IBM Security Directory Integrator
V 7.2.0.1

入门指南

G151-1408-03

IBM Security Directory Integrator
V 7.2.0.1

入门指南

G151-1408-03

注释

在使用本资料及其支持的产品之前，请阅读第 103 页的『声明』中的常规信息。

修订版通知

注: 本版本适用于 *IBM Security Directory Integrator* **V7.2.0.1 (5724-K74)** 许可程序及所有后续发行版和修订版，直到在新版本中另有声明为止。

目录

图

关于本出版物

使用《IBM® Security Directory Integrator 快速入门指南》可以学习基本的 Security Directory Integrator 概念并了解关于设计有效数据集成解决方案的背景知识。

本文档介绍了有关 IBM Security Directory Integrator 的概念性信息并提供了示例帮助您开始使用本产品。

访问出版物和术语

请阅读 IBM Security Directory Integrator V7.2.0.1 库及可在线访问的相关出版物的描述。

本节提供:

- 『IBM Security Directory Integrator 库』 中的出版物列表。
- 指向 第 viii 页的『在线出版物』 的链接。
- 指向 第 viii 页的『IBM Terminology Web 站点』 的链接。

IBM Security Directory Integrator 库

IBM Security Directory Integrator 库中提供以下文档:

- *IBM Security Directory Integrator V7.2.0.1 Federated Directory Server 管理指南*

  包含关于使用 Federated Directory Server 控制台来设计、实施和管理数据集成解决方案的信息。也包含关于使用身份管理的跨域身份管理系统（SCIM）协议和界面的信息。

- *IBM Security Directory Integrator V7.2.0.1 快速入门指南*

  包含 IBM Security Directory Integrator 的简要教程和介绍。包含用于创建 IBM Security Directory Integrator 的交互与实际操作学习的示例。

- 《*IBM Security Directory Integrator V7.2.0.1 用户指南*》

  包含关于使用 IBM Security Directory Integrator 的信息。包含关于使用 Security Directory Integrator 设计器工具（配置编辑器）来设计解决方案或从命令行运行现成的解决方案的指示信息。还提供了关于界面、概念和 AssemblyLine 创建的信息。

- *IBM Security Directory Integrator V7.2.0.1 Installation and Administrator Guide*

  包含了关于安装、从前版本迁移、配置记录功能和 IBM Security Directory Integrator 远程服务器 API 底层安全模型的完整信息。包含关于如何部署和管理解决方案的信息。

- *IBM Security Directory Integrator V7.2.0.1 Reference Guide*

  包含关于 IBM Security Directory Integrator 的独立组件（连接器、功能组件、解析器和对象等 – AssemblyLine 的构建块）的详细信息。

- *IBM Security Directory Integrator V7.2.0.1 Problem Determination Guide*

提供关于 IBM Security Directory Integrator 工具、资源和技术的信息，这些信息可以协助确定和解决问题。

- *IBM Security Directory Integrator V7.2.0.1 Message Guide*

  提供与 IBM Security Directory Integrator 关联的所有参考、警告和错误消息的列表。

- *IBM Security Directory Integrator V7.2.0.1 Password Synchronization Plug-ins Guide*

  包括安装和配置下列五个 IBM Password Synchronization Plug-in 的完整信息: Windows Password Synchronizer、Sun Directory Server Password Synchronizer、IBM Security Directory Server Password Synchronizer、Domino® Password Synchronizer 以及 Password Synchronizer for UNIX and Linux。还提供了针对 LDAP 密码存储和 JMS 密码存储的配置指示信息。

- *IBM Security Directory Integrator V7.2.0.1发行说明*

  描述文档中未包含的关于 IBM Security Directory Integrator 的新功能和最新信息。

在线出版物

当发布产品和更新出版物时，IBM 会在以下位置发布出版物:

IBM Security Directory Integrator 库

产品文档站点（http://www-01.ibm.com/support/knowledgecenter/SSCQGF/welcome）会显示此库的欢迎页面和导航。

IBM Security Systems Documentation Central

IBM Security Systems Documentation Central 提供一个列表，该列表按字母顺序列示所有 IBM Security Systems 产品库以及指向每种产品的特定版本的联机文档的链接。

IBM 出版物中心

IBM 出版物中心站点（ http://www-05.ibm.com/e-business/linkweb/publications/servlet/pbi.wss）提供定制的搜索功能来帮助找到您需要的所有 IBM 出版物。

相关信息

以下位置提供了与 IBM Security Directory Integrator 相关的信息:

- IBM Security Directory Integrator 使用 Oracle 的 JNDI 客户机。关于 JNDI 客户机的信息，请参阅 *Java Naming and Directory Interface™ Specification* at http://download.oracle.com/javase/7/docs/technotes/guides/jndi/index.html 。
- 在以下地址可以找到可能帮助解答与 IBM Security Directory Integrator 相关的问题的信息: https://www-947.ibm.com/support/entry/myportal/over-accesspubsview/software/security_systems/tivoli_directory_integrator。

IBM Terminology Web 站点

IBM Terminology Web 站点将一个位置的产品库的术语合并到一起。您可以访问 Terminology Web 站点，地址是 http://www.ibm.com/software/globalization/terminology。

辅助功能选项

辅助功能选项帮助有身体残疾（例如行动不便或视力受限）的用户成功使用软件产品。通过此产品，您可以使用辅助技术来听取和浏览界面。您还可以使用键盘代替鼠标来操作图形用户界面的所有功能。

有关更多信息，请参阅*配置 Directory Integrator*中的辅助功能选项附录。

技术培训

有关技术培训的信息，请参阅以下 IBM Education Web 站点，地址是 http://www.ibm.com/software/tivoli/education。

支持信息

IBM 支持为代码相关问题和例程、短时段安装或使用问题提供帮助。您可以直接访问 IBM 软件支持站点，地址是 http://www.ibm.com/software/support/probsub.html。

故障诊断 提供的详细信息关于:

- 联系 IBM 支持之前要收集哪些信息。
- 联系 IBM 支持的各种方法。
- 如何使用 IBM 支持助理。
- 自己找出问题并解决问题所需的指示信息和问题确定资源。

良好的安全实践声明

IT 系统安全包含通过预防、检测和响应来自企业内部和外部的不正当访问来保护系统和信息。不正当访问可能导致信息被改变、毁坏、挪用或滥用或可能导致损坏或误用您的系统，包括用来对他人进行攻击。没有任何 IT 系统或产品应视为是完全安全的，并且没有任何产品、服务或安全措施能够完全有效地预防不正当使用或访问。IBM 系统、产品和服务设计为综合安全方法的组成部分，将必然涉及额外的操作过程，还可能要求其他系统、产品或服务达到最大有效性。IBM 不能保证任何系统、产品或服务都有免疫力，或使您的企业免于任何一方的恶意或非法行为。

第 1 章 简介

本书提供对一个简单系统的简单介绍。一点没错，*简单*一词用在这里可谓恰如其分，因为专注于解决一个复杂问题的最佳方法就是简化该问题；将问题分解为更易于管理的若干部分，然后解决这些组成部分。分而治之，各个击破。这是您在解决日常问题时本能使用的一种技巧，它同样也可用于设计跨办公室、跨企业或全球范围的信息交换。

如果急于从文件、目录、数据库或 Lotus Notes® 快速开始抽取信息并将该数据传输到别处，那么您可能希望直接跳到附录 第 81 页的『EasyETL 指南』。该功能使您利用 IBM Security Directory Integrator 的功能而不必首先学习核心概念。而是选择源和目标，然后按“运行”并观察您的数据流。如果另一方面您想更多地控制读取、过滤、充实、变换和移动数据的方式，那么继续阅读此处；您将来仍可从此处访问 第 81 页的『EasyETL 指南』，轻松阅读其中的内容。

IBM Security Directory Integrator[1] 是基于即使最复杂的集成问题也能够分解为以下三个基本部分这一前提设计和构建的:

- 通信中涉及到的系统 – 也称为*数据源*,
- 这些系统之间的*数据流*,
- 触发数据流的*事件*。

通过 IBM Security Directory Integrator，可将对集成问题的这一最基本的了解直接转换为解决方案，方法是通过不断反馈和验证以递增方式（一次增加一个流）构建该解决方案。该方法使得集成项目更易于评估和规划，有时在这方面减少的工时可用于对要实施的各个数据流进行计数和成本核算。以可运行的步骤完成任务还使您能够定期向项目干系人演示进度。

通过将数据源之间的技术差异提炼出来，IBM Security Directory Integrator 进一步加快了开发速度，从而使您能够将更多时间集中于业务需求上。

利用 Eclipse 的功能，IBM Security Directory Integrator 开发环境将既全面又可扩展。集成项目产生组件库和业务逻辑，可以快速重用这些组件库和业务逻辑来解决新的难题。因此，整个组织的各个团队可以共享 IBM Security Directory Integrator 资产，这样各个独立项目（甚至是点解决方案）都直接适用于一致集成和管理的基础结构。

本文档向您介绍了以上描述的简化和解决方法。您还需执行开始的几个步骤以实现对 IBM Security Directory Integrator 工具集（具体而言就是以下两个程序）的精简:

- 开发环境，称为*配置编辑器*或简称“CE”,
- 运行时引擎，简称为*服务器*。

1. 请别让名称欺骗了您，IBM Security Directory Integrator 不限于目录工作，它还支持所有主要的数据存储、传输、协议和 API – 当然还包括 LDAP 目录。

将使用 CE 汇总 IBM Security Directory Integrator 解决方案，并且使用一个或多个服务器来支持这些解决方案。这些程序将协同工作，以使用户体验无缝运作，甚至可使您跨平台工作；例如，在笔记本电脑上进行开发，同时在大型机上远程运行测试和调试解决方案。

使用 JavaScript 编写脚本

如上所述，您可通过 IBM Security Directory Integrator 快速汇总集成解决方案。但是，为了使用您自己定制的处理和流行为来扩展内置自动化功能，您必须编写脚本片段。

脚本编制是使用 JavaScript 完成的，IBM Security Directory Integrator 包含可提供快速可靠的脚本编制环境的 IBM JSEngine。因此，您将需要了解和使用核心 JavaScript 语言。有几个不错的在线和硬拷贝资源可用于学习 JavaScript。请查看 IBM Security Directory Integrator 新闻组和 Web 站点以获取相关建议和链接。

有关在 IBM Security Directory Integrator 中编写脚本的更多信息，请参阅*配置 Directory Integrator*。

安装 IBM Security Directory Integrator

IBM Security Directory Integrator 将在几分钟内完成安装，您可以立即开始构建、测试和部署解决方案。该产品可在各种平台上运行，包括 Microsoft Windows、IBM AIX®、IBM System z® 以及多种 UNIX 和 Linux 环境。

安装 IBM Security Directory Integrator 时会涉及三条路径，安装程序将要求您指定前两条路径:

1. *安装目录*，保存程序文件以及用于启动各种工具的批处理文件或脚本的位置。
2. *解决方案目录*，通常缩写为“SolDir”，它是运行 IBM Security Directory Integrator 的当前文件夹。您将注意到，“配置编辑器”开发环境（`ibmditk`）和服务器（`ibmdisrv`）的启动批处理文件和脚本都是以可将目录更改到解决方案目录的命令开头。因此，将从解决方案目录扩展解决方案中使用的所有相对路径。
3. *工作空间*文件夹。这是保存项目和资源[2]文件的位置。该位置将缺省为解决方案目录中名为“`workspace`”的文件夹。

关于安装 IBM Security Directory Integrator 的更多信息，请参阅 *安装和管理* 中的 *IBM Security Directory Integrator 安装指令*。

安装教程文件

本书中的教程练习需要位于 IBM Security Directory Integrator 安装目录的 `examples/Tutorial` 子文件夹中的支持数据文件。例如，标准的 Windows 安装会将这些文件置于以下目录中:

```
C:\Program Files\IBM\TDI\V7.2\examples\Tutorial
```

“Tutorial”目录应该包含以下文件:

- `CreatePhoneDB.assemblyline`
- `index.html`

2. 这些术语已在第 9 页的第 2 章，『介绍 IBM Security Directory Integrator』中进行了说明

- OtherPage.html
- People.csv
- PhoneNumbers.xml
- readme.txt
- Return web page.script

注: 如前一部分中所提到的，安装程序将要求您指定解决方案目录的位置。该目录是将存储项目和资源文件的位置，它通常是主区域下名为 My Documents\TDI 的子目录。

将 Tutorial 文件夹复制到解决方案目录，以便可以从“配置编辑器”工具更方便地访问该文件夹。

先简化再解决

本部分帮助您了解设计数据集成解决方案时应从何处着手。虽然设计策略不断增大，但是它适用于任何大小的集成和系统部署项目（包括大型项目）。

使用类似于此策略的策略，设计使用 IBM Security Directory Integrator 来逐步进行数据集成的解决方案:

- 通过将问题分解成较小且易于管理的部分，降低复杂程度。
- 从总体解决方案的一部分（最好是可以在一周或两周内完成的部分）着手。
- 从总体解决方案中可以单独投入生产的一部分着手。

处理复杂的项目

消化大型集成和系统部署项目的最佳方法是将问题分解成许多小型且易于管理的部分，从而降低复杂程度。完成此操作后，您可以开始处理总体解决方案的某个部分，最好是可以单独部署的一部分。这样，当您处理其余组成部分时，它已经在提供投资回报。

把要处理的片段隔离之后，就可以通过重点处理基本通信单元（数据流本身）来进一步进行简化。现在您已准备好，可以开始实施了。集成开发是使用 IBM Security Directory Integrator 配置编辑器（缩写为“CE”）通过一系列的“尝试 – 测试 – 优化”周期来完成的，这使得该过程成为一个重复的、甚至探索性的过程。这不仅能帮助您发现更多有关自己的安装的信息，而且，随着您对问题集及其对基础结构的影响的了解不断加深，您就可创建自己的集成解决方案。

相关主题

请参阅以下主题，以了解关于 IBM Security Directory Integrator 如何允许您使用 AssemblyLine 变换数据的说明。

- 『内核/组件体系结构』
- 第 4 页的『“条目 – 属性 – 值”数据模型』
- 第 5 页的『数据流 = AssemblyLine』

内核/组件体系结构

IBM Security Directory Integrator 的基本质量取决于其内核/组件设计。

术语*内核*在此是指快速集成开发（RID）框架，它使您能够快速汇总集成解决方案并提供自动化执行逻辑来驱动这些解决方案。即使最简单的数据流也可直接使用如日志/跟踪模块、连接恢复、更改检测、错误处理以及外部管理 API 这样的功能部件，如果未提供这些功能部件，您将需要手工进行编码（因此经常被忽略）。

除了这个一般内核功能外，IBM Security Directory Integrator 还提供了一组特定于数据源的组件，即帮助程序对象，这些对象提炼出与数据源进行交互的技术详细信息。最常使用的两种类型的组件将是*连接器*和*解析器*。

连接器提供与各种数据源的连接以及对结构化数据的固有处理（而不管其底层组织如何）。某些连接器还充当事件处理程序，例如绑定至 IP 端口并等待传入连接，或“侦听”将在目录、数据库或文件中发生的更改。

另一方面，解析器用于处理非结构化的数据，即字节流（如在文件、POP3/SMTP 电子邮件、MQ 消息中发现的字节流）和流经 IP 端口的数据。

IBM Security Directory Integrator 提供了连接器和解析器的可扩展库，每个组件都旨在用于处理特定系统、服务、API、传输或格式。IBM Security Directory Integrator 组件的可互换性使您能够基于测试数据（例如文本文件）来构建解决方案，然后只需换出所使用的连接器就可将解决方案指向实时源以进行验证和部署。

此外，IBM Security Directory Integrator 组件可以直接使用，并且易于构建和扩展。可以增补库来处理定制数据源和服务，方法是：从社区 Web 站点下载新组件、用 Java™ 编写您自己的组件，或直接在 CE 中使用脚本以交互方式构建和测试这些数据源和服务。

“条目 - 属性 - 值”数据模型

数据的组织和存储方式因系统的不同有很大差异。

- 数据库将信息存储在通常具有固定列数的行中，每列具有该记录的单个值；
- 目录保存可包含不同数量属性的面向对象的条目。这些条目则不包含值或包含一个或多个值[3]；
- Lotus® Domino 数据库包含由字段组成的文档，每个字段都可定义为单值或多值；
- 仍然有其他系统将其数据内容表示为节点、对象、记录、格式化字节流或键值集。

为了使通信对所有参与者都有意义，数据格式必须兼容，或者必须进行转换以适合所涉及的每个系统。这称为*数据编组*，通常是集成专家首先要面对和克服的、会很快消耗极大部分项目资源的障碍。IBM Security Directory Integrator 连接器通过将特定于源的类型自动转换为一致的规范表示来为您处理这个问题。各个数据值将转换为相关的 Java 对象，并具有以相同方式表示的可比较的本机类型。例如，从文件中读取的行、LDAP 字符串属性、Domino 文本字段以及 RDBMS CHAR 和 VARCHAR 列都将通过各自的连接器转换为 `java.lang.String`。

然后这些已编组的值将累计进属性中：由 IBM Security Directory Integrator 定义的专用 Java 对象。如上所述，某些源仅允许每列或每字段具有一个值，而其他源允许在同一个属性名称下存储多个值。IBM Security Directory Integrator 属性支持单值和多值实施，如有必要（例如表示数据库中可空列时）甚至可以不包含值。

---

3. 请考虑一下这个事实：您可能具有多个电子邮件地址，所有这些地址可能都存储在公司员工目录中标题为“邮件”的多值属性中

组成单个数据单元（即记录、消息、文档等）的所有属性都将收集在称为*条目*的其他 IBM Security Directory Integrator 对象中。一个条目可以保存任意数量的属性，或不保存任何属性。

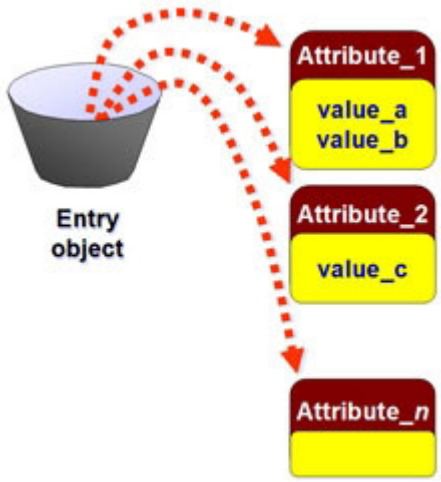

图 1. “条目 - 属性 - 值”数据模型

每个数据流都具有称为其“Work 条目”的主条目“存储区”。只要连接器读入数据，它就会创建属性，并将这些属性置于 Work 条目中。为输出配置的任何连接器都使用已在 Work 条目中找到的属性来驱动对目标系统的更改。

这一包含两个阶段的方法对数据传输、变换、过滤和扩充提供了几乎无限的灵活性。这还意味着，您甚至还不必考虑连接到输出系统，就可以完全使用输入连接器初始构建您的数据流，然后在读取和处理数据时使用 CE 以交互方式检查数据。

正如您将在稍后所看到的，IBM Security Directory Integrator 条目处理复杂的分层数据就像处理单层模式一样容易。

数据流 = AssemblyLine

解决方案中的每个数据流都作为 IBM Security Directory Integrator *AssemblyLine* 实施，在本文献和其他文献中也缩写为“AL”。

AL 是组件的有序列表，形成从输入源到目标的单一连续路径。内核提供的内置行为将组件关联在一起，并将 Work 条目中携带的数据从一个组件传递到下一个组件。

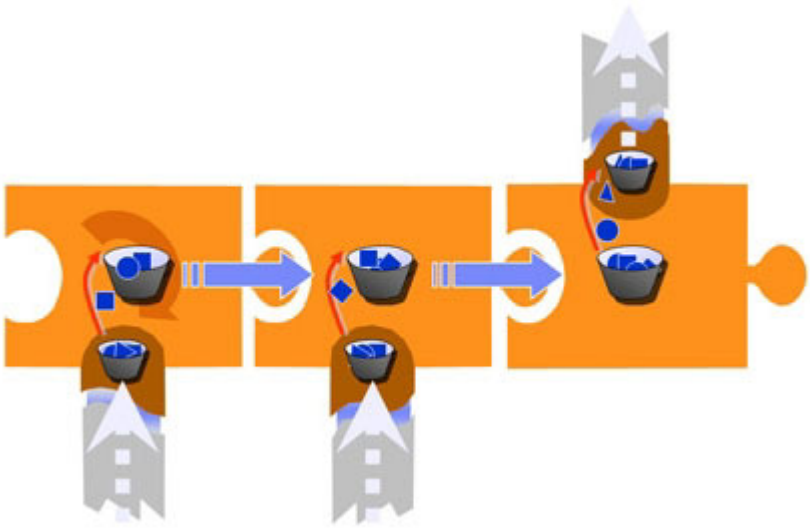

图 2. 沿 AssemblyLine 向下流动的数据

人们常说一图胜千言，上图也不例外。三个拼图块表示链接起来构成 AssemblyLine 的连接器。每幅拼图块的较深色“主干”突出显示连接器特定于数据源的部分（即到已连接系统的*接口*）称为*连接器接口*（缩写为“CI”）。每个拼图块剩下的颜色较浅的部分描绘了内核的一般功能：使所有的组件以相似且可预见的方式工作，这使组件可链接起来，并提供了具有用于定制的控制点的自动行为模式[4]。

该图说明了一些更重要的概念。例如，除了上面显示的从组件到组件沿 AssemblyLine 流动的 Work 条目，在每个连接器接口中另外还包含一个 Java“存储区”。每个本地条目对象在该 CI 执行的读写操作期间都用来高速缓存数据，并且该对象称为其 *Conn 条目*。

现在请注意说明不同 Conn 条目和 AL Work 条目之间的数据流动的弯曲箭头。这些箭头是*属性映射*，每个箭头都表示将数据移入或移出以及传入和传出 AL 的一组规则。将数据从 Conn 条目提升到 Work 条目中的映射称为*输入映射*，因为它们确定了哪些数据进入 AssemblyLine 。最右边的拼图块中显示另一个方向（从 Work 条目到 Conn 条目）的数据移动的箭头称为*输出映射*。

