮政编码得分输入的**权重**值设置为 0.4。

加权平均值转换高级属性

配置属性以确定数据集成服务如何处理加权平均值转换的数据。

可以配置日志的跟踪级别。

在**高级**选项卡上配置以下属性：

跟踪级别

此转换的日志中显示的详细信息量。 可以选择精简、普通、详细初始化或详细数据。 默认值为“普通”。

非本地环境下的加权平均值转换

非本地环境中的加权平均值转换处理取决于运行转换的引擎。

请注意以下非本地运行时引擎的支持情况：

- Blaze 引擎。无限支持。
- Spark 引擎。在批处理映射中无限支持。在流映射中不受支持。
- Databricks Spark 引擎。无限支持。

第 51 章

窗口转换

如果要将流式传输数据累积到数据组中，然后处理数据集，可以使用窗口转换。窗口转换是一种被动转换。

从未绑定的源读取数据时，您可能需要将数据累积到已绑定的数据组中以进一步进行处理。要将已绑定的间隔引入到未绑定的数据，可以使用窗口转换。

配置窗口转换时，请按时间定义窗口类型和数据边界。要指定数据边界，请配置窗口大小和窗口滑动间隔。窗口大小定义将数据累积到数据组的时间间隔。滑动间隔定义进一步处理已累积的数据组前的时间间隔。水印延迟定义要累积到数据组中的延迟事件的阈值时间。

只能在 Spark 引擎上为流映射运行窗口转换。

有关窗口转换的详细信息，请参阅《*Data Engineering Streaming 用户指南*》。

第 52 章

写入转换

本章包括以下主题：

写入转换概览

写入转换是一种被动转换，映射使用写入转换来将数据写入到目标。写入转换不可重用。

可以从物理数据对象、逻辑数据对象或参数创建写入转换。如果要基于从 PowerExchange 适配器源导入的物理数据对象创建写入转换，映射编辑器可能会提示您必须指定写入操作才能基于该数据对象创建写入转换。

可为写入转换配置的属性取决于用于创建该转换的数据对象的类型。

写入转换可以表示动态目标。可将写入转换配置为动态更新其端口、元数据和其他属性。有关配置动态目标的信息，请参阅*《Informatica Developer 映射指南》*。

写入转换属性

创建写入转换后，可为其配置属性。

在转换的**属性**视图中的选项卡上配置写入转换属性。您可以配置的选项卡取决于写入转换所表示的目标的类型。

下表介绍了每个属性选项卡并标识了对应选项卡所针对的目标类型：

属性选项卡	说明	目标类型
常规	指定转换属性和行为。 对于关系和自定义数据对象源，将转换输入端口与源同步。	全部
数据对象	指定转换数据源。 对于关系和自定义数据对象源，在运行时从数据源获取数据对象列。	平面文件 关系 自定义数据对象
格式	平面文件数据源的输入设置	平面文件

属性选项卡	说明	目标类型
端口	按关联的数据对象或映射流设置端口定义。	全部
运行时	在运行时将数据写入目标时数据集成服务使用的属性，如拒绝文件发送到的位置。 对于平面文件目标，拒绝文件名和目录。	平面文件 关系
数据对象参数	查看数据对象参数。为映射配置参数值，或将参数绑定到映射参数。	平面文件 自定义数据对象 逻辑数据对象
运行时链接	创建新的运行时链接并查看链接属性。	全部
高级	设置跟踪级别和行顺序。 对于关系或 Hive 目标，设置目标架构策略。	平面文件 关系 自定义数据对象 逻辑数据对象

常规属性

可以配置写入转换的名称和说明。 还可以配置以下属性：

列元数据更改时：

可用于关系目标和自定义目标。 选择以下选项之一：

- 同步输入端口。 Developer tool 使用模型存储库存储的数据对象的元数据更改来更新写入转换输入端口。
- 不同步。 Developer tool 不显示数据对象中的元数据更改。