由于任何时候都仅有一个 Work 条目，因此您可以得出这一结论： AssemblyLine 一次仅处理一项：例如，一个数据库行、目录条目、MQ 消息等等。尽管 AssemblyLine 每秒可循环几百甚至几千个条目，但这仍是 IBM Security Directory Integrator 很重要的一个方面[5]，这是设计解决方案时要考虑的重要事项。当然可以跨多个 AssemblyLine 展开工作，您将在其他 IBM Security Directory Integrator 文献中找到该技术和其他技术以优化 AL 的性能。

入门

对于任何集成项目，为手头的问题创建一个图都是个不错的起点。

4. 正如您所看到的，每个 AssemblyLine 组件都反映 IBM Security Directory Integrator 的内核/组件体系结构。如果您决定生成自己的组件，那么只需实施其接口。AL“包装器”及其丰富的内置功能随 IBM Security Directory Integrator 内核自动可用。

5. 性能将取决于 AssemblyLine 的设计和复杂程度以及运行服务器的机器的配置。

使用铅笔和一张纸以粗线条草拟出所需的流。这个练习不仅能帮助您将任务范围以直观的形式呈现出来，还可以作为在 IBM Security Directory Integrator 中实施这些流的蓝图。

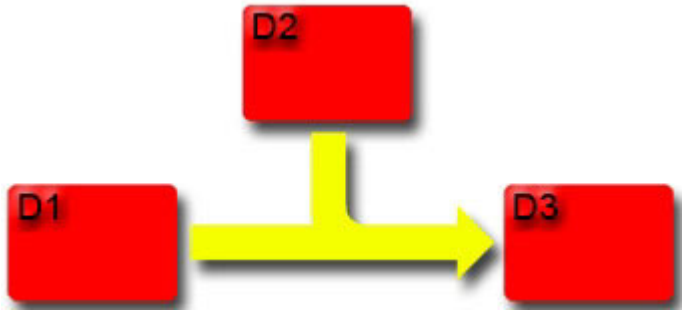

图 3. 教程场景

创建 IBM Security Directory Integrator 解决方案的第一步是将数据源之间的数据流转换为由连接器组成的 AssemblyLine 。“简化并解决”这一 IBM Security Directory Integrator 真言指示应以递增方式构建解决方案，入手时应尽量简单。

要说明这一点，请考虑将用于第一个 AssemblyLine 的示例场景。该集成任务涉及标记为 D1、D2 和 D3 的三个数据源。期望的解决方案是将 D1 的内容迁移至 D3，并使用 D2 中发现的值增补该数据。将该需求转换为 AssemblyLine ，将得到下面的三个连接器（每个连接器代表一个数据源）：

1. 第一个连接器在 D1 中进行*迭代*以将该数据填入流中;
2. 后跟第二个连接器，该连接器*查找* D2 中的相关记录，并将这些值与来自 D1 的值合并;
3. 最后是第三个连接器，该连接器配置为将这些增补记录*添加*至 D3。

您可通过 IBM Security Directory Integrator 仅从以下两个连接器入手来简化任务，而不是一次性攻克整个问题: 一个连接器将 D1 的内容读入 AL，另一个连接器将这些值写入 D3。一旦该最小的 AssemblyLine 正常工作，就可以使用连接器将该 AssemblyLine 扩展至 D2 以加入更多属性。这准确说明了如何创建您的第一个 IBM Security Directory Integrator 解决方案，且本指南的其余部分包含了指导您完成该过程的步骤。

第 2 章 介绍 IBM Security Directory Integrator

本部分提供了在了解 IBM Security Directory Integrator 基础时很有用的信息，还提供了一组教程练习以使您获得使用开发环境的实际经验。

了解该产品的第一步是启动 IBM Security Directory Integrator 开发工具（也称为“配置编辑器”，或简称为 CE）。

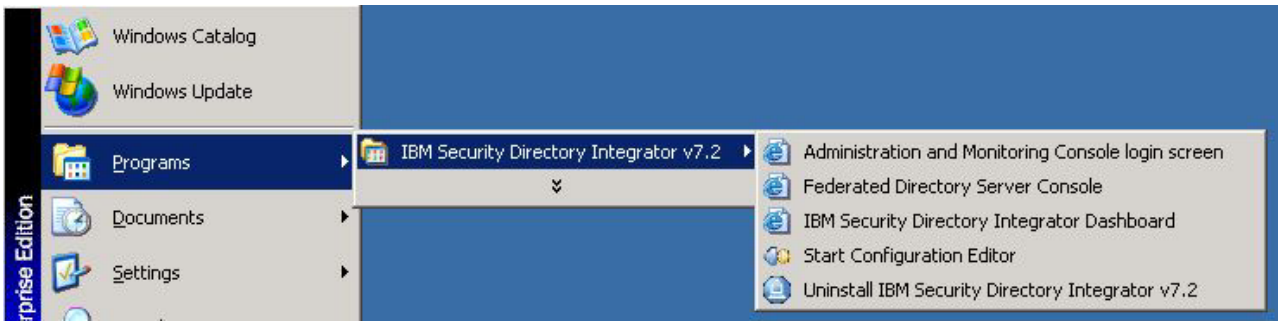

图 4. 启动配置编辑器

首次启动 CE 时，将看到用于指定工作空间的以下对话框。

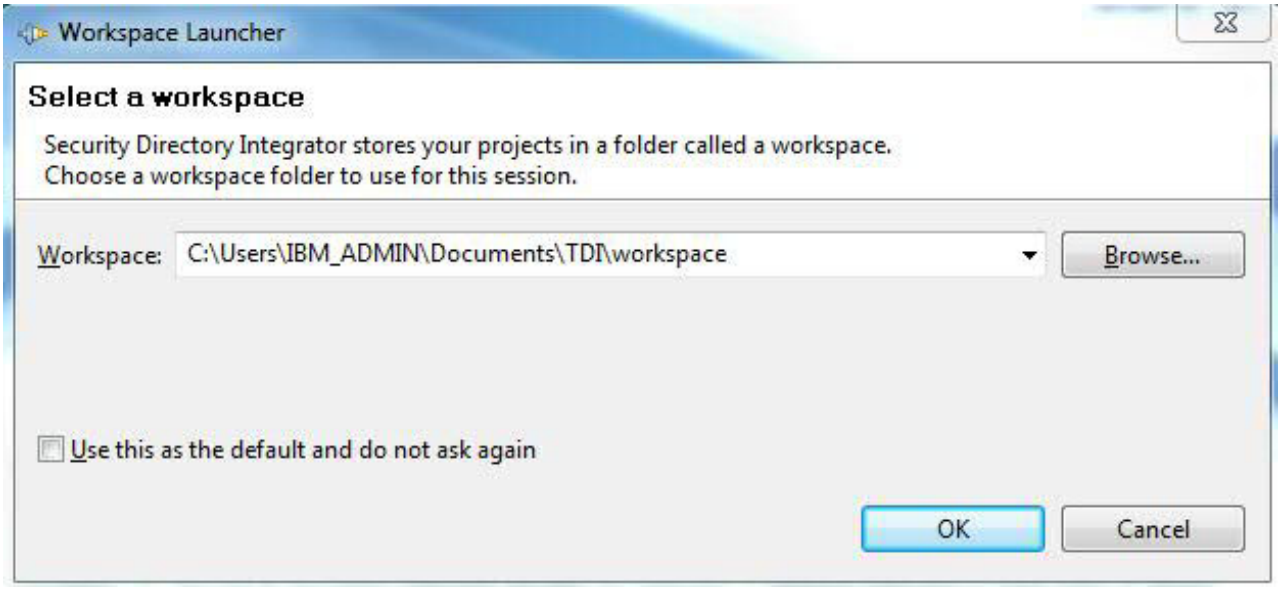

图 5. 选择工作空间

工作空间是配置编辑器将存储项目文件（包括组件和 AssemblyLine ）的位置，该工作空间通常位于解决方案目录下。

如果您对工作空间的位置感到满意，请按**确定**按钮。此时将显示“欢迎”屏幕。

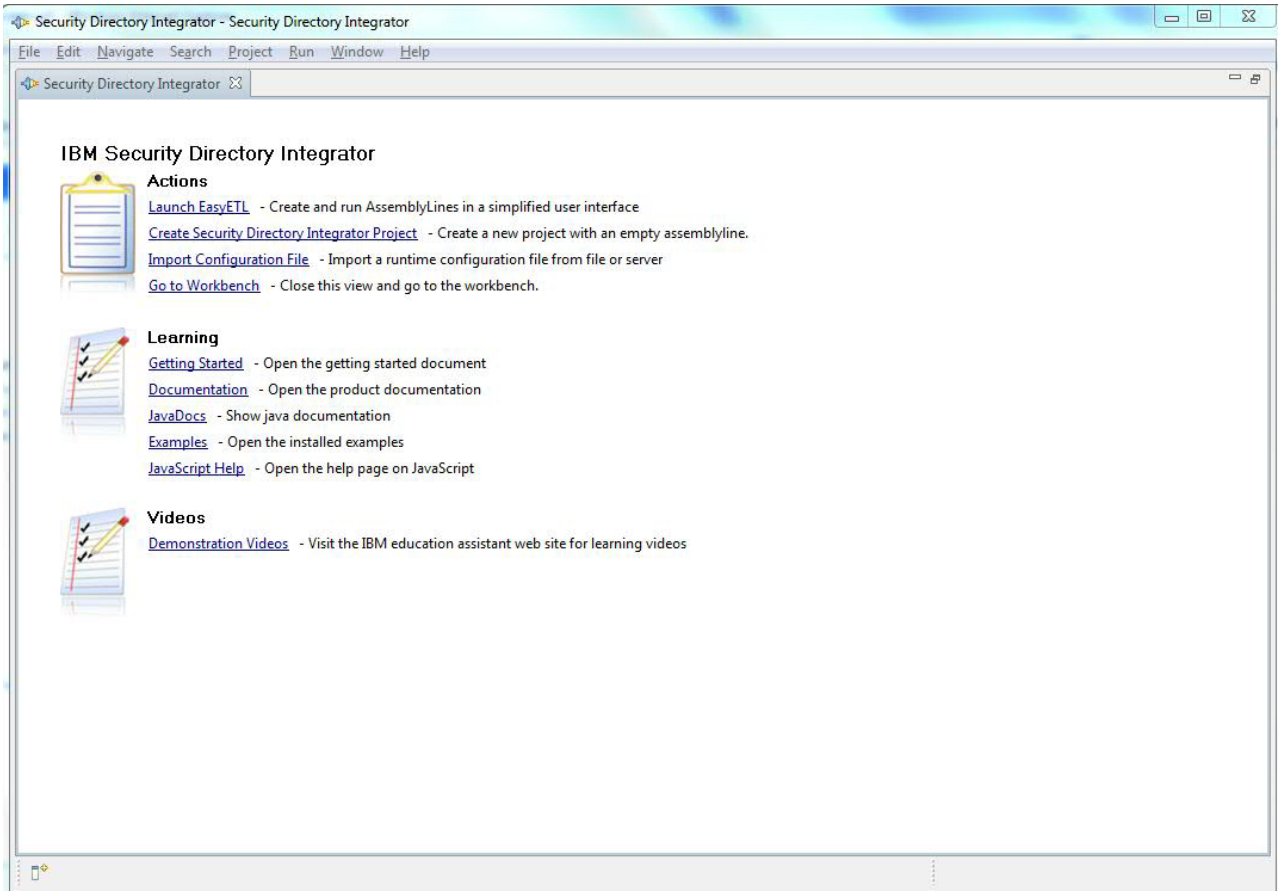

图 6. 配置编辑器"欢迎"屏幕

"欢迎"屏幕提供了大量快速启动链接[6]。

只要您使用 IBM Security Directory Integrator 构建、测试或修改集成解决方案，就将在项目中进行处理。项目是 AssemblyLine 及其构成组件的集合，每个项目都显示在您的工作空间中它自己的子文件夹中。构成项目的 AssemblyLine 和组件将存储为单独的文件，而这些文件又位于项目文件夹的子目录中。

选择该页面的顶部的第二个链接[7]（*创建Security Directory Integrator项目*）来设置您的第一个项目。现在必须提供新项目的名称。将它命名为"Tutorial"，然后按**完成**。

6. 通过在主菜单中选择**帮助** > **欢迎**可以随时返回到该屏幕

7. 最顶部链接（启动 EasyETL）打开简化的工作台并涵盖在附录第 81 页的『EasyETL 指南』中。

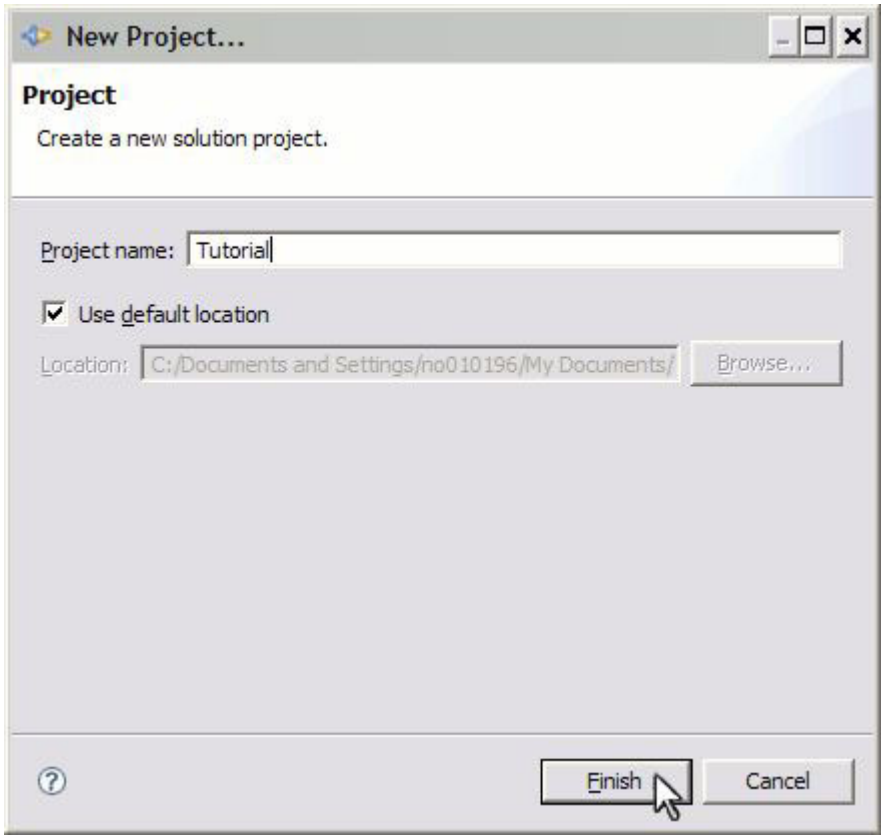

图 7. 命名新项目

此时将看到主开发工作空间。可在此处调整所有面板的大小，并且可决定屏幕的组织方式。在屏幕的此处您将看到的内容是缺省 IBM Security Directory Integrator 透视图[8]。

8. *透视图*只是开发环境面板的组织结构。如果更改了布局，且想返回至缺省的 IBM Security Directory Integrator 透视图，只需单击顶端菜单中的**窗口**并选择**复位透视图**选项即可。

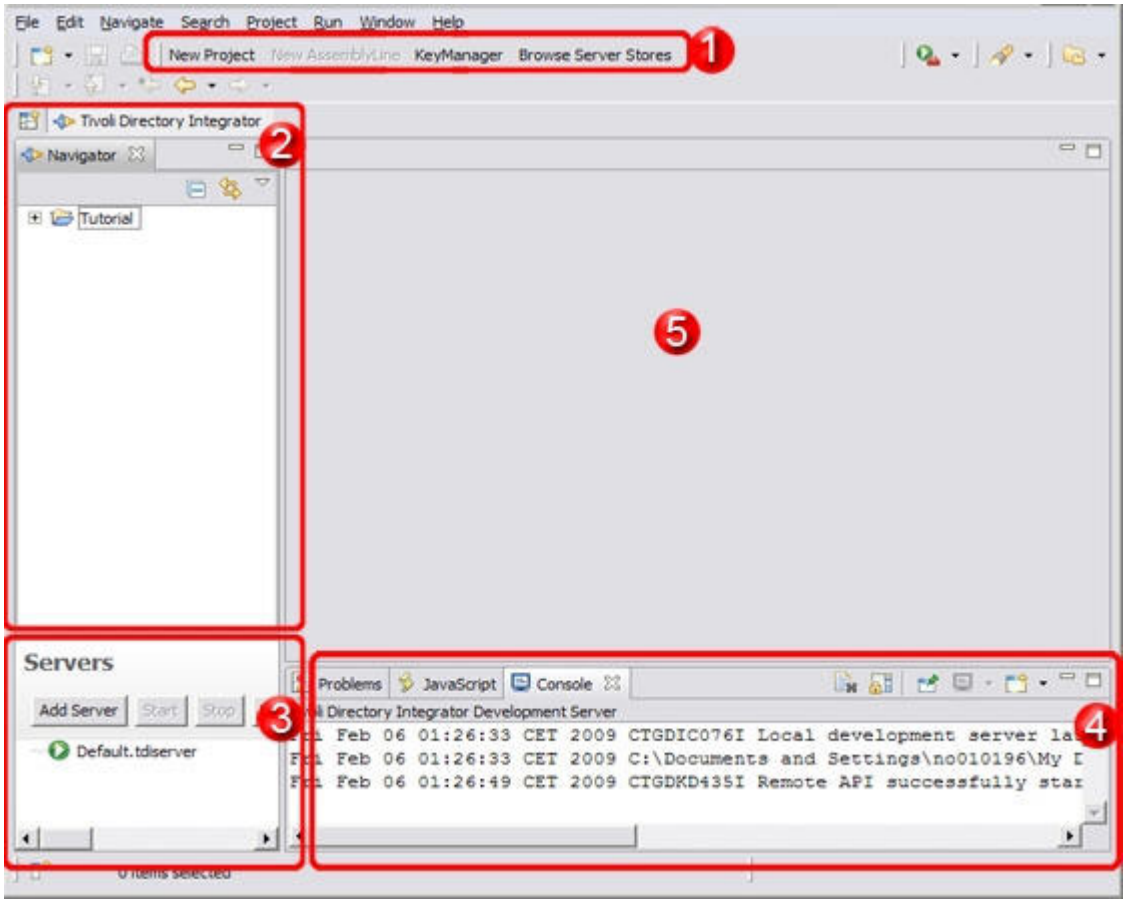

图 8. "配置编辑器"主屏幕

注: 您需要使用的唯一透视图是 **Security Directory Integrator** 或 **Easy ETL** 透视图。如果配置编辑器不具有您预期的外观，或者不具有您预期的窗格，请尝试下列步骤:

- 选择**窗口 > 打开透视图 > Security Directory Integrator**。该选项选择 Security Directory Integrator 透视图。
- 如果以上步骤无效，请选择**窗口 > 重置透视图...**。此命令确保将所有缺省窗格置于应该位于的位置。

这是主屏幕，使用 IBM Security Directory Integrator 时将在此处花费大部分时间。不用深入讨论此处所有导航元素的详细信息[9]，我们来看一下以上屏幕快照中突出显示的已编号区域:

1. 主按钮行中间是一组用于创建新项目的快捷方式，如果在导航器中选择了项目，那么这组快捷方式将用于在项目中创建新的 AssemblyLine。还有一个用于启动 KeyManager 工具以使用证书密钥和信任库的按钮，以及用于从与该项目相关联的 IBM Security Directory Integrator 服务器中检索各种属性设置的**浏览服务器存储**按钮。
2. 这是*导航器*面板，它提供了开发资产的树形视图。新"Tutorial"项目应在此处显示。

9. 如同大多数基于 Eclipse 的应用程序一样，将有很多方法执行相同操作。*配置 Directory Integrator*描述了所有可用的不同选项和面板。

3. “服务器”面板显示所有已配置服务器的状态。可以使用该服务器已为您启动的“Default.tdiserver”旁边的箭头图标进行查看。该面板还提供用于定义新服务器、启动和停止服务器以及刷新列表和查看服务器的日志的按钮[10]。

   请注意，只要您启动 AssemblyLine，配置实例[11] 和 AL 也将显示在该面板上。
4. 您将在此处看到一组选项卡，当前选定的选项卡显示来自服务器的控制台输出。现在此处显示的消息告诉您服务器正在运行，且其 API 已初始化并已就绪待用。
5. 该屏幕快照中的灰色区域是创建和打开 AssemblyLine 和组件时显示*编辑器*面板的位置。对于每种类型的资源（连接器、解析器、AssemblyLine 等），都特别设计了其各自的编辑器。

创建您的第一个 AssemblyLine

返回到简介中概述的示例场景，您现在将创建可将信息从 D1 迁移到 D3 的 AL，暂时忽略来自 D2 的数据的连接。

图 9. 仅包含两个数据源的简化的场景图

“Tutorials”文件夹（您应该已将其从 *TDI 安装目录*/examples 复制到解决方案目录）包含名为 People.csv 的文件:

```
First;Last;Title
Bill;Sanderman;Chief Scientist
Mick;Kamerun;CEO
Jill;Vox;CTO
Roger
Gregory;Highpeak;VP Product Development
Ernie;Hazzle;Chief Evangelist
Peter;Belamy;Business Support Manager
```

您可以看到上面的列表中使用的是*字符分隔值* 格式（CSV）。该文件表示 D1 输入数据源。AL 将抽取该数据并将它传输到将作为 D3 输出目标的 XML 文档中。

单击最上方工具栏中的**新建 AssemblyLine** 并将新 AL 命名为“CSV2XML”。

10. 如果由于某种原因，您的服务器未正常启动，请打开“TDI 服务器”，然后双击“Default.tdiserver”。这将打开相关联的服务器文档。请确保“安装和解决方案目录”设置正确，然后按该面板顶部的**创建解决方案目录**选项。如果这样未能更正问题，请联系支持人员。

11. 只要 IBM Security Directory Integrator 服务器装入配置，它就会创建包含该项目的 AssemblyLine 的*配置实例*，并允许这些 AssemblyLine 在包含它们的环境中运行。这意味着可以在同一服务器上多次装入同一配置，这样就会产生不同的配置实例，所有这些实例都包含同一组 AL，且各个 AL 之间不会互相干扰。

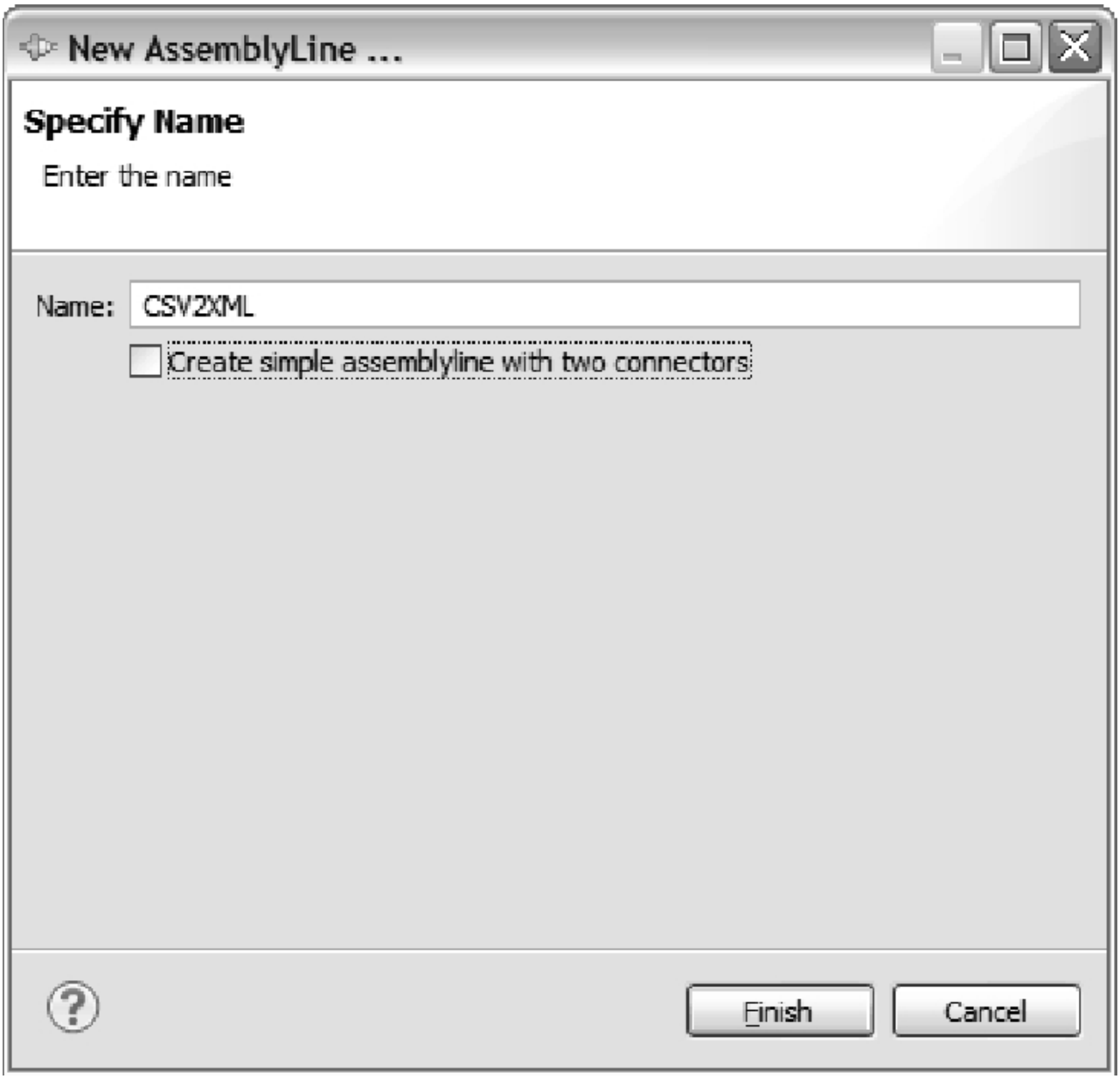

图 10. “新建 AssemblyLine ”对话框

现在按**完成**按钮以在“ AssemblyLine ”编辑器选项卡中打开该 AL。

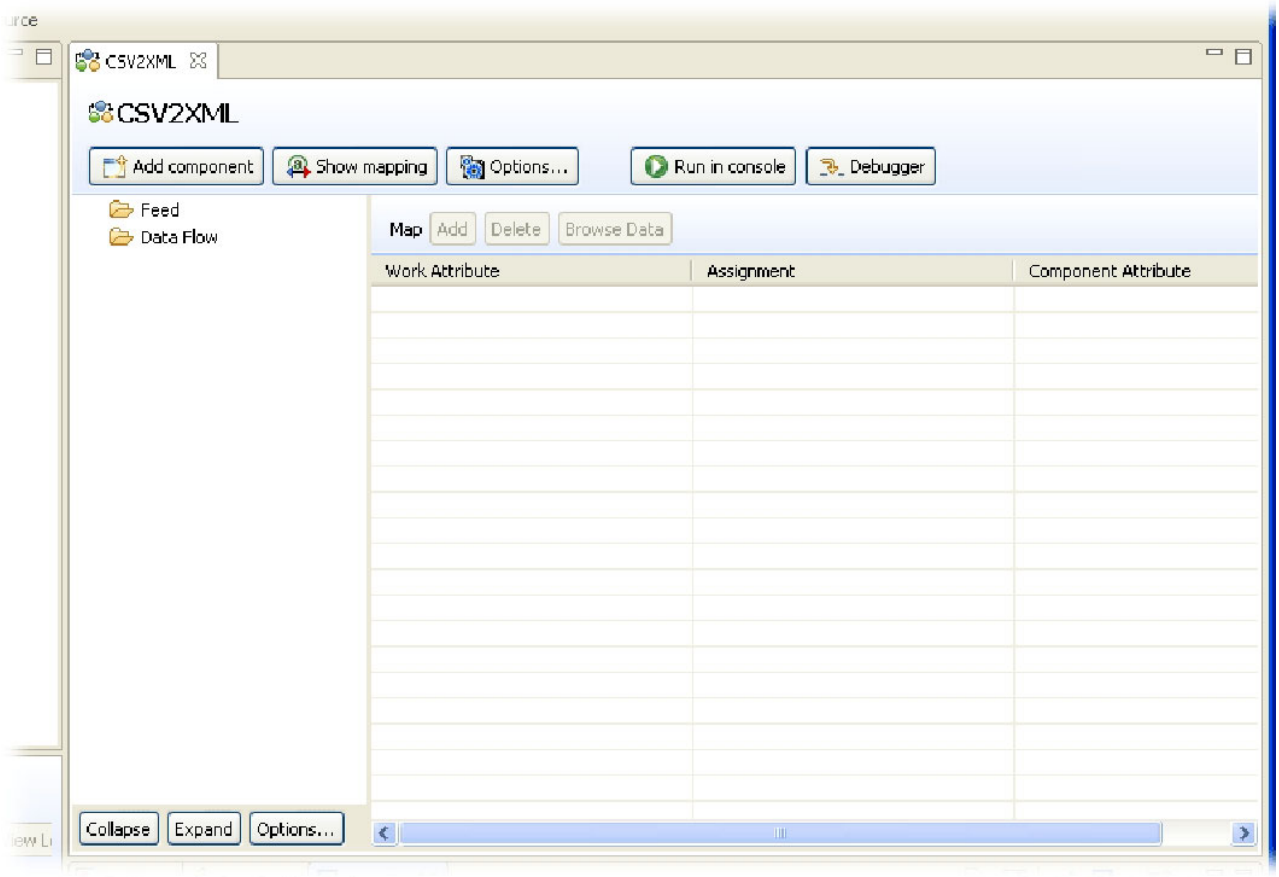

图 11. 空的 AssemblyLine 编辑器

AL 编辑器左边的部分包含了构成该“ AssemblyLine ”的组件列表，并且此刻除了名为*订阅源*和*数据流*的部分外其余位置为空白。右边的区域显示映射到 AL 内部和外部的所有“属性”。

要了解这些“ AssemblyLine ”部分，请思考一下我们想要该新 AL 执行的操作：*对于 CSV 文件中的每行，在 XML 文档中创建一个新节点*。只要有来自*订阅源*部分[12]中连接器的输入数据，IBM Security Directory Integrator 内核就会自动提供循环行为，从而驱动 AL *数据流*部分下列出的组件。

通过将连接器添加到“订阅源”部分以在 CSV 输入文件中进行读取来利用该功能。通过右键单击*订阅源*部分文件夹并选择**添加组件...** 来执行此操作。

12. 注意：一次仅会有一个*订阅源*连接器将数据传递给 AL。如果您在此处放置多个迭代器连接器，那么最上方的连接器将先为空，然后行中的下一个连接器才开始从其源进行读取。

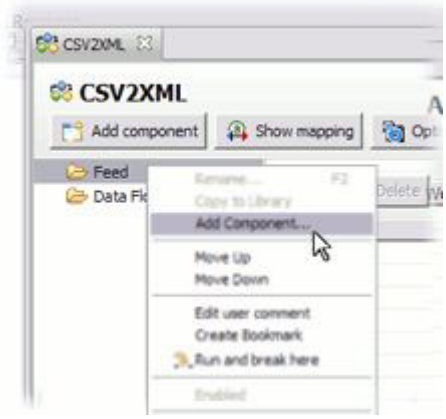

图 12. 插入新组件

此时将显示**选择组件**向导。

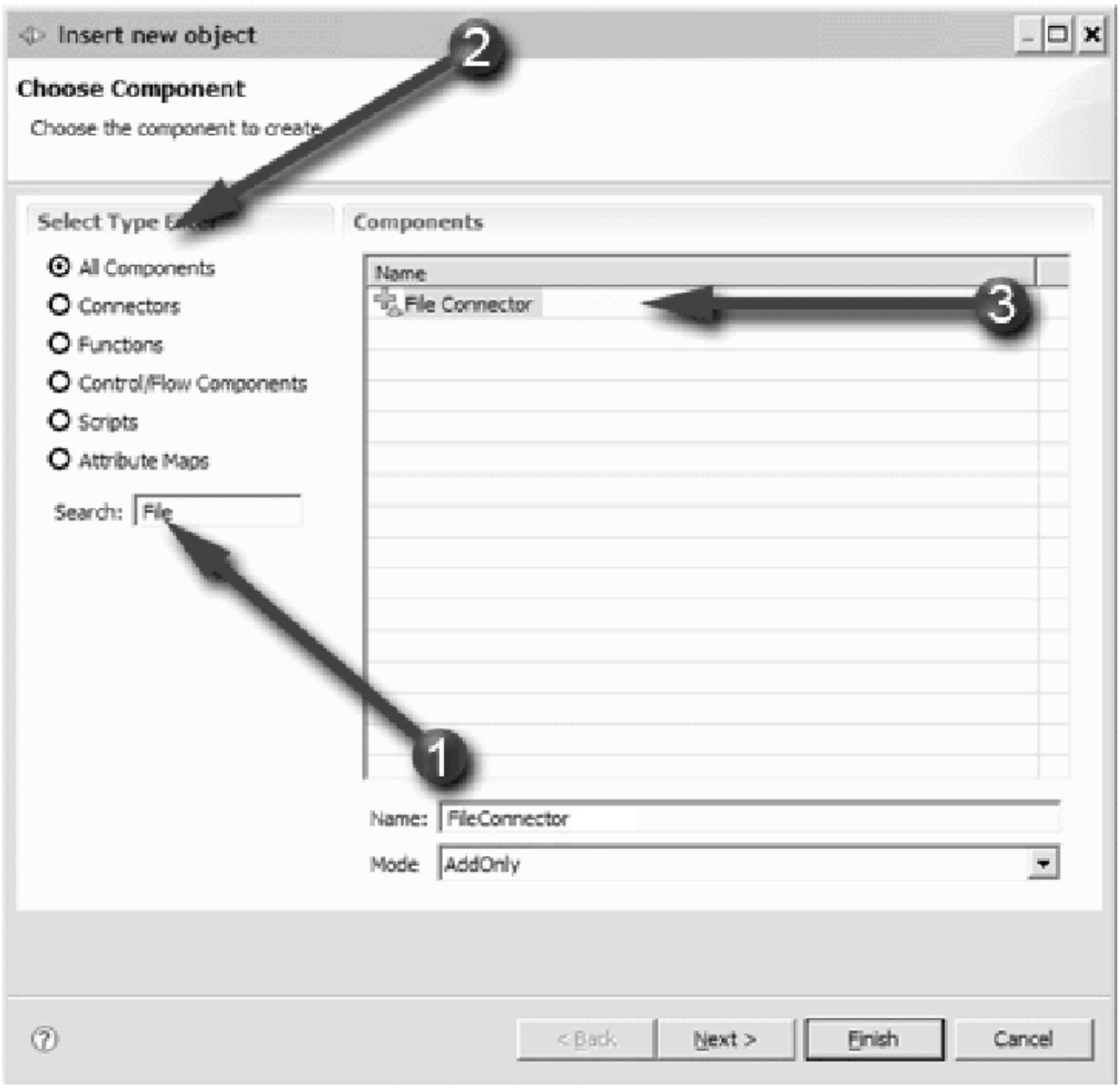

图 13. 选择组件

此对话框向您提供了几个选项以便您查找并选择所需组件:

1. 在文本字段中输入组件名称的任何部分，然后右边的选择列表将相应地进行过滤。对于此示例，输入“文件”。
2. 可以选择将选择列表限制为仅包含一种类型的组件 – 连接器、解析器以及脚本等。
3. 在该列表中查找并选择所需组件。在此示例中，该组件是“文件连接器”。

将自动为您把新连接器命名为“FileSystemConnector”。将该名称更改为“Read_CSV_File”以使其在解决方案的上下文中更有意义[13]，然后从**方式**下拉列表中选

13. 尽管可以任意命名连接器，但建议您以命名脚本变量的方式来命名连接器：以字母开始，后跟任意个数的字母、数字和下划线字符。这是因为所有的 AL 组件都会自动注册为脚本变量，从而在您稍后想从脚本代码重新配置和驱动组件时将会更加容易。

择**迭代器**。

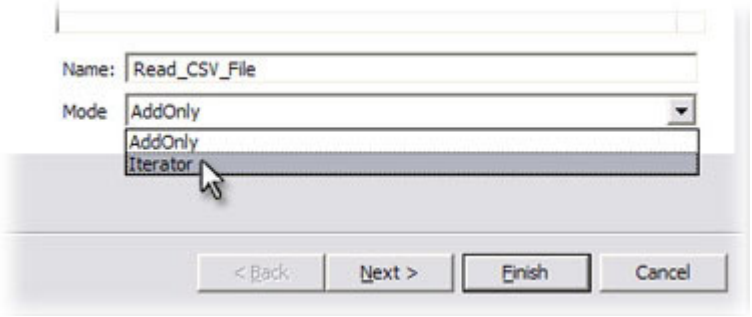

图 14. 重命名连接器并更改其方式

连接器的“方式”设置向内置 AL 执行逻辑指示了此组件在流中所起的作用。迭代器方式会产生所需的*针对每个*行为，从而将数据从 CSV 文件（一次一个条目）驱动到将添加至*数据流*部分的组件。

现在按**下一步**按钮以前进到选定连接器的配置面板。

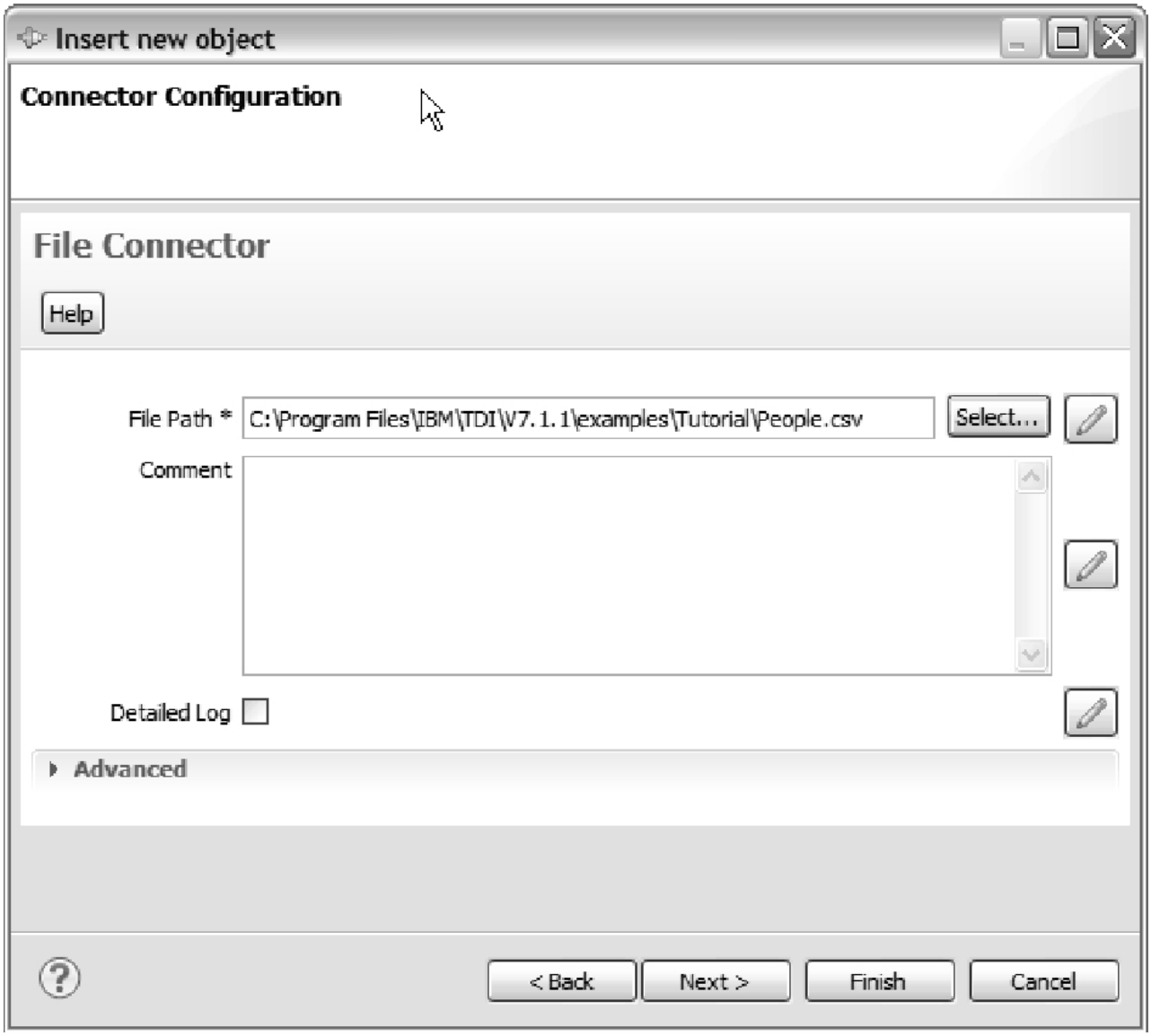

图 15. “文件连接器配置”面板

每个组件都提供自己的一组配置参数。现在屏幕上显示的内容是“文件连接器”的参数，它只有一个必需参数：**文件路径**。输入 `People.csv` 文件的路径（完整路径或以上屏幕快照中所显示的“解决方案目录”的相对路径）[14]–或按**选择**按钮以启动文件浏览器来查找该文件。

因为格式化文本文件是字节流而不是结构化数据源（如数据库或目录），所以您必须设置解析器以在读取流时解释其格式。IBM Security Directory Integrator 提供了强大且通用的“数据浏览器”功能部件，用于以交互方式测试“连接器/解析器”选择和配置。

一会我们将看一看该配置，但首先您需要再按一次**下一步**并继续进行**解析器配置**以完成此向导。在此处单击 **CSV 解析器**以将其选中。

14. 此技术可使您的解决方案更易移动和分享，因为您只需指定所需“解决方案目录”即可，所有相对路径无需改变就可工作。

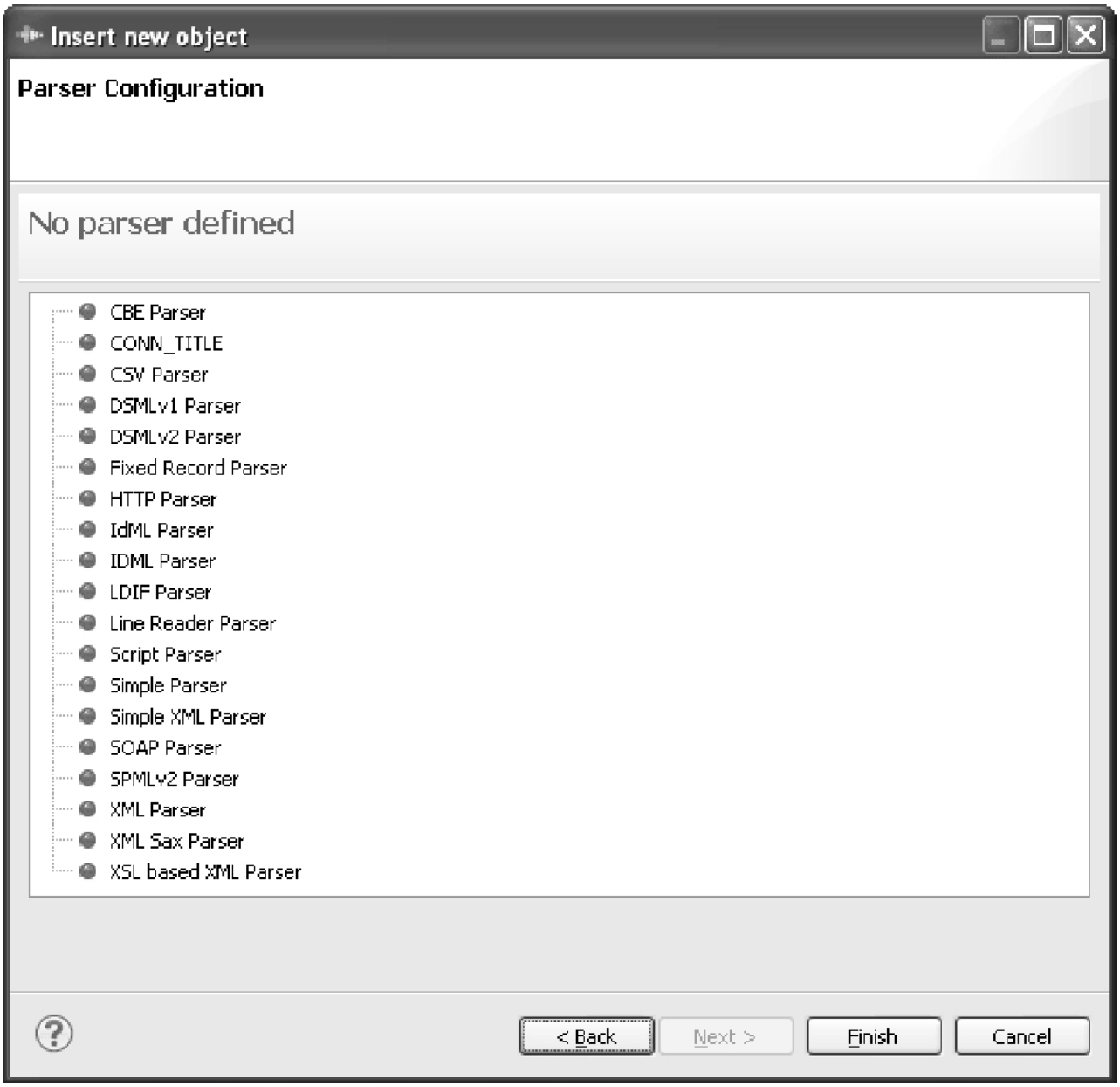

图 16. 在插入新对象期间选择解析器

一旦选择了 CSV 解析器，那么按**完成**以关闭向导。现在将看到“解析器配置”面板。由于无需更改缺省设置，只需再按一次“完成”以完成该向导即可。

下一步是使连接器*发现*输入源的模式以将这些值映射到“ AssemblyLine ”。这是会用到数据浏览器的地方[15]。通过右键单击“ AssemblyLine 组件”树中的新“迭代器连接器”并从上下文菜单中选择**浏览数据**，启动数据浏览器。