默认值为同步输入端口选项。

物理数据对象

可用于平面文件和自定义目标。 用于创建转换的对象。

您可以选择数据对象名称并配置其属性。

数据对象属性

在“数据对象”选项卡上，可以指定或更改写入转换目标并使关系和自定义数据对象目标动态化。

可以配置以下属性：

指定依据

要指定写入转换的目标列和元数据，请选择下列选项之一：

- 值。写入转换使用关联的数据对象指定目标列和元数据。
- 参数。写入转换使用参数指定目标列和元数据。

默认为“值”选项。

数据对象

如果是从现有数据对象创建的写入转换，则该字段会显示对象的名称。单击**浏览**可将数据对象更改为与写入转换相关联。

参数

选择或创建要与写入转换相关联的参数。

运行时，从数据源获取数据对象列

启用此选项时，数据集成服务会将元数据和数据定义更改从目标表提取到写入转换。

要预览数据集成服务如何提取元数据和数据定义更改，使用已解析的参数查看映射。

端口属性

在“端口”选项卡上可以配置以下属性：

列定义依据

选择以下选项之一以定义写入转换列：

- 关联的数据对象。 使用“数据对象”选项卡中数据对象的列名称、元数据和其他属性。
- 映射流。 映射从其上游对象提取列名称、元数据和其他属性。

默认为“关联的数据对象”选项。

列资源属性

可用于平面文件和自定义目标。 每个列的资源是列从中提取名称、元数据和其他属性的数据对象。 选择资源名称可更改资源属性。

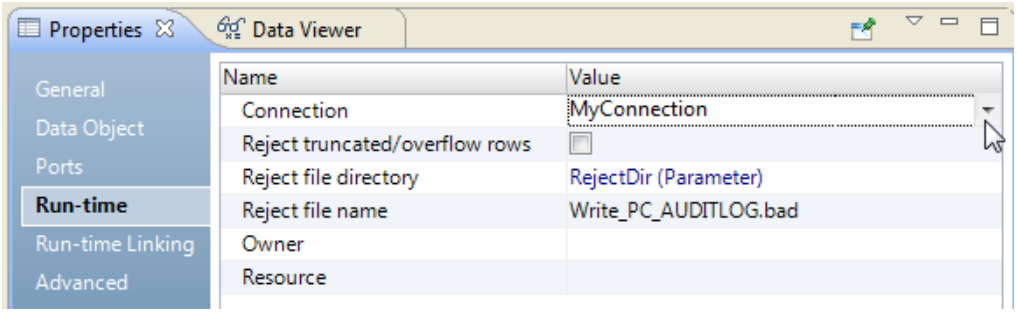
运行时属性

可以在“运行时”选项卡上配置以下“写入”事务属性：

连接

可用于关系目标。 转换所使用的连接。 单击字段右侧可更改连接。

下图显示了要单击的下拉按钮的位置：



拒绝截断/溢出的行

可用于关系目标和自定义目标。

Developer tool 允许您通过在端口之间传递数据来转换数据。 有时，转换会导致在包含字符的列中发生数值数据溢出或字符串截断。 例如，将 Decimal (28, 2) 端口的数据传递到 Decimal (19, 2) 端口会导致数值溢出。 同理，如果将 String(28) 端口的数据传递到 String(10) 端口，数据集成服务会将字符串截断为 10 个字符。

如果转换导致溢出，默认情况下数据集成服务将跳过该行。 数据集成服务不会将数据写入到拒绝文件。 对于字符串，数据集成服务将截断该字符串并将其传递到下一个转换。

选择此选项可以在会话拒绝文件中包含上一个转换和目标之间所有截断和溢出的数据。 数据集成服务会将所有截断的行和任何溢出的行发送到会话拒绝文件或行错误日志，具体视您对会话的配置而异。

拒绝文件目录

拒绝文件所在的目录。 默认为 RejectDir 系统参数。

拒绝文件名

拒绝文件的文件名。默认为 <output_file_name>.bad。

如果多个分区写入平面文件目标，则每个分区都会写入单独的一个名为 <output_file_name><partition_number>.bad 的拒绝文件。

运行时链接属性

在**运行时链接**选项卡上创建和配置运行时链接。 运行时链接是转换之间使用参数和/或链接策略来确定在运行时链接哪些端口的组到组链接。 运行时链接在映射编辑器中以粗线显示。