15. 由于您知道文件是 CSV 格式，最快的方法就是只需单击“迭代器连接器”的“模式”区域中的**连接**和**下一步**按钮即可。然后根据需要将已发现的“属性”拖动到“输入映射”中。如果不确定是何种格式，数据浏览器将很有用。但我还是认为您无论如何都应该尝试一下数据浏览器 :)

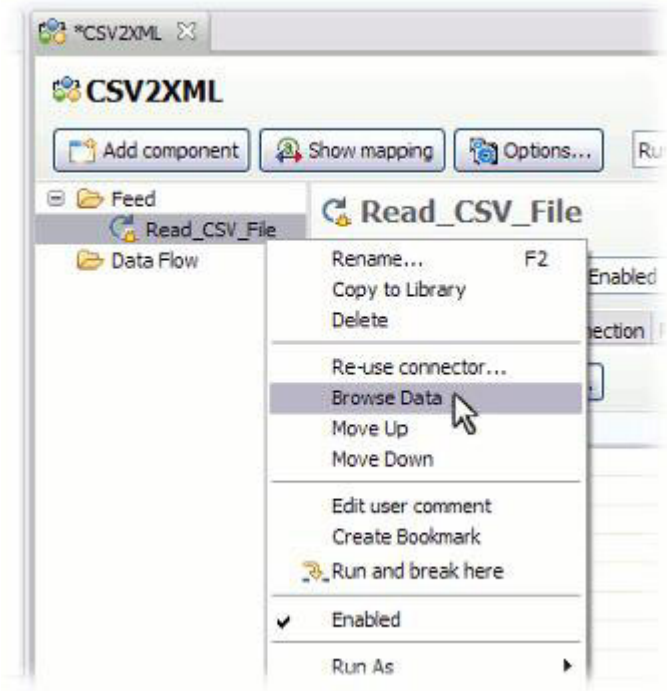

图 17. “浏览数据”上下文菜单选择

这将在新编辑器选项卡中打开数据浏览器。

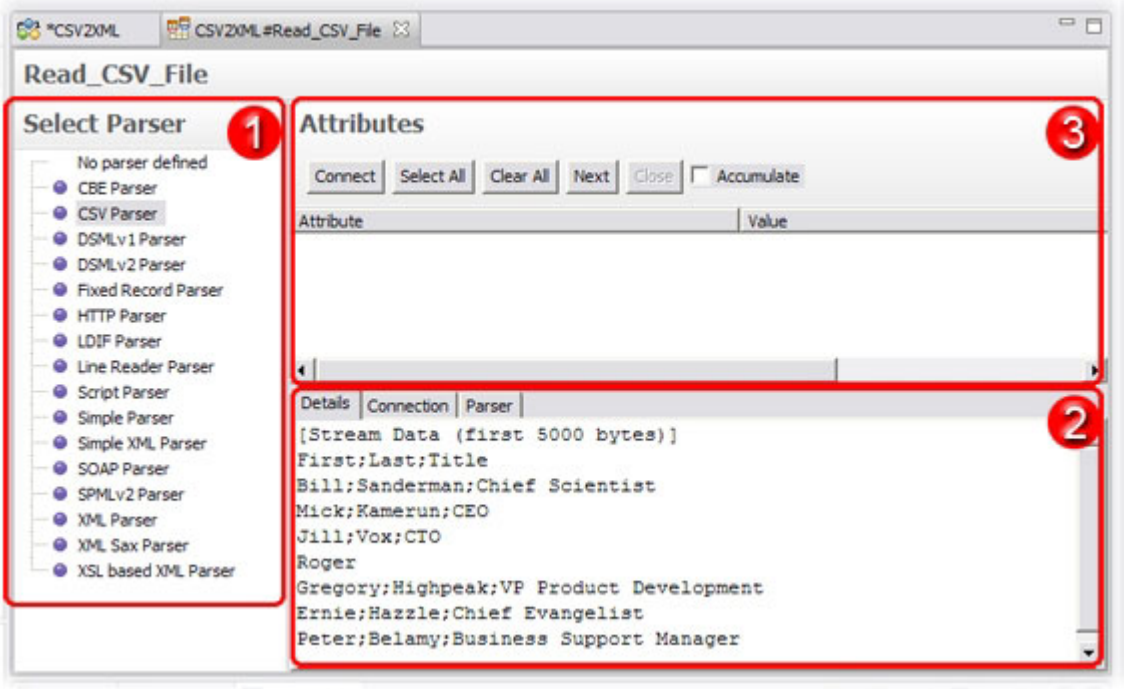

图 18. 数据浏览器

上面屏幕快照中标记为 1 的区域用于选择和更改选定解析器。区域 2 提供显示将解析的原始字节流的“详细信息”选项卡。还有用于更改“连接器的连接”参数的选项卡以及用于配置选定解析器的选项卡。

此对话框的最后一个部分（屏幕快照中的 #3）用于连接到数据源并发现可用的属性。通过先按**连接**，然后按**下一步**按钮执行此操作。

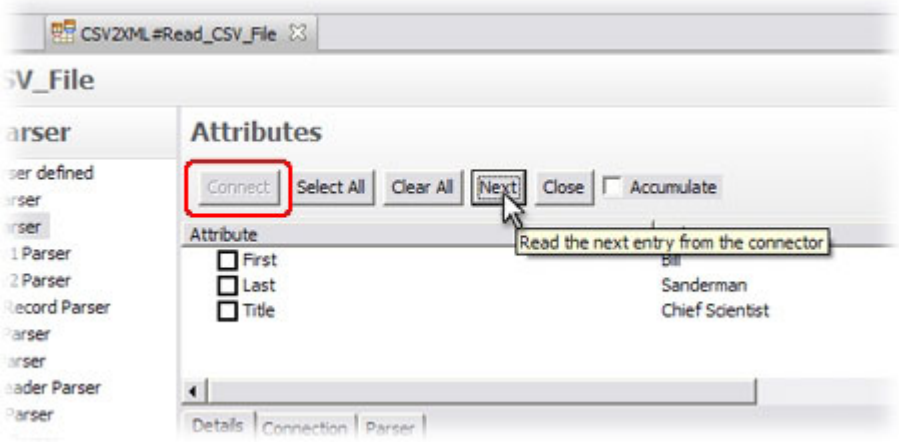

图 19. 通过浏览实时数据以交互方式发现模式

现在已发现该文件的模式。通过选中相邻的复选框或使用**全选**按钮，选择要映射到的“属性”（在此例中是所有的属性）。

使用 Ctrl-W 快捷方式关闭“数据浏览器”选项卡，或仅单击选项卡右边的“关闭”符号（X）并返回到 AssemblyLine 编辑器，其中 AL 应该像下面的屏幕快照一样[16]。

16. 如果由于某种原因，您的连接器处于*数据流*部分中，只需将其拖动到*订阅源*中。如果模式设置不是“迭代器”，那么请右键单击“连接器”，选择**方式**，然后选择**迭代器**。

图 20. 迭代器连接器就位的 AL

选定组件的详细信息显示在 AL 组件列表的右边，其中包括您刚在“输入映射”中设置的三个映射规则。每个“属性映射”项具有一个相应**赋值**，它是为设置目标属性的值而求值的脚本片段。

在继续之前，请花一点时间来回顾这些*赋值*: 您记得在第 4 页的『“条目 - 属性 - 值”数据模型』部分中，“ AssemblyLine ”具有将所有数据向下传输到 AL 的全局可用的 *Work 条目*。通过使用预先注册的脚本变量 `work` 在脚本代码中引用该对象。此外，每个连接器接口都有其自己的用作读取和写入的高速缓存的 *Conn 条目*。通过预先定义的变量 `conn`[17]从脚本访问该特定于组件的对象。要进行说明，请考虑第一个映射规则。它在 Work 条目中创建名为“First”的属性。该属性的值从以下赋值派生而来:

```
conn.First
```

此简写表示法引用了刚读入 `conn` 条目中的名为“First”的属性，该属性的值用于填充新 Work 条目属性。可比较的赋值脚本为:

```
return conn.getAttribute("First");
```
[18]

回到练习，现在您需要添加输出连接器以创建目标 XML 文档（数据源 D3）。这次请尝试使用“ AssemblyLine 组件”面板顶部的**添加组件**按钮。

---

17. `conn` 变量只可用于有限周期，如 IBM Security Directory Integrator 挂钩流程图中所示。在该范围外仍可通过查询其 Conn 条目的组件进行访问。

18. 对于熟悉 V6.x 以及更早版本的用户，还可以使用 7.0 之前的语法:

```
ret.value = conn.getAttribute("First")
```

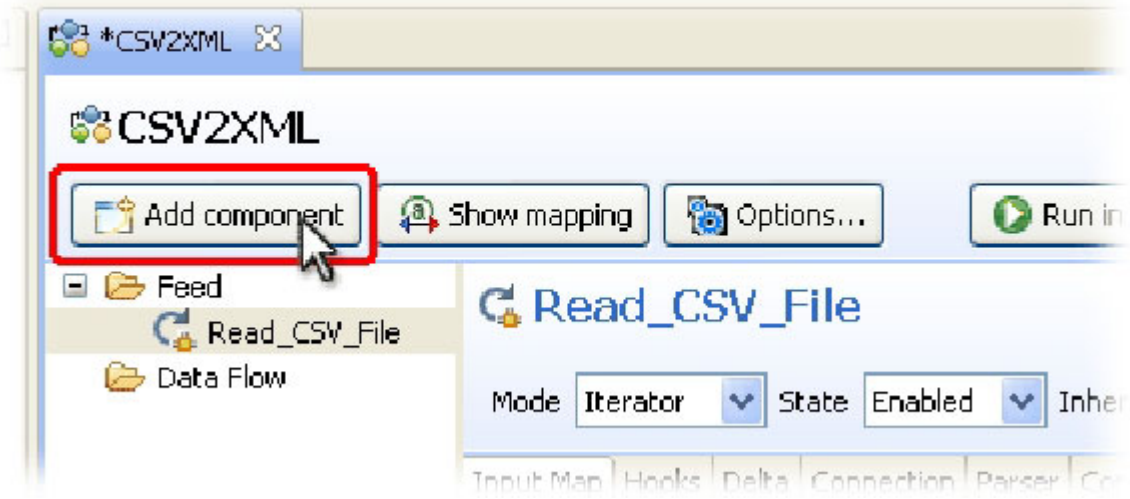

图 21. “添加组件”按钮

再次选择“文件连接器”，将其重新命名为“Write_XML_File”。将“方式”设置保留为**仅添加**，然后按**下一步**。

在“连接器配置”面板中，设置**文件路径**参数以写入“Tutorial”文件夹中名为 `Output.xml` 的文件。然后在下一个“向导”面板中选择“XML 解析器”。现在可以按**完成**，因为您无需更改 XML 解析器配置。请注意，对于输出连接器，不能执行“模式发现”，因为没有可用来执行发现的 `Output.xml` 文件。

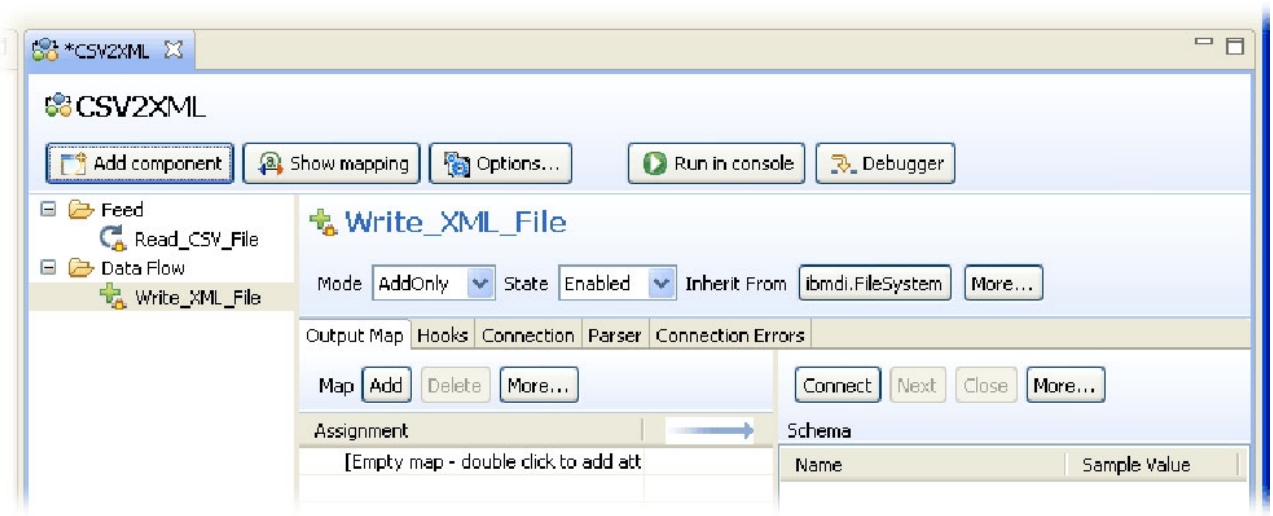

图 22. 具有两个就位的连接器的 AL

您可能已注意到，在选择组件时，其详细信息会出现在编辑器屏幕的右边部分。只要选择“订阅源”或“数据流”文件夹，就会显示该 AssemblyLine 的所有**属性映射**的概述。利用这些显示内容，就可方便地将输入属性复制到最新连接器的输出映射，因此现在请单击“订阅源”或“数据流”来显示此屏幕。

在此处可看到通过迭代器方式连接器放入 AL 中的属性（共三个）的列表。选择这些“输入映射”属性[19] 并将它们拖动到“Write_XML_File”连接器的“输出映射”，完成数据流。

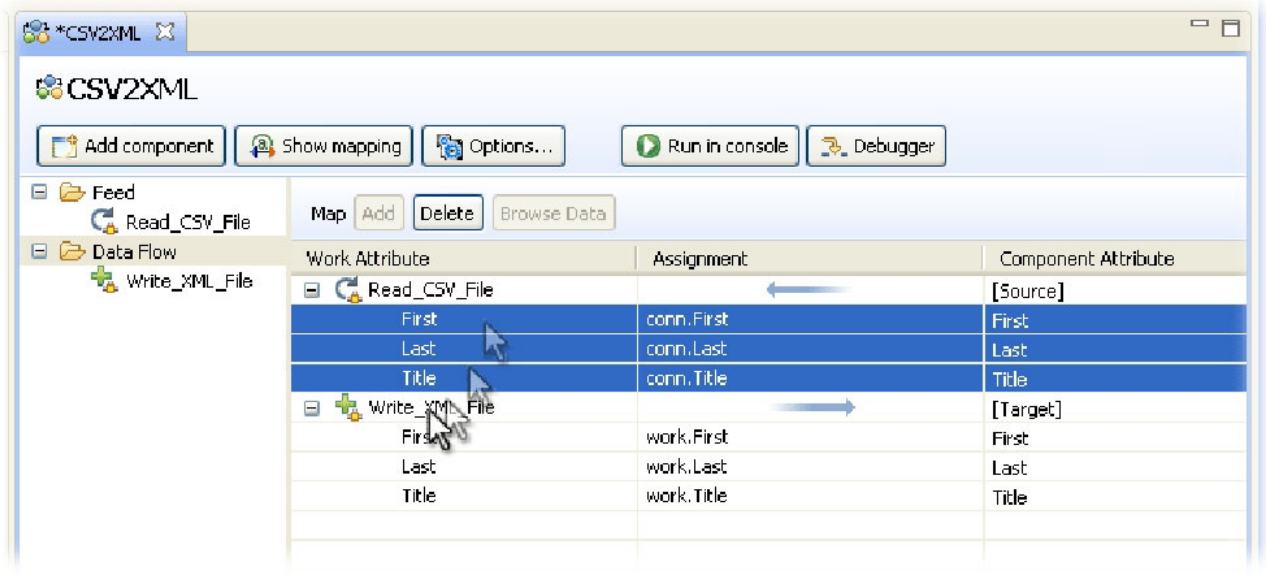

图 23. 将属性拖动到“输出映射”

请注意“赋值”如何自动从输入格式转换为输出格式。例如，“Read_CSV_File”连接器的“输入映射”中的第一个映射项将在 Work 条目中创建名为“First”的属性以保存在 `conn.First`（即读入 Conn 条目中的名为“First”的属性）中找到的任何值。在将此输入映射规则拖动到“输出映射”时，其赋值将更改，因此，现在该值将换为来自 Work 条目的值，并将在连接器的高速缓存（Conn 条目）中创建目标属性。

如果想更改任何映射规则的源，那么请编辑赋值。要更改将映射到的属性的名称，只需对它单击右键并重命名即可。为前两个“输出映射”规则执行此操作。

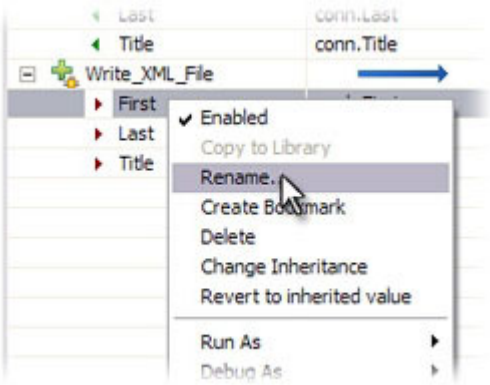

图 24. 重命名“属性映射”规则

将“First”更改为“FirstName”，将“Last”更改为“LastName”。

19. 您可以按住 Ctrl 键同时单击鼠标以选择多个属性，或按住 Shift 键同时单击鼠标以选择某个范围。

现在通过右键单击“Write_XML_File”输出映射本身并选择**添加属性**将新映射项添加到此输出映射。

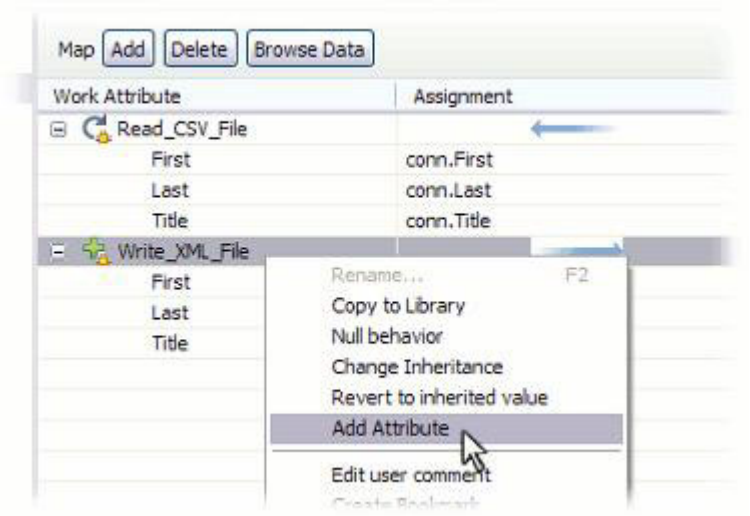

图 25. 将“FullName”属性添加到“输出映射”

将此新映射规则的目标命名为“FullName”，按**确定**然后双击它以编辑其赋值。这将打开“脚本”编辑器面板，并向您显示缺省的赋值脚本：`work.FullName`。在 Work 条目中当然没有“FullName”属性，因此该映射将无法设置任何值。您必须通过更改脚本来*计算*该值以便它能连接“First”和“Last”属性，并在这些值之间留一个空格：