在以下情况下创建和配置写入转换的运行时链接：

- 写入转换中的目标数据对象使用参数。
- 上游转换中的端口可在运行时更改。

注意: 根据映射流定义目标列时，请勿创建写入转换的运行时链接。

可以在**运行时链接**选项卡上执行以下任务：

创建运行时链接。

在**链接**区域中，单击**新建**按钮，然后在“新链接”对话框中选择在运行时要链接到写入转换的端口的来源转换。

配置运行时链接属性。

在**链接属性**区域中，配置以下运行时链接属性：

参数

端口名称可在映射运行之间更改且您知道端口名称值时，选择此选项。 使用“输入链接集”类型的参数按名称值连接映射运行之间的端口。 “输入链接集”映射参数的语法包含以逗号分隔的端口对：

```
Afield1->Bfield2, Afield3->Bfield4 .
```

链接策略

如果要自动链接具有匹配名称的端口，请选择此选项。 例如，当两个映射对象都包含名为 SALARY 的端口时，数据集成服务会将其链接起来。 您可以忽略端口名称中的前缀和后缀。

在运行时，数据集成服务按下列顺序在端口之间建立链接并对其进行解析：

- 您在映射编辑器中手动创建的链接。
- 基于您为运行时链接所配置的参数的链接。
- 基于您为运行时链接所配置的链接策略的链接。

有关运行时链接的详细信息，请参阅*《Informatica Developer 映射指南》*。

高级属性

配置高级属性以确定数据集成服务如何为写入转换处理数据。

在**高级**选项卡上配置以下属性：

跟踪级别

控制映射日志文件中的详细信息量。

目标负载类型

目标加载的类型。选择“正常”或“批量”。可以为关系资源或自定义的数据对象设置目标加载类型。

如果选择“正常”，数据集成服务将正常加载目标。加载到 DB2、Sybase、Oracle 或 Microsoft SQL Server 时可以选择“批量”。如果为其他数据库类型指定“批量”，数据集成服务将还原为正常加载。批量加载可以提高映射性能，但由于不进行数据库日志记录，会限制恢复功能。使用批量加载写入 Oracle 目标时，可以通过在 Oracle 数据库中禁用约束优化性能。

如果映射包含更新策略转换，选择“正常”模式。如果选择“正常”，并且 Microsoft SQL Server 目标名称中含有空格，则应在连接对象中配置以下环境 SQL：

```
SET QUOTED_IDENTIFIER ON
```

更新替代

替代目标的默认 UPDATE 语句。

删除

删除所有标记为要删除的行。

默认情况下启用。

注意: Databricks Spark 引擎在写入 Delta Lake 目标时会忽略此属性。要删除行，请使用“更新策略”转换。

插入

插入所有标记为要插入的行。

默认情况下启用。

注意: Databricks Spark 引擎在写入 Delta Lake 目标时会忽略此属性。

目标架构策略

关系或 Hive 目标表的目标架构策略的类型。

可以选择以下目标架构策略之一：

- **RETAIN - 保留现有目标架构。**数据集成服务保留现有的目标架构。
- **CREATE - 在运行时创建或替换表。**数据集成服务在运行时丢弃目标表，然后将其替换为基于您标识的目标表的表。
- **分配参数。**可以分配一个参数来表示目标架构策略的值，然后在运行时更改该参数。

用于创建或替换的 DDL 查询

根据您定义的 DDL 查询在运行时创建或替换目标表。选择 **CREATE - 在运行时创建或替换表**目标架构策略选项时适用。

截断目标表

在目标表加载数据前将其截断。

默认情况下启用。

截断目标分区

在内部或外部分区 Hive 目标加载数据之前将其截断。选择此选项之前，必须先选择**截断目标表**。

默认为禁用。

更新策略

更新现有行的策略。可以选择以下策略之一：

- 更新为更新。数据集成服务将更新所有标记要更新的行。

- 更新为插入。数据集成服务将插入所有标记要更新的行。还必须选择**插入**目标选项。

  注意: 如果目标是 Databricks Delta Lake，则您无需选择**插入**。
- 首选更新，否则插入。数据集成服务将更新标记要更新的行（如果这些行存在于目标中），然后插入其余任何标记要插入的行。还必须选择**插入**目标选项。

  注意: 如果目标是 Databricks Delta Lake，则您无需选择**插入**。

PreSQL

数据集成服务在读取源之前对目标数据库运行的 SQL 命令。

Developer tool 不会验证 SQL。

PostSQL

数据集成服务在写入目标后对目标数据库运行的 SQL 命令。

Developer tool 不会验证 SQL。

维持行顺序

保持目标输入数据的行顺序。如果数据集成服务不应执行任何可能改变行顺序的优化，请选择此选项。

数据集成服务执行优化时，可能会失去之前在映射中建立的行顺序。您可以使用已排序的平面文件源、已排序的关系源或排序器转换在映射中建立行顺序。如果将某个目标配置为维持行顺序，则数据集成服务不会对此目标执行优化。

约束

用于表级引用完整性约束的 SQL 语句。仅适用于关系目标。

创建写入转换

创建写入转换时，您将根据创建转换所基于的资源来选择下列方法之一：

从模型存储库中的数据对象创建转换。

如果要使写入转换元数据基于数据对象，则使用此方法。

要在映射编辑器中创建写入转换，请使用**新建写入转换**向导，或从**对象浏览器**拖动数据对象并选择“写入”作为转换类型。

从映射对象流创建转换。

使用此方法可使写入转换中的端口基于上游映射转换中的元数据流。

使用参数创建转换。

如果希望写入转换从参数所表示的对象继承端口，则使用此方法。

使用映射中的其他转换创建转换。

如果希望写入转换具有与源转换相同的端口，则使用此方法。

从数据对象创建写入转换

可以在映射编辑器中创建和配置写入转换。

如果要为转换配置特定参数，可能需要使用此方法。 例如，您可以创建转换，然后将其配置为在运行时从数据源获取列元数据。

1. 在映射编辑器中右键单击，然后选择**添加转换**。

   此时将打开**添加转换**对话框。

2. 选择**写入**，然后单击**确定**。

   此时将打开**新建写入转换**向导。

3. 选择**物理数据对象**或**逻辑数据对象**，然后单击**浏览**。

   此时将打开**选择数据对象**对话框。

4. 选择一个数据源，然后单击**确定**。

5. 要按关联的数据对象定义写入转换，请选择**关联的数据对象**。

   写入转换端口与关联的数据对象的列相同。

6. 要在运行时使用目标文件中的更改动态更新目标对象列，请选择**在运行时，从数据源获取数据对象列**。

   映射运行时，数据集成服务将刷新写入转换的列元数据。

7. 单击**完成**。

从映射流创建写入转换