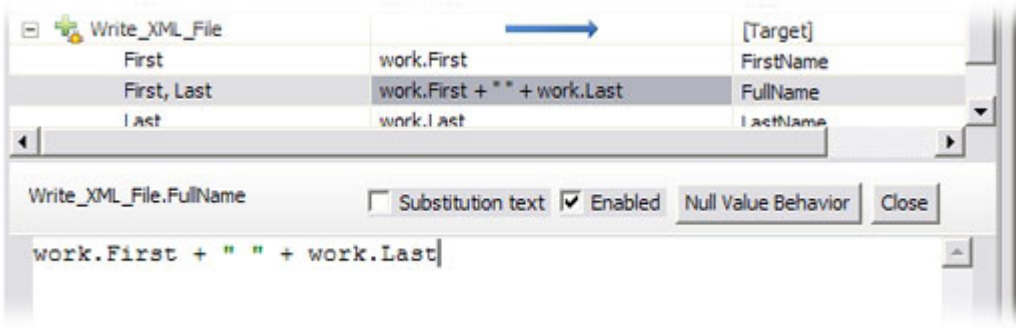

图 26. 编辑赋值

脚本应如下：

```
work.First + " " + work.Last
```

请注意，对于此类一行属性映射赋值脚本不需要表示结束的分号[20]。完成后，请按“脚本”编辑器面板右上方的**关闭**按钮。

运行您的 AssemblyLine

现在该测试您的 AssemblyLine 了。

20. 7.0 之前的语法也是受支持的，因此映射赋值脚本仍可以“ret.value =”开头。

按 AssemblyLine 编辑器顶部的**运行**按钮可执行此操作。

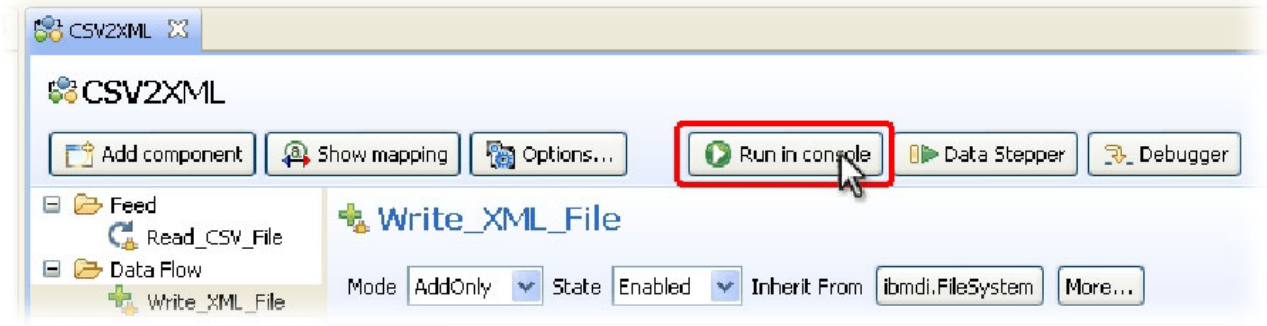

图 27. "运行"按钮

您现在将看到一个新选项卡打开，其中包含一个*运行窗口*，显示来自您的 AssemblyLine 的日志输出。

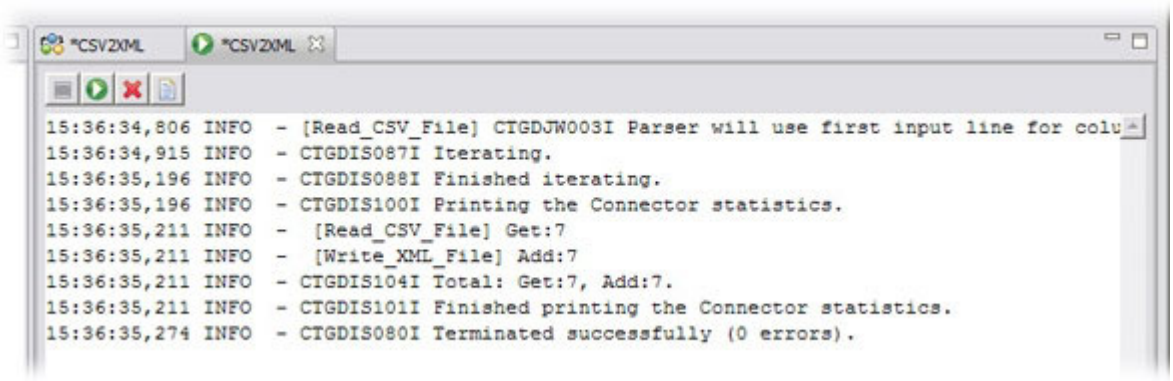

图 28. 运行 AssemblyLine 的日志输出

CE 实际上执行了 AssemblyLine 及其所有组件，并导出了*配置*，它是定义分配给 IBM Security Directory Integrator 服务器的工作的 XML 文件。然后它会将此配置输送给服务器，并指示服务器运行您的 AL，同时捕获要显示在屏幕上的所有日志输出。

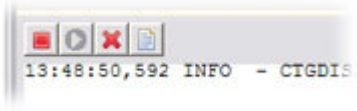

图 29. "日志输出"窗口的按钮栏

"运行"窗口包含一个按钮栏，其中包括用于停止 AL、一旦 AL 停止就将它重新启动以及清除日志内容的选项。最右边的按钮用于在外部编辑器中打开当前的日志输出。

AssemblyLine 的日志输出以包含的所有组件的统计信息结束，这些组件在您的例子中只是两个连接器。从以上信息中可清楚地看出，从 CSV 文件读取了 7 个条目且已将 7 个节点写入 XML 文档中。

您可以在磁盘上查找您的输出文件，并在浏览器窗口中将其打开以确认结果。您还可以使用数据浏览器，方法是右键单击输出连接器（Write_XML_File）并选择**浏览数**

据。

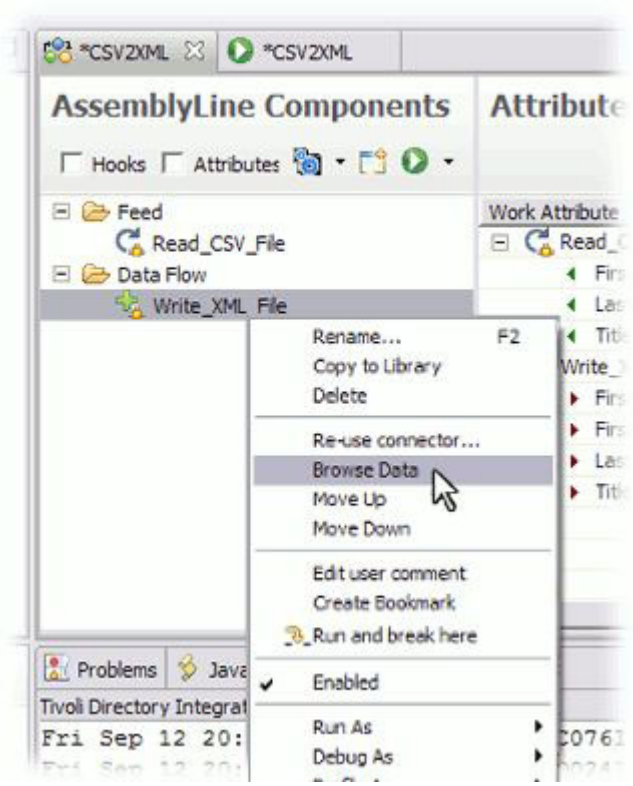

图 30. 浏览输出连接器创建的数据

此操作将为选择的连接器打开数据浏览器，该连接器已配置并已准备好运行。按**连接**按钮，然后单击**下一步**以读取并显示输出数据。

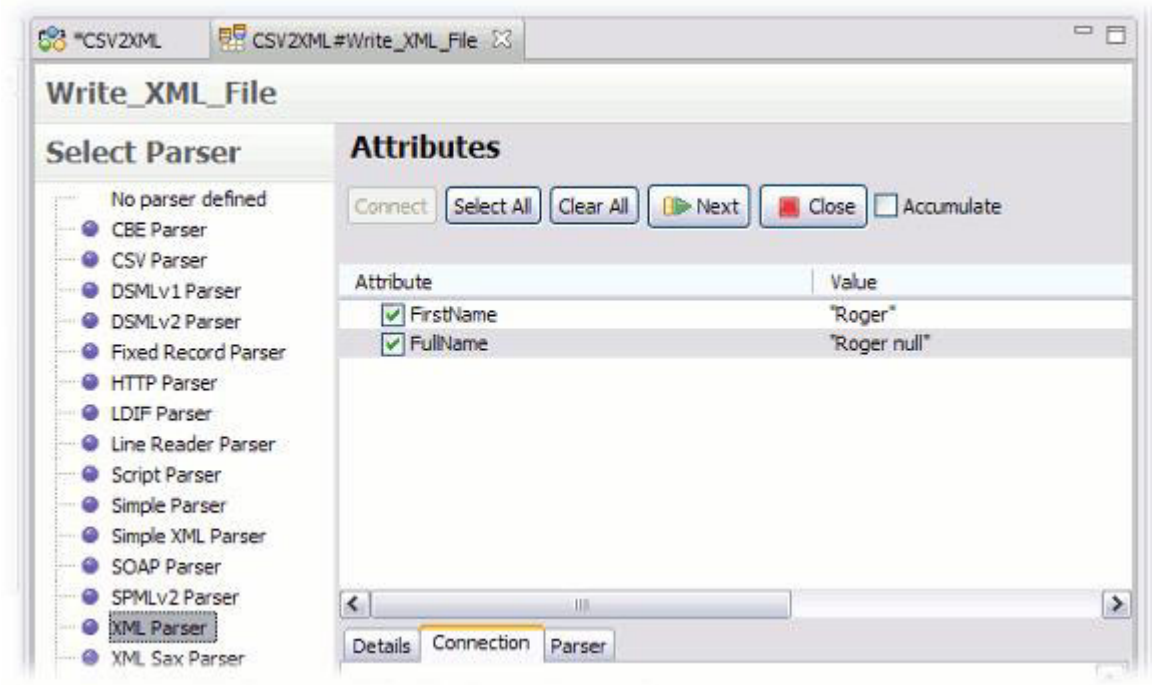

图 31. 浏览得到的 XML

输出 XML 应该是输入值的准确说明再加上您的映射逻辑；除第四个条目 (Roger) 之外的内容都正常，Roger 缺少“LastName”和“Title”，并且具有计算得出的以下“FullName”值：`Roger null`。

如果更仔细地检查输入数据，那么您将看到一个 CSV 行不完整：

```
First;Last;Title
Bill;Sanderman;Chief Scientist
Mick;Kamerun;CEO
Jill;Vox;CTO
Roger
Gregory;Highpeak;VP Product Development
Ernie;Hazzle;Chief Evangelist
Peter;Belamy;Business Support Manager
```

缺少输入数据和输入数据无效是常见的现象；在处理过程中，您需要准备解决方案来过滤或更正这种情况。

Null 行为：处理缺少的属性/值

要处理缺少的值，可以使用属性映射的内置 NULL 行为功能，也可以自己检测并处理此问题。

让我们从配置 Null 行为开始。通过以下方法执行此操作：右键单击“属性映射”面板中（而不是 AL 组件树形视图中）的“Read_CSV_File”连接器，从上下文菜单中选择 **Null 行为**[21]。

21. 或者，您也可以在 AL 组件列表中选择连接器本身；按“输入映射”顶部的**更多...** 按钮，然后在该处选择“Null 行为”。

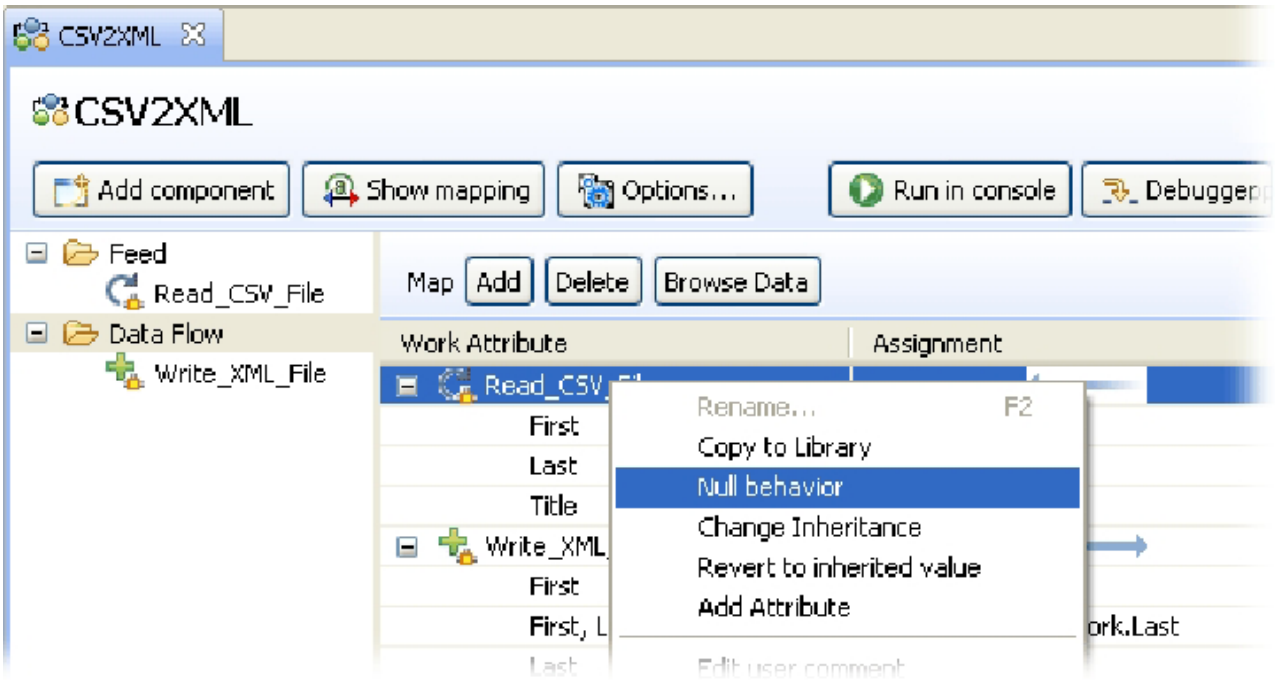

图 32. AL 级别配置的 Null 行为按钮

这将显示“Null 行为”对话框，您可以在其中配置如何定义 *null*（它可能根据您读取的源的类型而变化）以及应该如何处理此情况。缺省情况下，*null* 表示缺少属性，缺省处理方法是从映射操作中除去此属性。此操作的最终结果是在接收条目中将找不到具有指定名称的属性。

在“Null 行为”对话框中，请使用右边的单选按钮将 *null* 定义为空字符串值，然后使用左边的单选按钮指定在此例中要返回的缺省值“* missing *”。

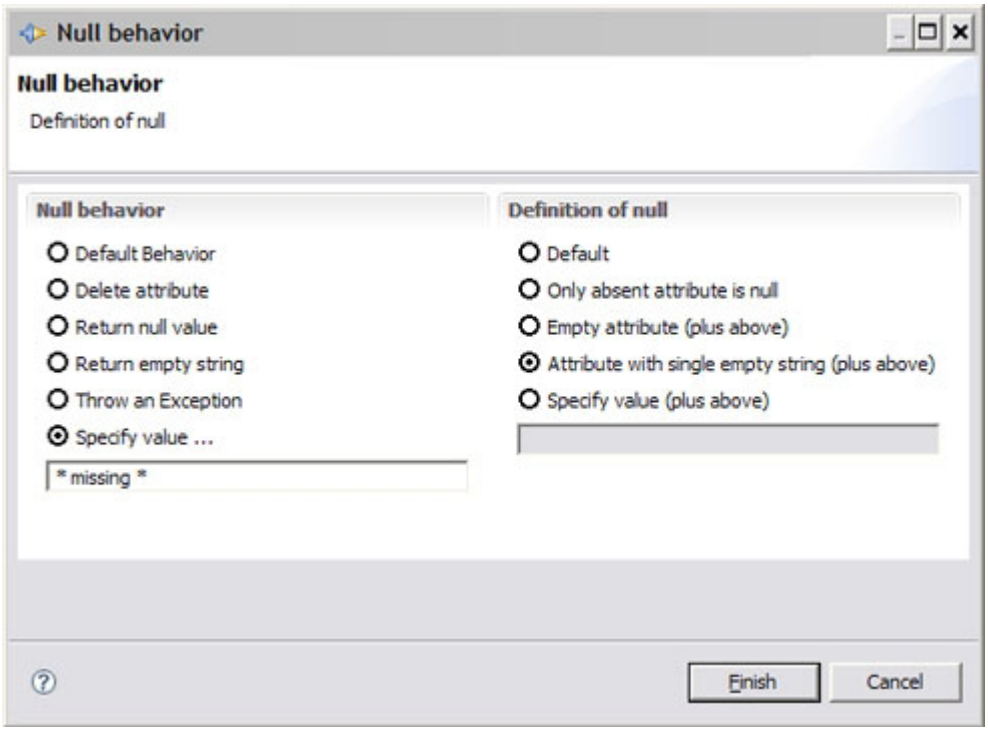

图 33. "Null 行为"配置对话框

重新运行 AssemblyLine 并刷新显示 `Output.xml` 内容的浏览器窗口。"Roger"条目的"Title"和"LastName"现在应具有特殊的 *null* 值（"* missing *"）。

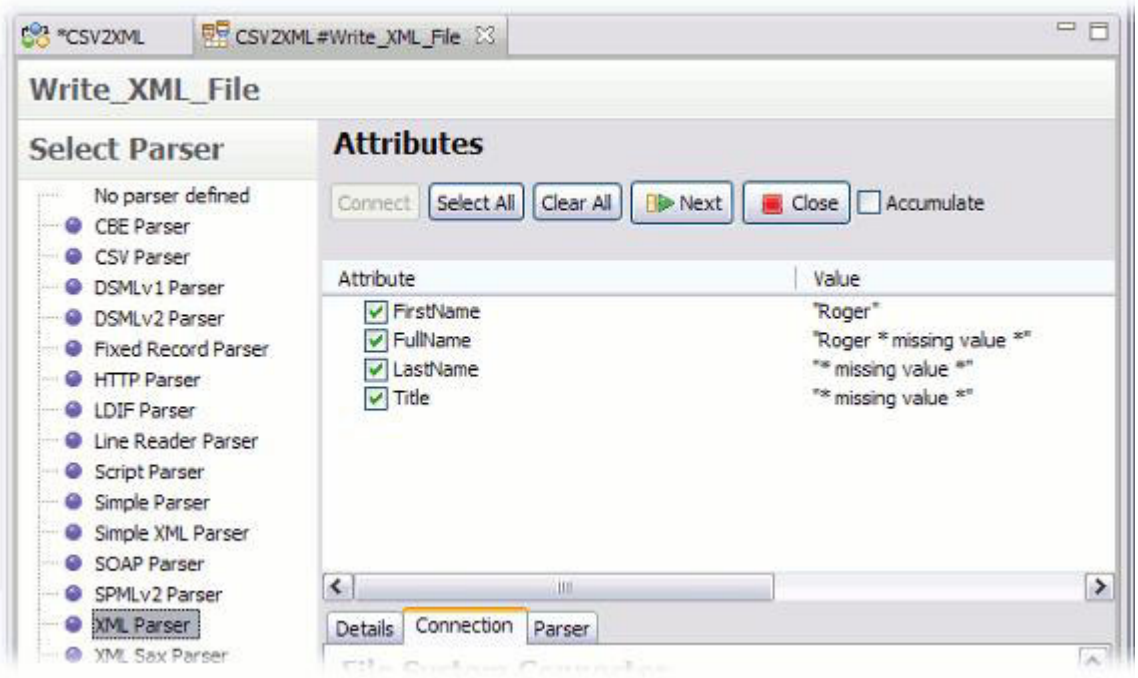

图 34. "NULL 行为"设置的 XML 输出中的结果

这在某种程度上较好 – 至少它是有意识的选择。但是，有时条目太不完整而无法继续处理。在我们的场景中，您至少需要“FirstName”和“LastName”的值才能计算“FullName”，因此您现在将过滤逻辑添加到 AssemblyLine 中以确保所有条目都满足此要求。

通过重新单击**插入组件**按钮，然后从左边的单选按钮选择**控制/流组件**启动。然后在列表中选择“IF”，将其命名为“Incomplete data”[22] 然后按**完成**。

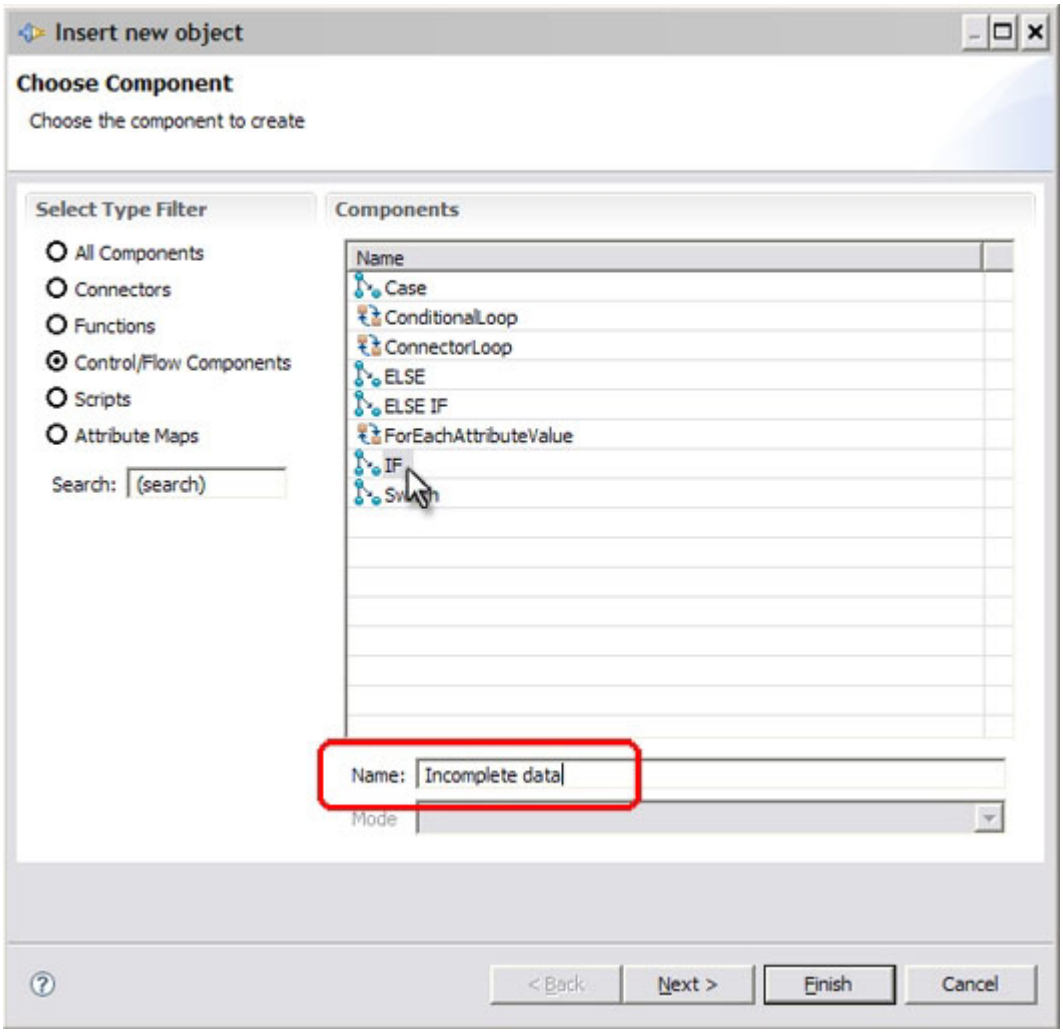

图 35. 选择 IF 分支组件

现在将此 IF 分支拖动到您的输出连接器上方。您可在 IF 分支编辑器区域中添加条件。

22. 先前鼓励您像命名脚本变量那样命名连接器。这也适用于功能组件。但是，对于属性映射组件、分支、循环和脚本，这不太重要，因为很少直接从脚本访问这些对象。在本指南中，更重要的是使 AssemblyLine 更易读。

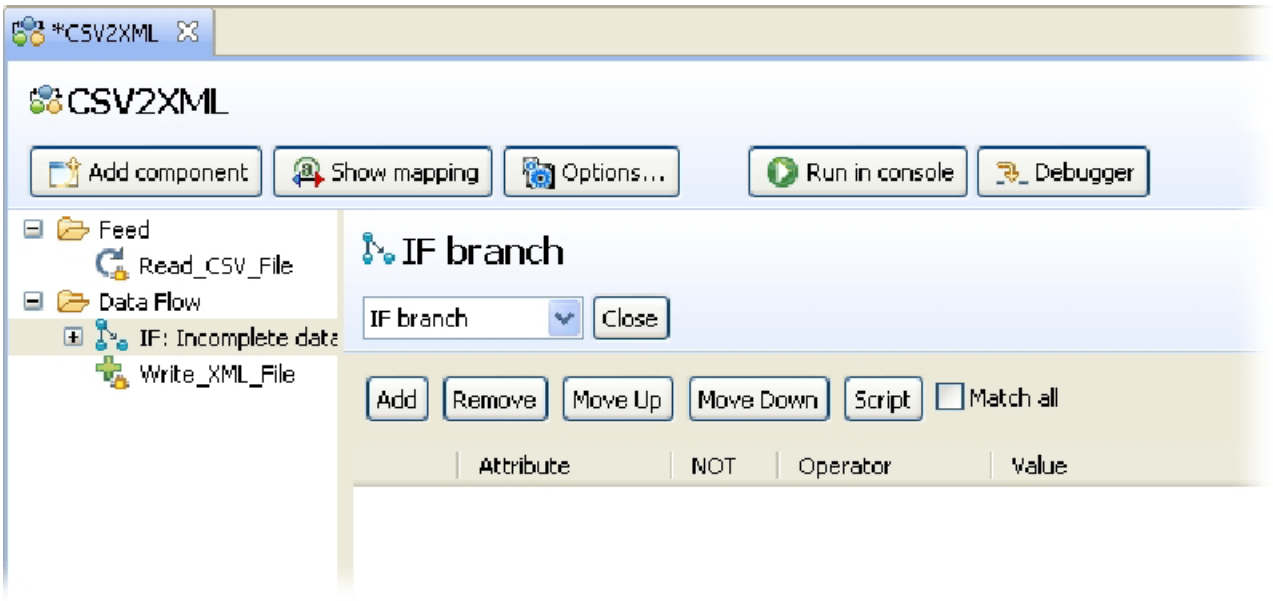

图 36. 编辑 IF 分支的条件

您可在此处选择添加简单条件和/或编写脚本片段。请注意，当取消选中决定是否对您的条件（简单条件和用脚本编写的条件）求值的**全部匹配**复选框时，条件之间具有隐式的 *OR*，当选中该复选框时，条件之间具有 *AND*。

按**添加**按钮添加简单条件。然后从最左边的下拉列表中选择“First”属性，然后选择“具有值”运算符。通过切换**非**列值对此条件取反。使用“具有值”运算符时，最右边的字段中不需要指定任何内容。现在也为名为“Last”的属性添加类似的*不具有值*条件。最后，确保不选中**全部匹配**复选框（暗示*任意匹配*），这样缺少任一属性的值都将触发此分支。

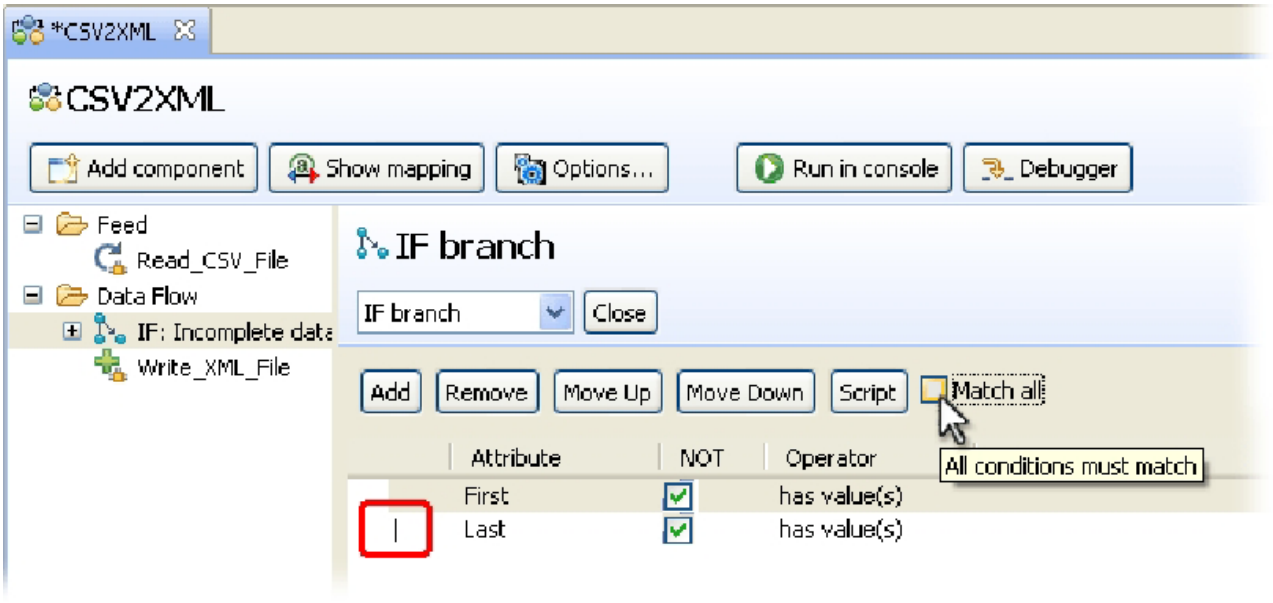

图 37. 向 IF 分支添加简单条件

只要指定条件求值结果为 *true*，IF 分支就会使 AssemblyLine 的执行流转向。在该例中，如果没有为“First”或“Last”属性赋值或者即使这两者中任何一个不存在于 Work 条目中，那么也将继续处理该分支下的组件。换句话说，“具有值”运算符还包括“存在性”检查。

现在展开 IF 分支，然后双击它下面显示的占位符以在此插入组件。在**选择组件**对话框中，单击**脚本**单选按钮，选择名为“Script”的脚本，然后按**完成**。将此脚本组件（也称为“SC”）重命名为“Write to log”，然后在其中输入以下 JavaScript 片段[23]:

```
task.logmsg("*** Skipping incomplete entry");
```

此处使用的 *task* 变量引用了 AssemblyLine 本身，并且它为您提供了大量有用的功能，如 `logmsg()` 方法。这一用脚本编制的调用将使指定文本消息写入日志输出。

要使日志输出更具参考性，请同时包括 Work 条目的当前内容。通过以下方法执行此操作：右键单击 IF 分支，选择**添加组件...**，然后再次选择**脚本**单选按钮。这次请选择标记为“转储 Work 条目”的预定义脚本组件。

最后，您必须指示 AL 在该位置停止当前循环，并将控制权返回给迭代器连接器，以使其能够在下一个 CSV 行中进行读取，这可有效地从 XML 输出中过滤当前条目。除非您自己指定此行为，否则控制权将继续转到 IF 分支后的第一个组件。因此必须再次右键单击 IF 分支，然后单击**添加组件...**，选择**脚本**，然后选择名为“Exit Flow”的 SC。现在您的 AL 应类似于:

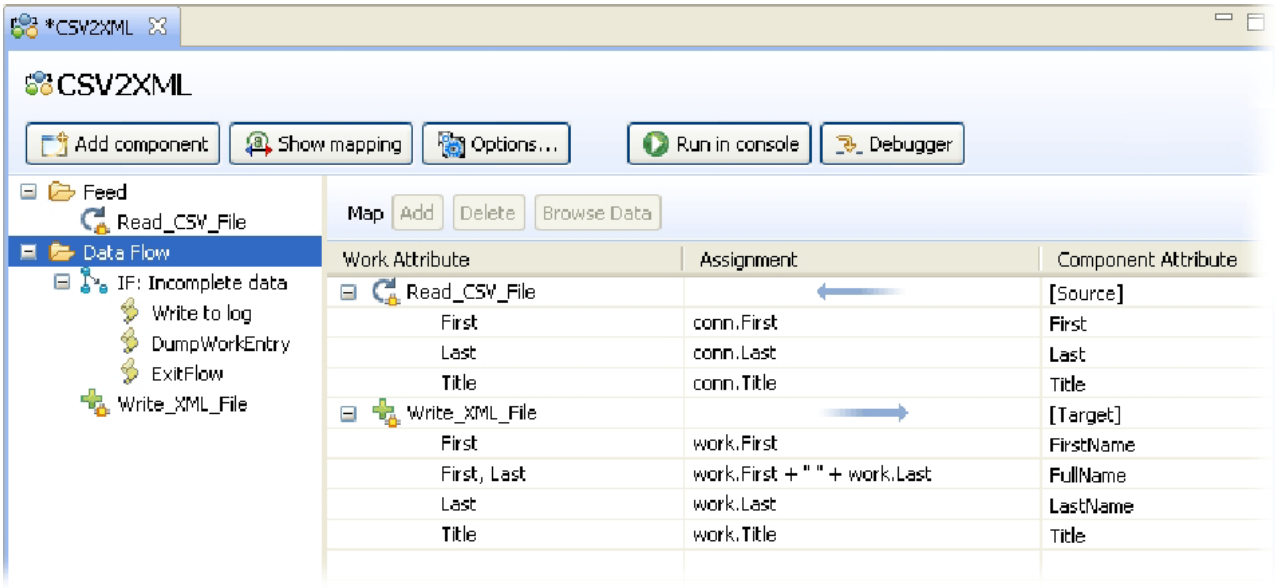

图 38. 第一个完整的 AssemblyLine

再次运行您的 AssemblyLine 之前，您将需要再次打开“Null 行为”对话框，并恢复缺省定义和行为选择，否则对您的 IF 分支条件求值永远不会得到 *true*。

23. 脚本编辑器提供称为*代码完成*的功能，该功能显示了您可以选择的选项。例如，在脚本编辑器中输入“tas”，然后按 Ctrl + Space 组合键以打开代码完成下拉列表，该列表中应该提供单个选项：**task**。如果按 Enter 键，那么将选择并在您的脚本中输入此选项。现在输入句点（.）使您的脚本变为“task.”，然后稍等片刻；您将看到一个新的代码完成下拉列表，这次该列表具有您可以在 `task` 对象中访问的所有方法和属性。

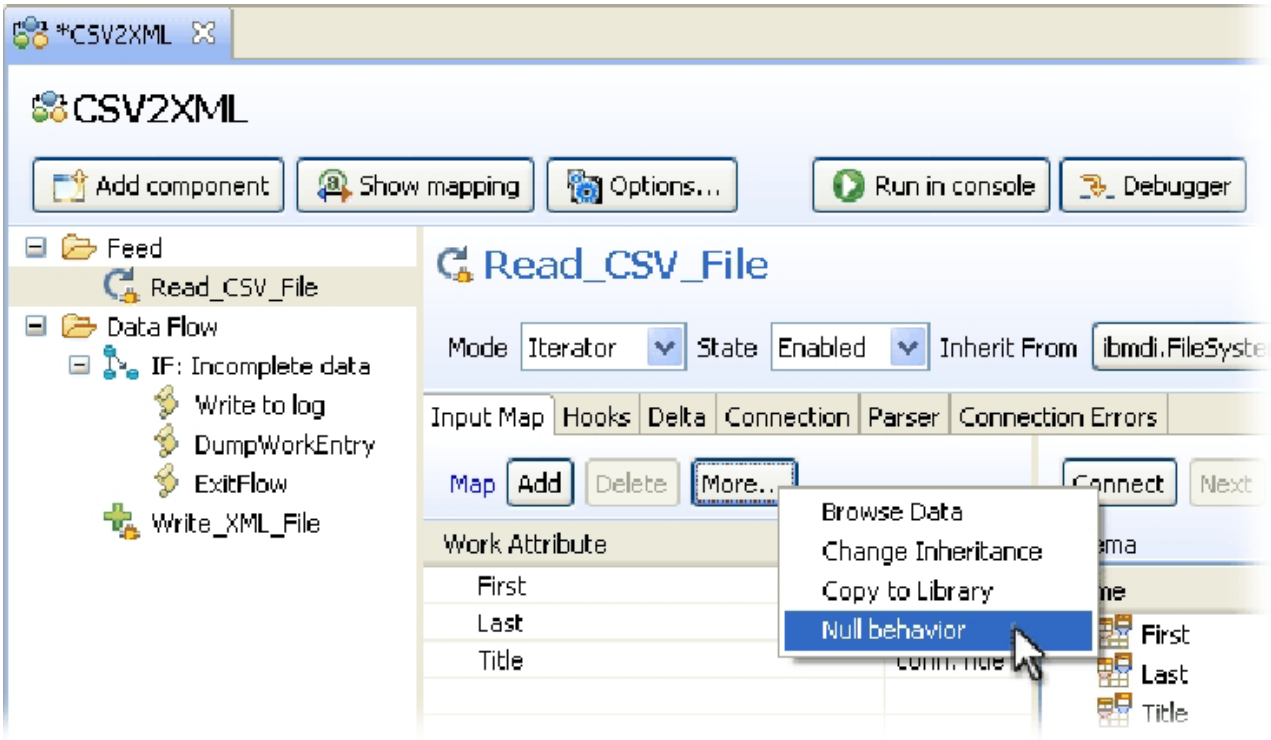

图 39. 复位 AssemblyLine 的 NULL 行为

现在当您再次运行 AssemblyLine 时，您将看到您的消息后跟有 Work 条目转储:

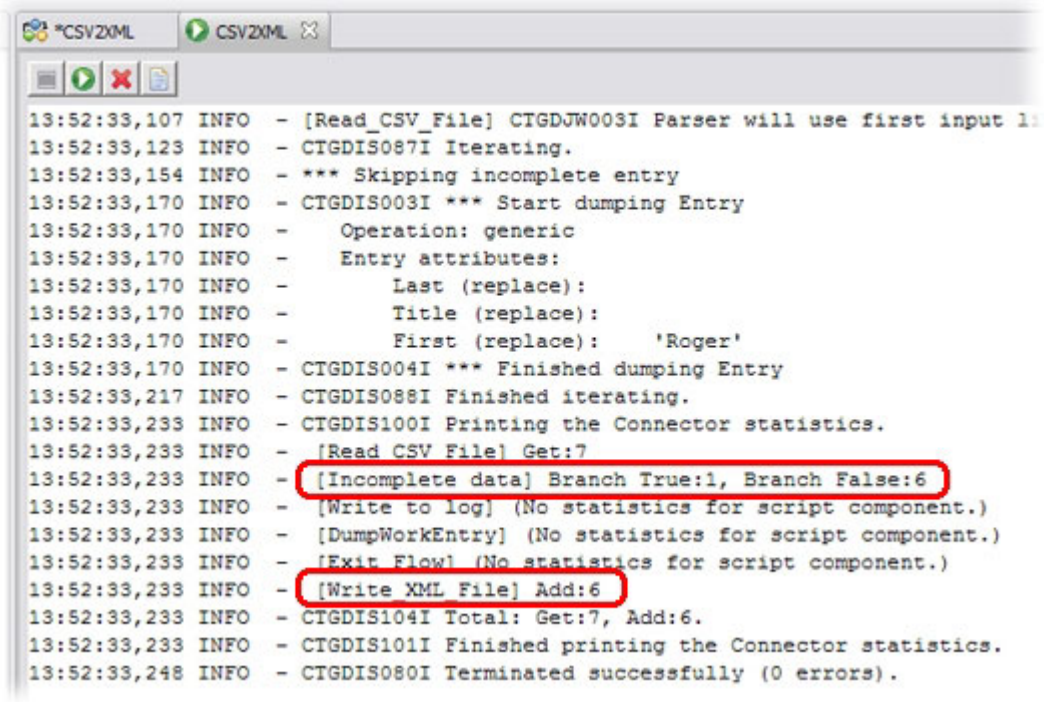

图 40. 包含消息和 Work 条目转储的日志输出

从统计信息中，您可以看到您的“数据不完整”IF 分支曾经为 true，这导致仅 6 个节点添加到您的 XML 输出中。如果再次刷新数据浏览器窗口，那么您将看到“Roger”确实已消失了。

调试 AssemblyLine

IBM Security Directory Integrator 最强大的一个功能部件是其内置的 AssemblyLine 调试器，您可通过该调试器完成对 AL 的执行过程，同时查看甚至修改正在处理的数据。

通过单击**启动调试**会话按钮单步执行您的第一个 AssemblyLine 。

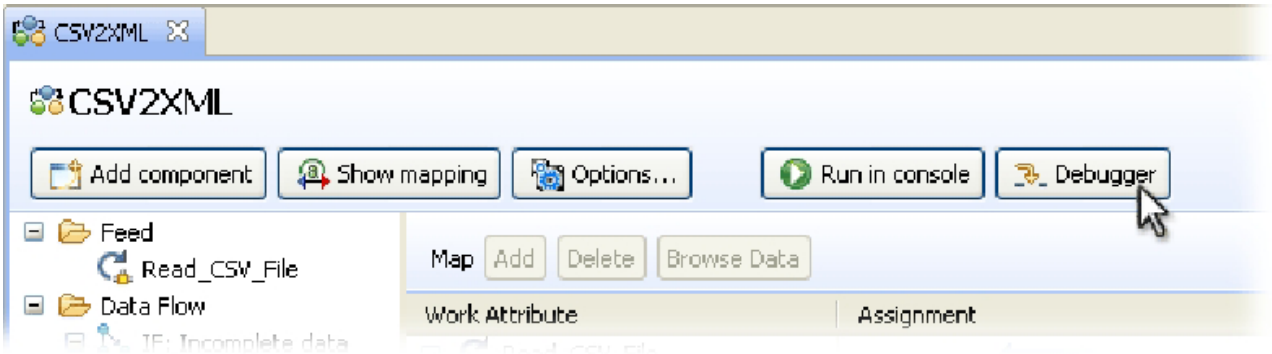

图 41. 调试 AssemblyLine

您将发现自己处于调试器中，而不是在标准的“运行控制台”窗口。

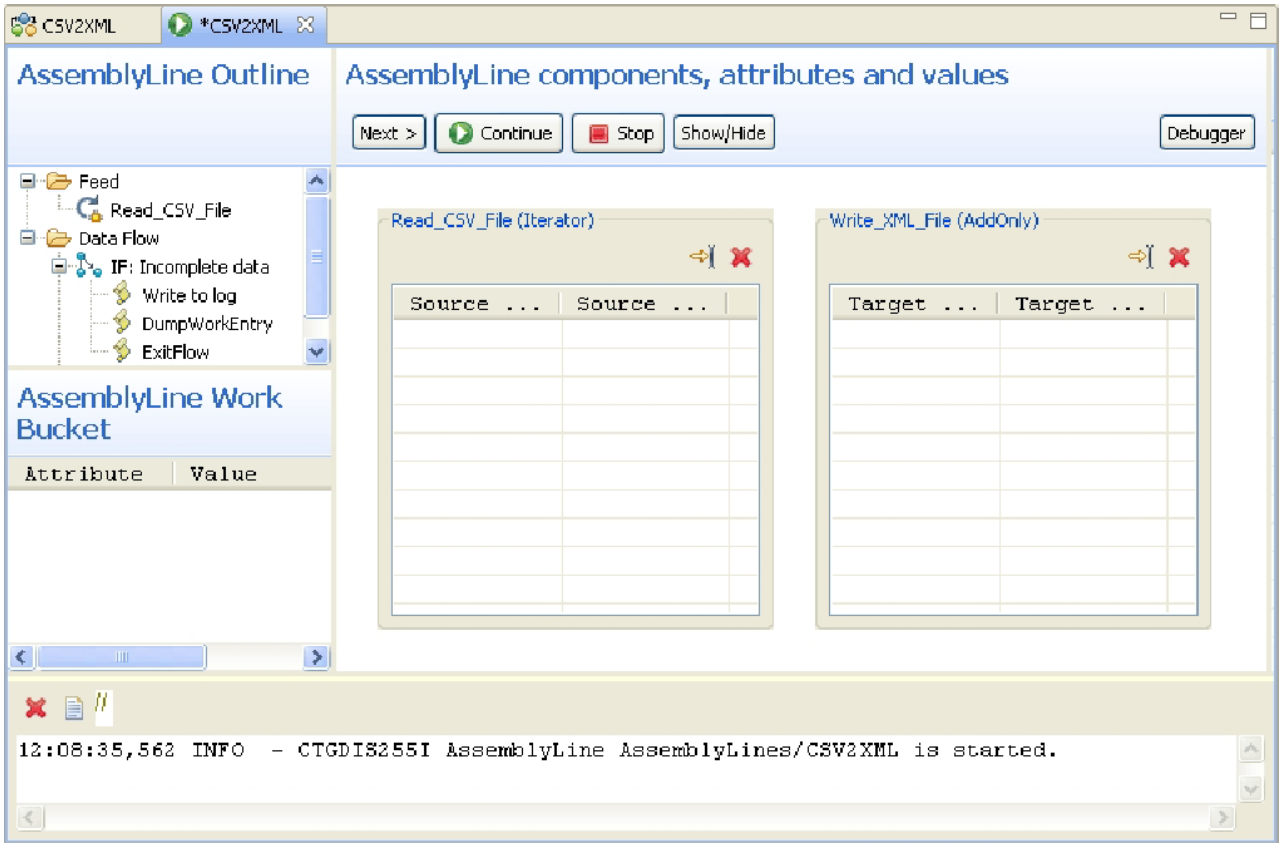

图 42. AssemblyLine 数据步进器

AssemblyLine 调试器提供两种方式: 数据步进器（提供简单直接的测试功能）; 高级 AL 调试器，其中您可以更深入挖掘 – 如步进脚本，与 Java 库交互性协作并修改文件中的数据。