可以在映射编辑器中创建和配置写入转换。

如果要为转换配置特定参数，可能需要使用此方法。 例如，您可以创建转换，然后将其配置为根据映射流定义列以及在运行时从数据源获取列元数据。

1. 在映射编辑器中右键单击，然后选择**添加转换**。

   此时将打开**添加转换**对话框。

2. 选择**写入**，然后单击**确定**。

   此时将打开**新建写入转换**向导。

3. 选择**物理数据对象**或**逻辑数据对象**，然后单击**浏览**。

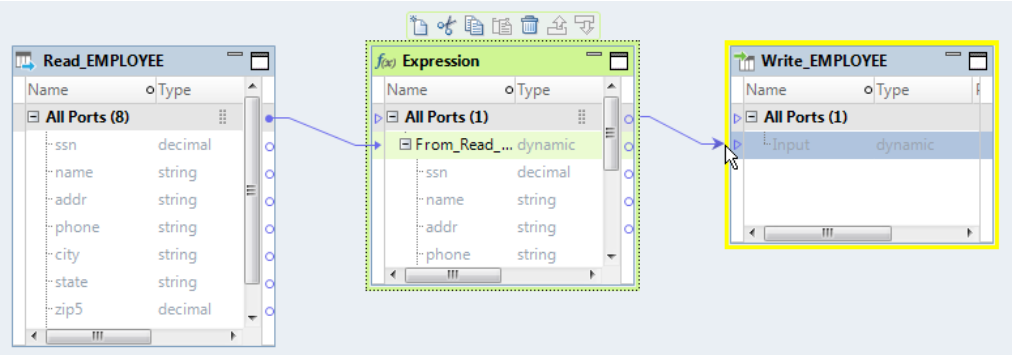
   此时将打开**选择数据对象**对话框。

4. 选择一个数据源，然后单击**确定**。

5. 选择**映射流**，然后单击**完成**。

6. 将上游“所有端口”端口中的端口拖至写入转换的**输入**端口。

下图显示了如何将上游端口连接到写入转换的**输入**端口。



写入转换将收到来自上游映射对象的列定义。

7. 要在运行时使用目标文件中的更改动态更新目标对象列，请选择**在运行时，从数据源获取数据对象列**。

   映射运行时，数据集成服务将刷新写入转换的列元数据。

从参数创建写入转换

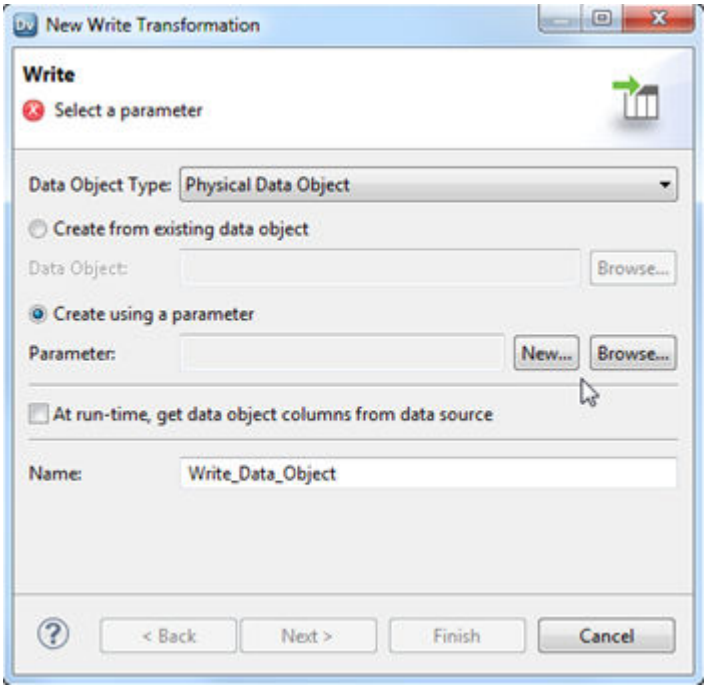
参数是可以在映射运行之间更改的常量值。

在动态映射中，可以使用参数更改源和目标。 也可以将参数用于输入规则、选择规则、运行时链接和转换属性。当您更改参数值时，数据集成服务将根据在参数中指定的值来创建或重新创建目标。

1. 在编辑器中右键单击，然后选择**添加转换**。
2. 从转换列表中选择“写入”，然后单击**确定**。

   提示: 键入转换的第一个字母以筛选列表。

   此时将打开“新建写入转换”对话框。



3. 选择**使用参数创建**。

4. 浏览选择参数，或单击**新建**创建一个新参数，然后浏览选择创建参数所基于的数据对象。
5. （可选）选择**在运行时，从数据源获取数据对象列**选项以在运行时刷新转换列。
6. （可选）单击**下一步**，选择如何定义转换列。
7. 单击**完成**。

从现有转换创建写入转换

可以在存在于映射中的转换中创建具有相同端口的写入转换。

可为复杂文件、平面文件或关系资源创建目标。如果现有转换中的端口之一包含复杂端口，您必须创建复杂文件目标，还必须按名称将端口链接起来或在运行时基于链接策略将端口链接起来。Developer tool 可以创建 Avro、Parquet、ORC 或 JSON 复杂文件目标。