数据步进器是用于步进执行 AssemblyLine 连接器以及查看读取、编写和变换的数据的有用工具。该屏幕分为三个主要区域:

- **AssemblyLine 大纲**显示您的 AL，突出显示暂停执行的位置;
- **AssemblyLine 工作存储区**，其显示映射到 Work 条目中的所有属性 – 即在输入映射或属性映射组件中找到的属性。
- **AssemblyLine 组件、属性和值**，其中您有一个按钮行用于控制调试会话，以及一组数据显示网格用于 AL 中所有连接器。

沿着底部，您可以看到控制台输出窗口，其中显示使用“运行”按钮运行 AssemblyLine 时获取的同一信息。

此时，AssemblyLine 已分派到测试服务器，并已准备好根据您的命令开始运行。按“下一步”按钮以开始步进。

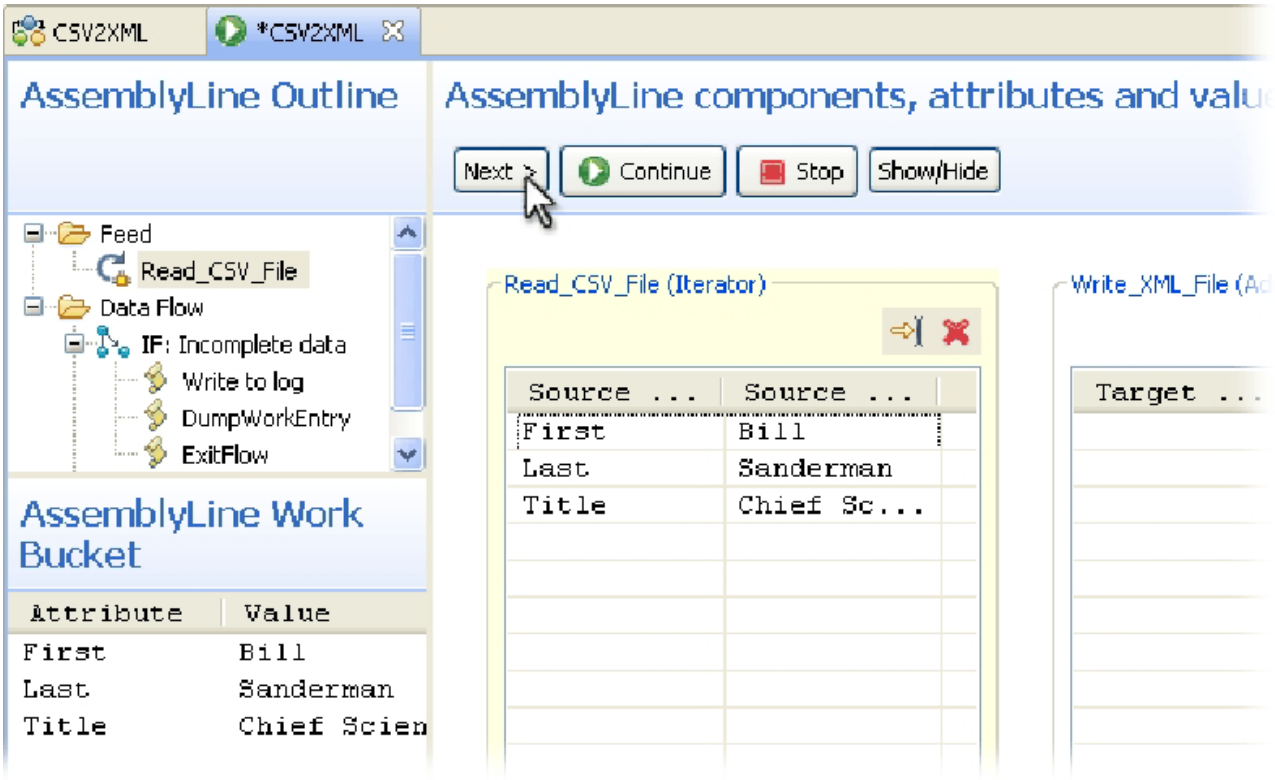

图 43. 步进到 AL 运行

请注意三个事件在屏幕上如何发生：**AssemblyLine 大纲**显示“Read_CSV_File”连接器当前处于活动状态；**AssemblyLine 工作存储区**显示该连接器刚读取的属性；还用这些属性填充了该连接器的数据显示网格。每次按**下一步**按钮，都会继续执行步骤并刷新信息显示。

您还可以使用连接器的数据网格顶部的**运行到此处**按钮以跳至该点。现在，为“Write_XML_File”连接器执行该操作。

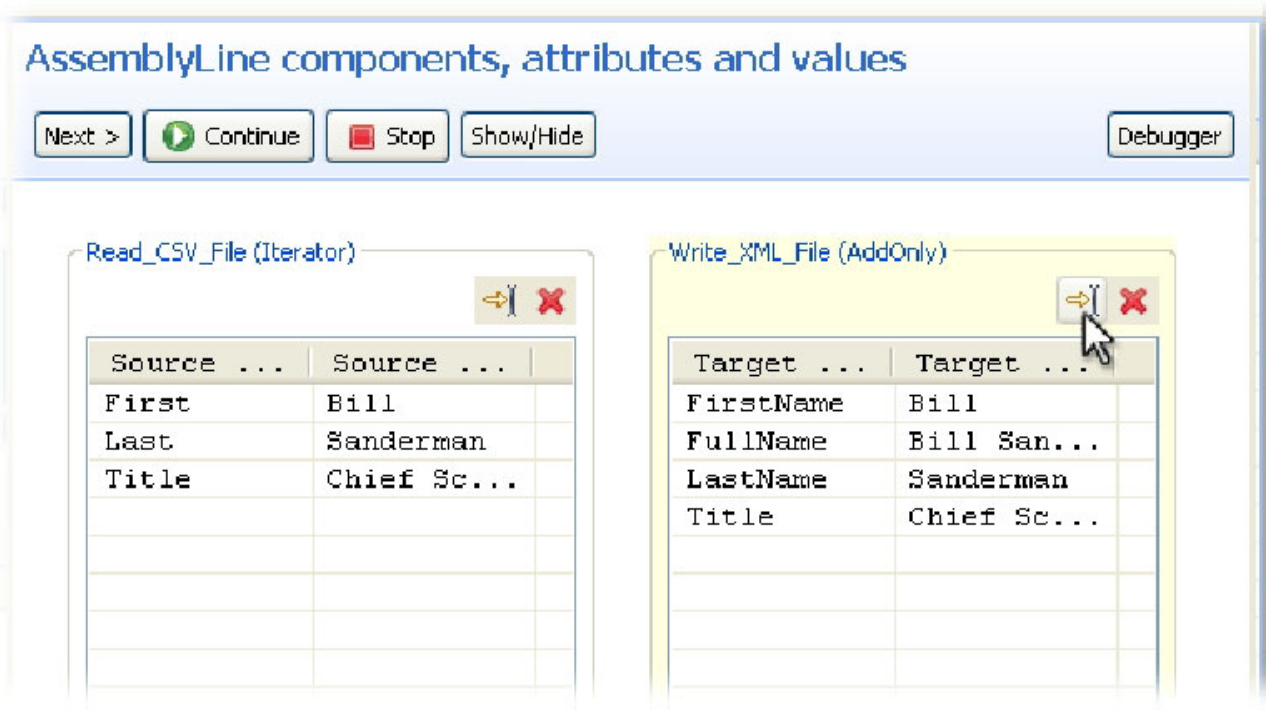

图 44. 步进到 Write_XML_File 连接器

然后，该数据网格在该连接器的输出映射中显示属性及其值 – 包括“FullName”的计算值。

现在，我们看看数据步进器工具栏按钮以查看您具有什么选项:

- **下一步 >** 将处理移到下一组件并更新所有数据显示区域;
- **继续**将使 AL 继续执行到完成。
- **停止**将立即终止运行;
- **显示/隐藏**让您决定显示哪些连接器数据网格;
- **调试器**将您切换到完整调试器方式，在该方式中，您可以步进浏览脚本代码，设置断点，查看和修改任何属性和脚本变量，并在运行 AssemblyLine 的上下文中交互执行 JavaScript 命令。

虽然数据步进器提供关于 AssemblyLine 执行方式的大量信息，但是有时您需要高级调试器的增强功能。请注意，在调试会话期间，您随时可以根据需要在步进器与高级方式之间进行切换。现在通过按位于“数据步进器”按钮行最右侧的**调试器**按钮来予以尝试。

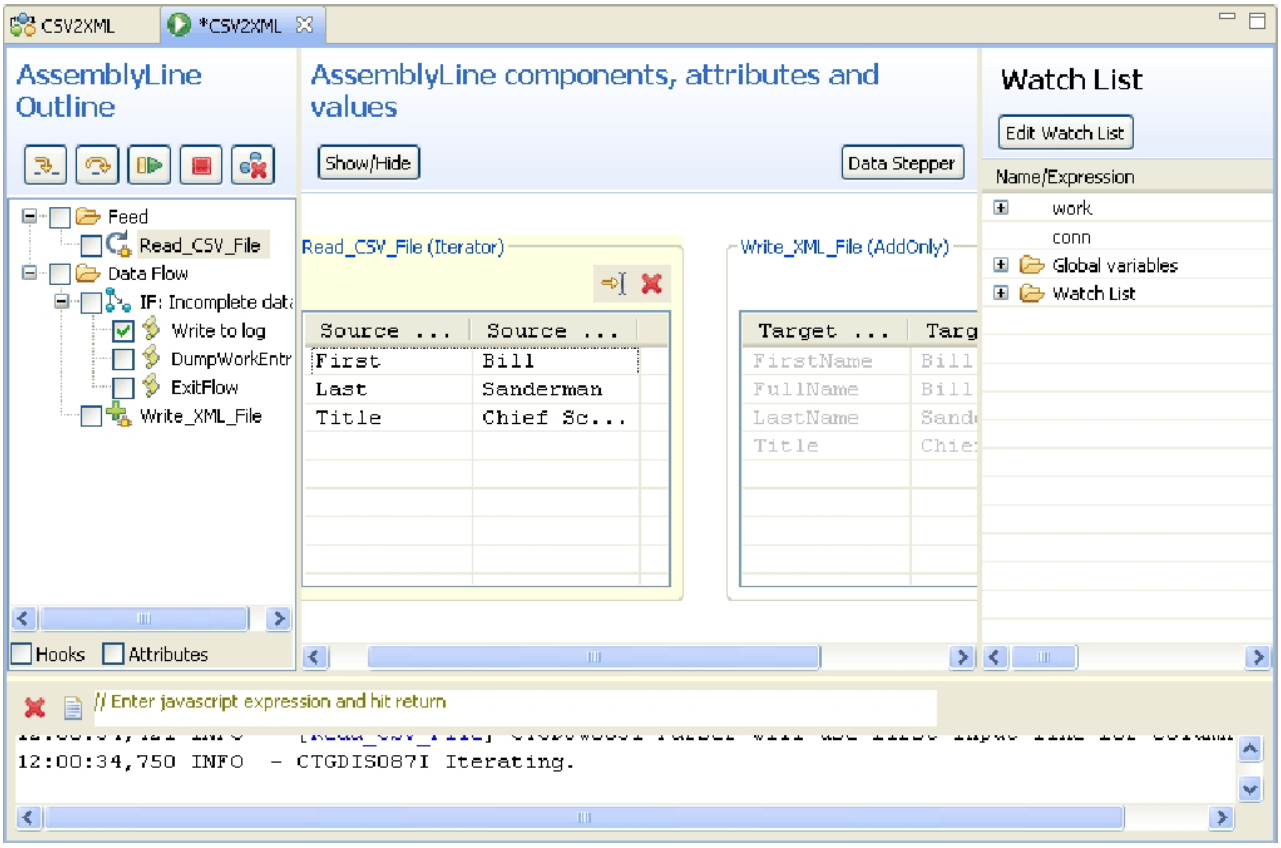

图 45. 高级调试器方式

切换方式时，将重新绘制屏幕以提供 **AssemblyLine 大纲**的新控件，并将 **AssemblyLine 工作存储区**替换为窗口右侧的**监视列表**。**监视列表**显示标准属性*存储区*：work 和 conn。还存在称为“Global variables”的文件夹，如果打开，其中会显示为 AssemblyLine 定义的所有变量：内置变量（如 **work** 和 **system**）以及用脚本代码定义的任何变量。最后一个 Watch 文件夹供您自己使用，可以通过使用该面板顶部的**编辑监视列表**按钮来添加要监视的变量或整个 JavaScript 表达式。

将我们的关注点转向 **AssemblyLine** 大纲，该树形视图中组件旁的框称为“断点”，在执行期间，您可以通过单击其中一个断点来指示 IBM Security Directory Integrator 暂停任何组件。还可以在任何节点上右键单击并选择**运行和在此中断**，以使 AL 执行到该点为止。大纲之上的工具栏提供的一些控件与数据步进器中的相同，还提供了一些新控件：

图 46. “调试器”按钮

从左边开始，这些按钮为:

- **步进**，它允许您步进属性映射、脚本甚至 AssemblyLine 及其组件的底层工作流程。内置流中的这些停留点称为“*挂钩*”，涵盖在稍后的练习中;
- **跨越**与您在数据步进器中看到的**下一步 >** 按钮相同。在调试器中，它还让您跨越脚本函数调用，而不是进入其中;
- **继续**会使 AL 运行到完成为止（就像在数据步进器中一样）或达到断点为止;
- **停止**将停止 AssemblyLine，就像在数据步进器中一样;
- **清除所有断点**将除去为 AL 设置的任何断点。

要感受断点如何工作，请尝试通过单击该组件旁的框来为“Write to log”脚本设置一个断点。

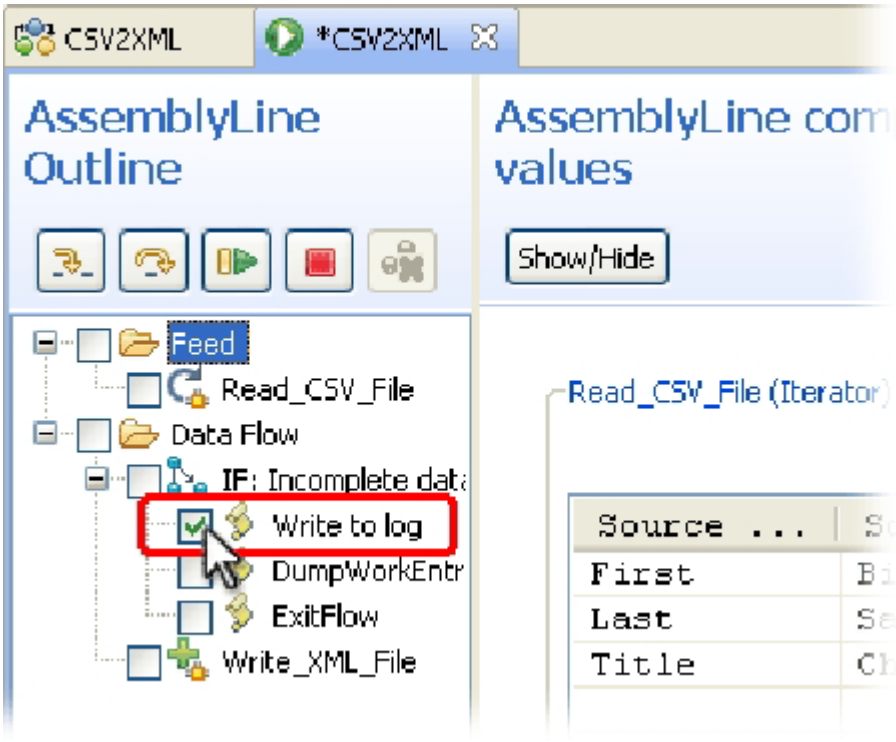

图 47. 设置断点

现在，按**继续**按钮，AL 就将运行到 IF 分支为 *true* 为止，此时您会获取对“Write to log”脚本的控制。IBM Security Directory Integrator 还会打开脚本区域，让您在此步进浏览代码。您甚至可以通过双击任何脚本行左侧的空白来在该行处设置断点。

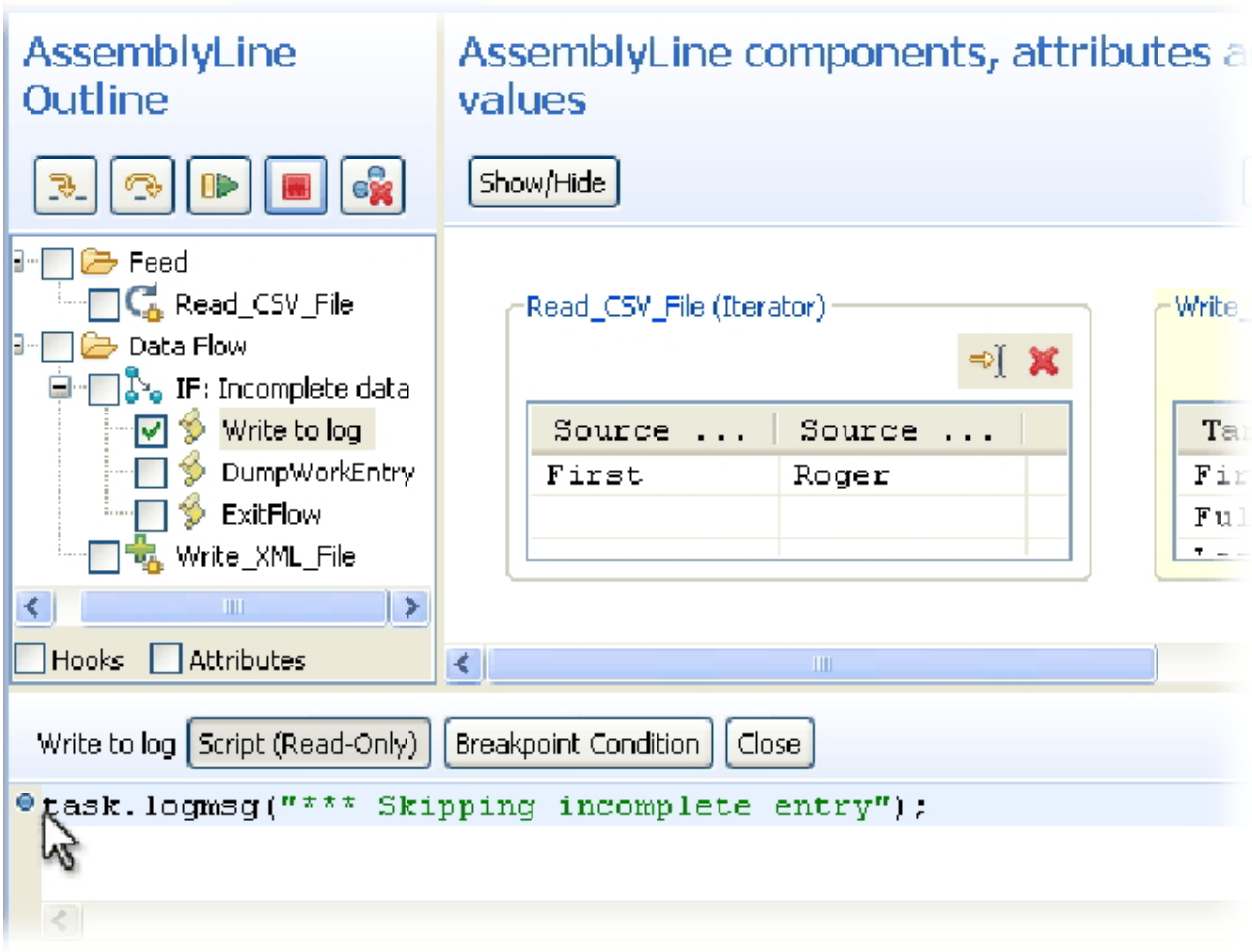

图 48. 在脚本中设置断点

而且，您可以双击组件列表中任何节点以产生调试显示。如您在上图中所见，有一个标题为**断点条件**的按钮。您可以将此用于设置 JavaScript 表达式，该表达式必须求值为 *true* 或 *false* 且将确定断点是否处于活动状态。例如，如果满足以下条件，以上所示断点就可以设置为 true:

```
work.First.startsWith("R")
```

或

```
mycounter > 1000
```

这对于调试只会在某个输入数据集内部深处发生的问题非常方便。

如果由于某种原因，您需要返回到上一步，那么只需停止并重新启动调试会话即可。还可以通过按“数据步进器”按钮来切换回数据步进器。

但是在离开高级调试器之前，还有一个值得注意的功能：**JavaScript 评估命令行**。

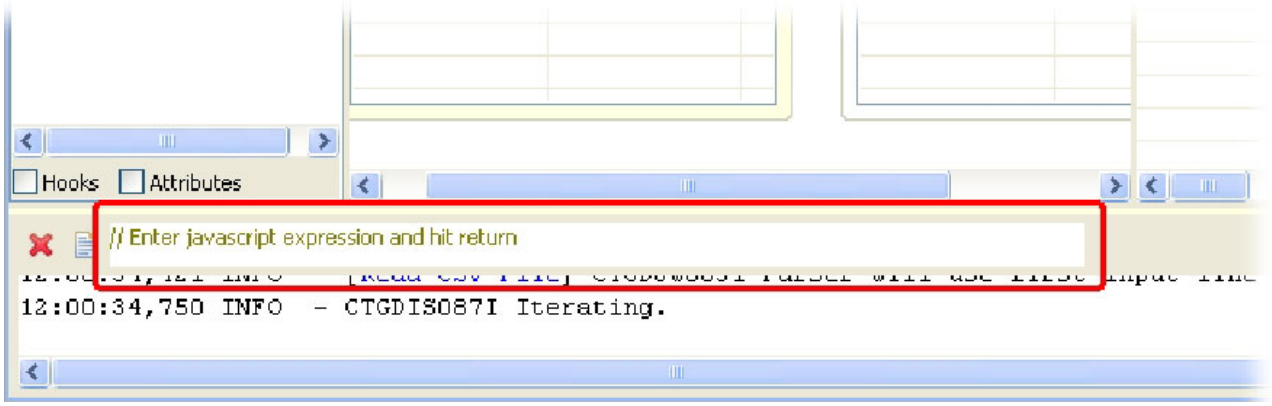

图 49. JavaScript 评估命令行

日志输出区域之上这一看似无用的输入字段允许您在运行的 AssemblyLine 的上下文中执行任何脚本片段。现在，通过输入该命令并按 Enter 键来予以尝试:

```
task.dumpEntry(work)
```