1. 在编辑器中打开一个映射。
2. 在映射编辑器中右键单击某个转换，然后选择**创建目标**。

   此时将打开**创建目标**窗口。
3. 将数据对象类型选择为“复杂文件”、“平面文件”、“关系”或“其他”。
4. 如果您选择**其他**，请从该列表中选择所需的目标对象类型。
5. 选择链接类型。

   可以从以下链接类型中选择一种类型：
   按名称将端口链接起来

   写入转换中的端口与源中的端口对应，并且具有相同名称。

   基于映射流将动态端口链接起来

   写入转换包含基于映射流中的上游对象的动态端口。

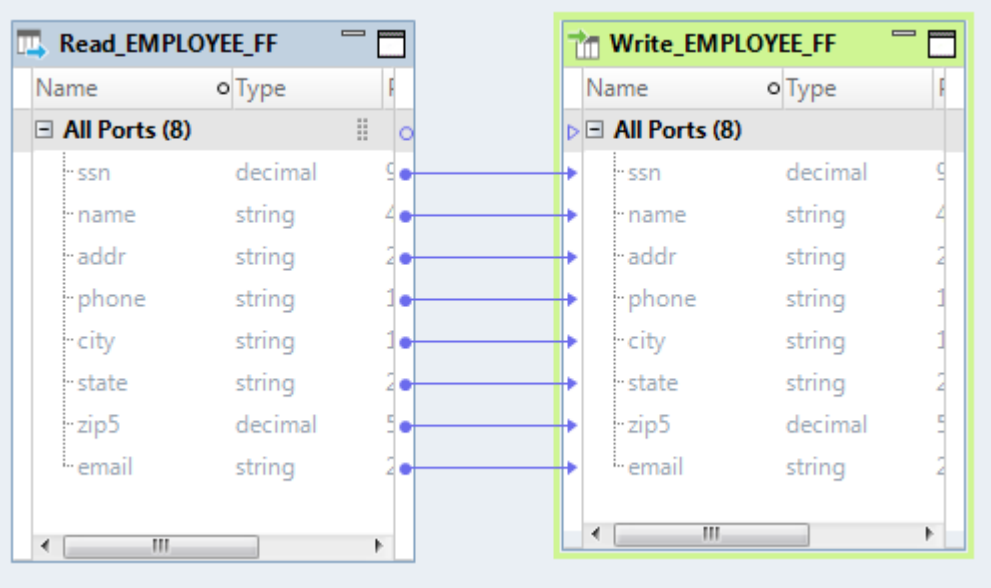
   在运行时基于链接策略将端口链接起来

   在运行时，系统会根据您在写入转换的“运行时链接”选项卡上配置的链接策略在目标中创建端口。

   有关动态端口和运行时链接配置的详细信息，请参阅*《Informatica Developer 映射指南》*。
6. 为新数据对象命名。
7. （可选）单击**浏览**，选择数据对象的位置。
8. 如果选择创建复杂文件目标，请在**资源格式**下拉框中选择 Avro 或 Parquet 作为复杂文件格式。
9. 单击**完成**。

Developer tool 将执行以下任务：

- 向映射添加写入转换。
  下图显示了从读取转换创建的写入转换：



- 将端口链接起来。
- 创建物理数据对象。
  可以配置物理数据对象属性。例如，必须为复杂文件数据对象指定 HDFS 连接。

附录 A

转换分隔符

本附录包括以下主题：

转换分隔符概览

转换分隔符指定数据字符串之间的分界线。

下表列出了转换用于解析和写入数据字符串的分隔符：

分隔符名称	分隔符号
at 符号	@
逗号	,
短划线	-
双引号	"
正斜杠	/
句点	.
哈希	#
管道符号	|
分号	;
单引号	'
空格	[空格键]
制表符	[Tab 键]
下划线	_

索引

符號

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

Y

Z