该操作将在日志输出窗口中显示 Work 条目的内容。现在尝试以下内容:

```
i = 42
```

在日志中将会出现以下消息: `i=42 >> 42.0`

该消息表明您已用值 42 定义了新变量（“i”）。表达式自身求值（所有脚本语句均如此）为分配的值。您还可以更改已经在 AL 中定义的变量和属性的值，例如:

```
work.First="Rudy"
```

执行该行后，“First”属性的值将为“Rudy”。有了在执行过程中修改数据的能力，您就可以确保 AssemblyLine 步进到所有分支逻辑中，从而允许您彻底地测试解决方案。

强烈建议您花一些时间来熟悉 AssemblyLine 数据步进器和调试器。这样不但可以从独特视角了解 AL 运作的方式（包括 IBM Security Directory Integrator 服务器内核提供的所有内置工作流程），而且还将有助于您验证自己关于数据的实施和设想。

从连续源中查找数据

继续学习我们的教程场景，下一步是添加在 D2 中进行的查找。

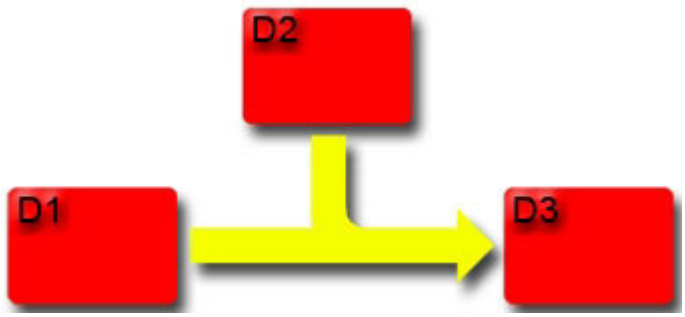

图 50. 场景流程图

Tutorial 目录包含名为 PhoneNumbers.xml 的文件，该文件将充当 D2 数据源。该文件保存了一系列 XML 条目，每个条目具有两个属性: “User”和“telephoneNo”。

您的作业将是把“telephoneNo”作为写入输出 XML 文档的数据的一部分包括在其中。由于您无法随机访问该文本文件以像对数据库或目录那样执行查找，所以将通过对文件执行循环查找并将当前 CSV 条目中的“User ”与“FullName”相比较来查找每个 CSV 条目的相应电话号码。

但是，在“Write_XML_File”连接器的输出映射中，将首先计算“FullName” - 换句话说，比较操作执行得太迟了。这表示您必须将该已计算的属性从“Write_XML_File”的输出映射移动至“Read_CSV_File”的输入映射。首先将“属性映射”项从一个映射拖至另一个映射来执行此操作。

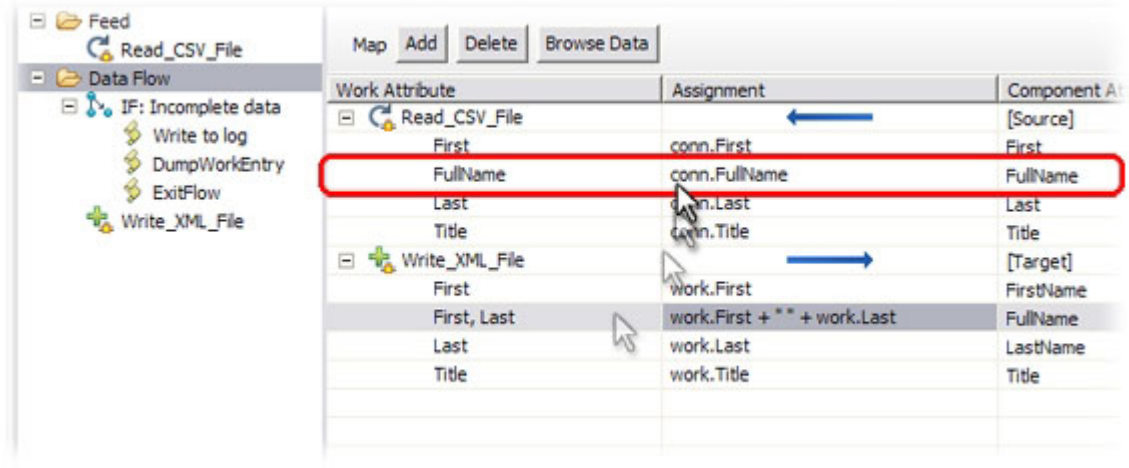

图 51. 将“FullName”拖至迭代器连接器的输入映射

此时两个映射中均有“FullName”映射项。因为此时要从 Conn 条目映射至 Work，所以需要调整“输入映射”赋值，而不使用对于“输出映射”所用的其他方式。通过双击“Read_CSV_File”下的“FullName”并将赋值更改为以下内容来执行此操作:

```
conn.First + " " + conn.Last
```

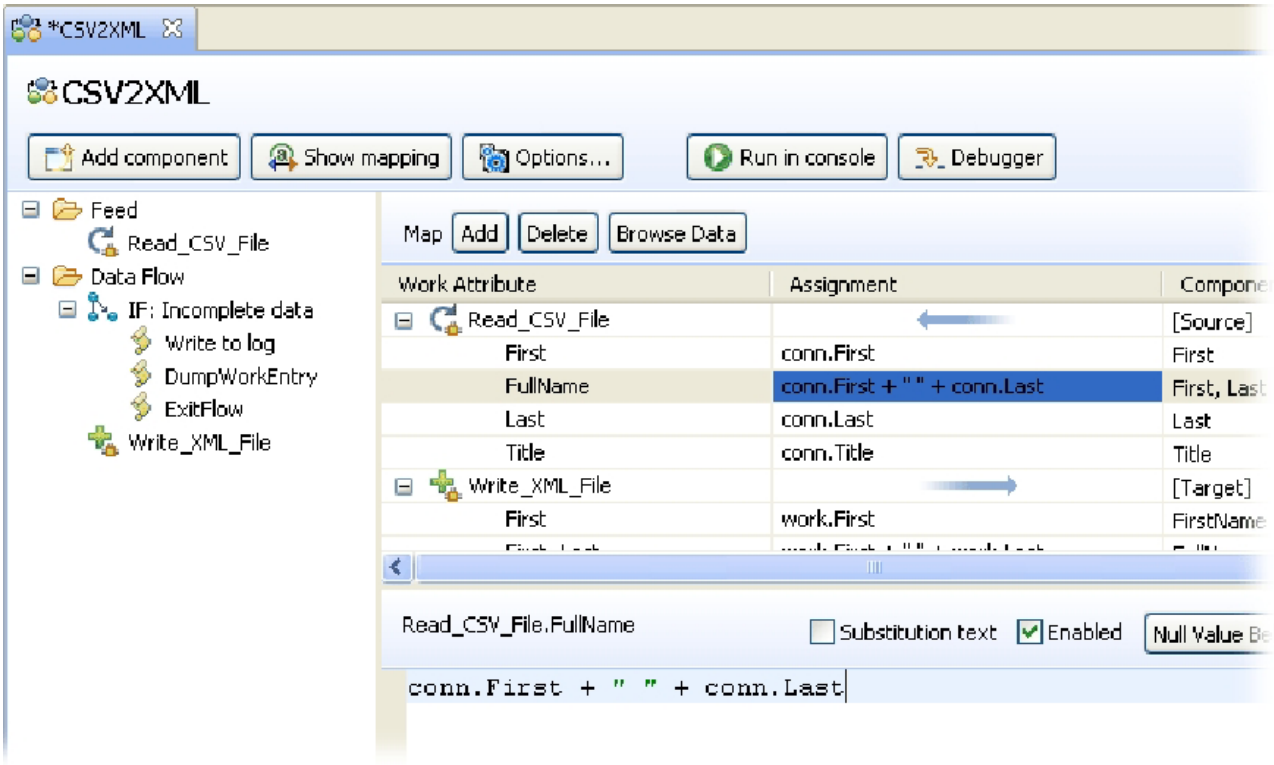

图 52. 编辑“FullName”的赋值

由于修改了输入映射，该属性此时才可用于 Work 条目，因此您还可以编辑原始的“FullName”输出映射项，以使其赋值只是 `work.FullName`。

现在重新运行 AssemblyLine，并检查 `Output.xml` 以确保其未更改。一旦确认这一点，此时将使用*循环*组件通读整个 `PhoneNumbers.xml` 文件，并搜索每个用户的编号[24]。

通过将新组件添加至**数据流**部分开始，这次选择名为 *ConnectorLoop* 的组件，然后将该组件命名为“TelephoneNumber”。ConnectorLoop 是一个循环组件，它使用连接器从数据源读取信息，然后对该连接器返回的每个条目将其下连接的所有组件循环一次。这类似于“订阅源”部分中迭代器连接器的*针对每个*行为，该连接器对读取的每个条目循环执行**数据流**部分中的组件。

将新的“TelephoneNumber”ConnectorLoop 拖至 IF 分支和“Write_XML_File”连接器之间。请确保该 ConnectorLoop 不会在 IF 分支*内部*结束。

24. 有三种类型的循环组件：1）*ConnectorLoop*，该组件使您能够对连接器以“迭代器”或“查找”方式所返回的数据进行循环。这是将在本练习中使用的循环类型；2）*ForEachAttributeValueLoop*，使用该组件可方便地对多值属性的值进行循环，例如在 Lotus Notes 和 LDAP 目录之类的系统中查找的值；3）*ConditionalLoop*，该组件使用简单条件和用脚本编写的条件（就像分支所使用的条件）来控制循环。

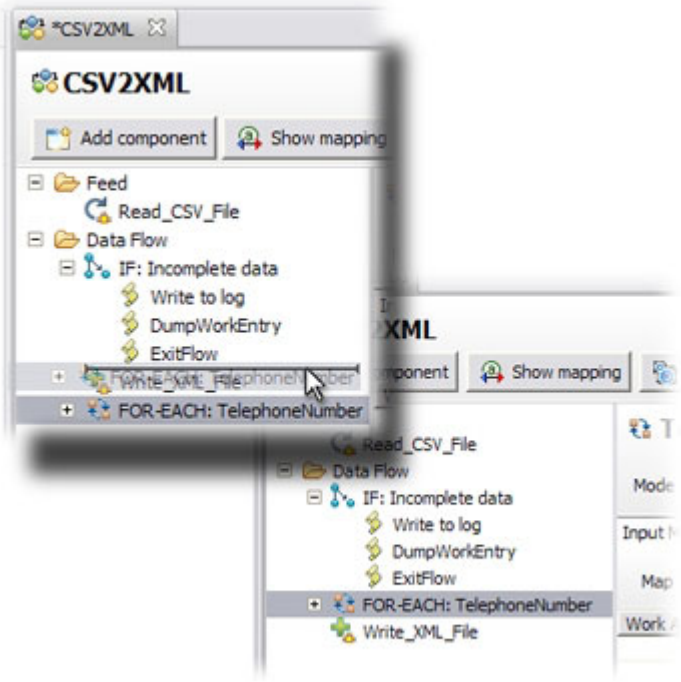

图 53. 拖动 ConnectorLoop

此时选择该组件以打开其编辑器，该编辑器类似于连接器编辑器。

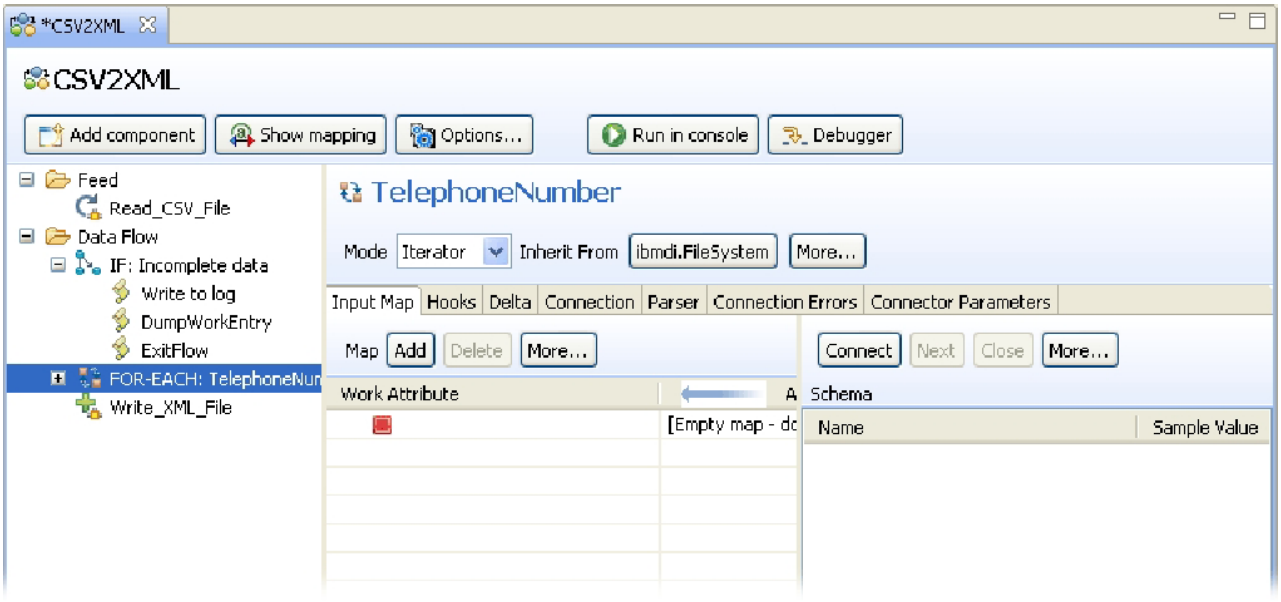

图 54. ConnectorLoop 配置

主要区别在于**方式**下拉列表仅包含**迭代器**和**查找**选项。此外还有一个**更多...** 按钮（它提供用于限制循环条目的选项）以及包含以下三个选项的**初始化**下拉参数:

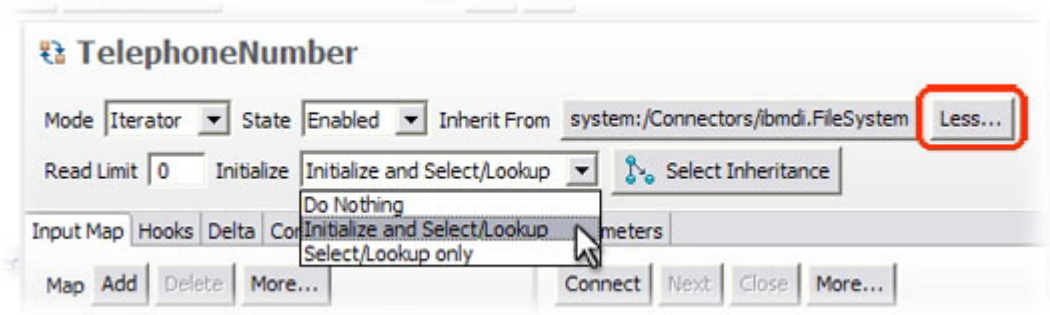

图 55. ConnectorLoop 高级设置

- **不执行任何操作**，这表示如果在 AL 处理期间到达该循环，那么在任何情况下都不会初始化其嵌入式连接器;
- **初始化并选择/查找**，选择此选项时，只要循环开始进行，就会初始化连接器。由于 ConnectorLoop 将从某个文件执行读操作，且您希望每次都从头开始该 ConnectorLoop，所以请使用此选项;
- **仅选择/查找**，当 ConnectorLoop 指向数据库、目录或某些其他可随机访问的数据源时，此选项很有用。对于这种情况，不必每次都重新初始化连接。只需重新发出搜索即可，对于“迭代器”方式，执行*选择*操作，对于“查找”方式，执行*查找*操作。

配置 LoopConnector（缺省情况下类型为“FileSystem”）以读取 `PhoneNumber.xml` 文件，然后选择“XML 解析器”。此时将打开“属性映射”选项卡以发现属性。

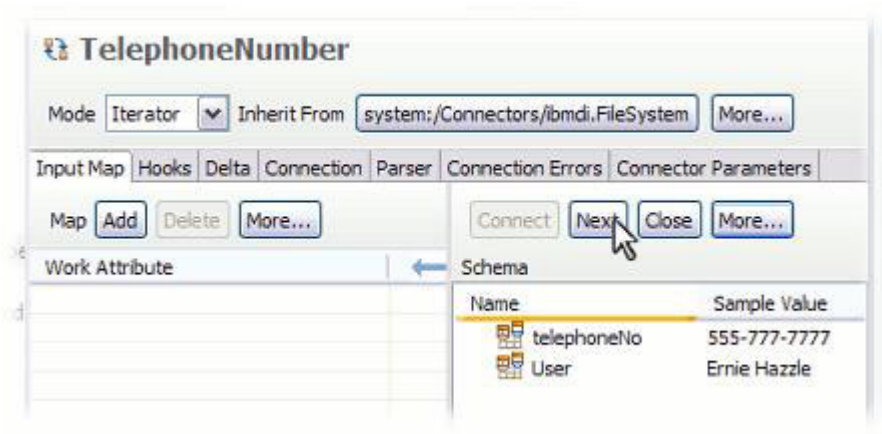

图 56. 分层属性

通过在“输入模式”中选择“User”和“telephoneNo”并将它们拖至“输入映射”可以执行属性级别的映射。

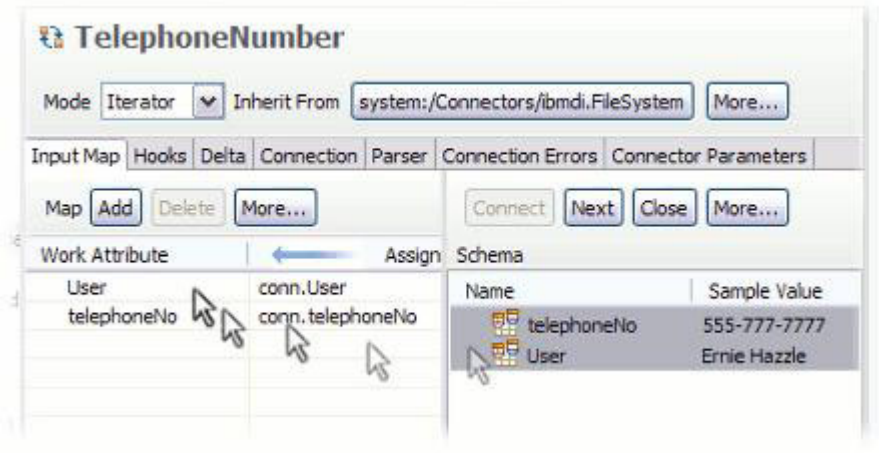

图 57. 从“模式”拖至“属性映射”

这样，对于名为“User”和“telephoneNo”的属性，您将各有一个映射规则。

此时可以关闭 LoopConnector 编辑器，然后通过右键单击 ConnectorLoop 并选择**添加组件...** 以在其下添加一个 IF 分支。将该 IF 分支命名为“找到匹配名称”。现在添加一个简单条件以检查“User”是否等于“$FullName”[25]。

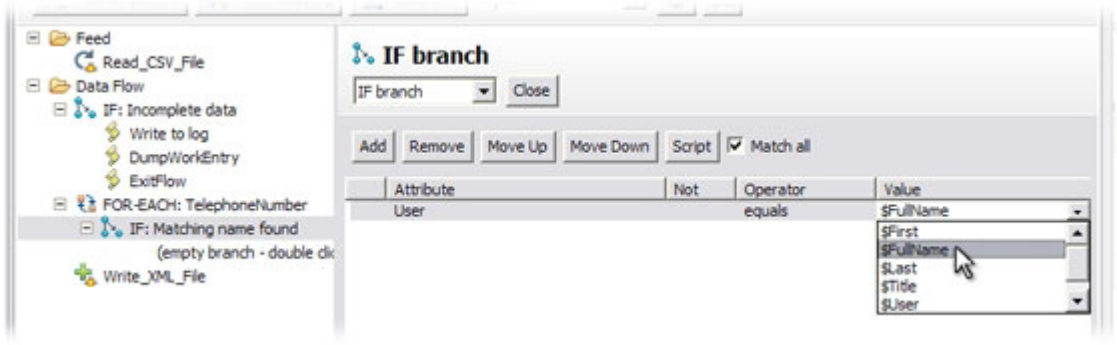

图 58. IF 分支的条件编辑器

只要找到匹配项，您就会希望 ConnectorLoop 退出且“User”和“telephoneNo”属性中有正确的值。要执行该操作，请添加您命名为“退出循环”的脚本组件并将以下脚本写入其中:

```
system.exitBranch("loop");
```

但是如果 ConnectorLoop 到达 PhoneNumbers.xml 末尾而未找到匹配项，将发生什么情况？“User”和“telephoneNo”属性包含从文件的最后一个条目中读取的值，因此仅检查空属性没有用处。您需要想出其他办法来检测失败的匹配。

答案是将脚本变量用作标志以指示是否找到了匹配项。现在通过将您命名为“找到用户”的脚本组件插入 IF 分支内，并将其拖至紧靠“退出循环”SC 之前的位置可以执行该操作。该脚本组件应该包含以下脚本片段:

```
foundUser = true;
```

25. 此处使用的美元符号是一种特殊字符，表示“FullName”不是要匹配的文字串，而是在 Work 条目中找到的属性的值。

要指示已到达输入文件末尾但未找到匹配项，只需选择 ConnectorLoop 并打开“挂钩”选项卡即可。

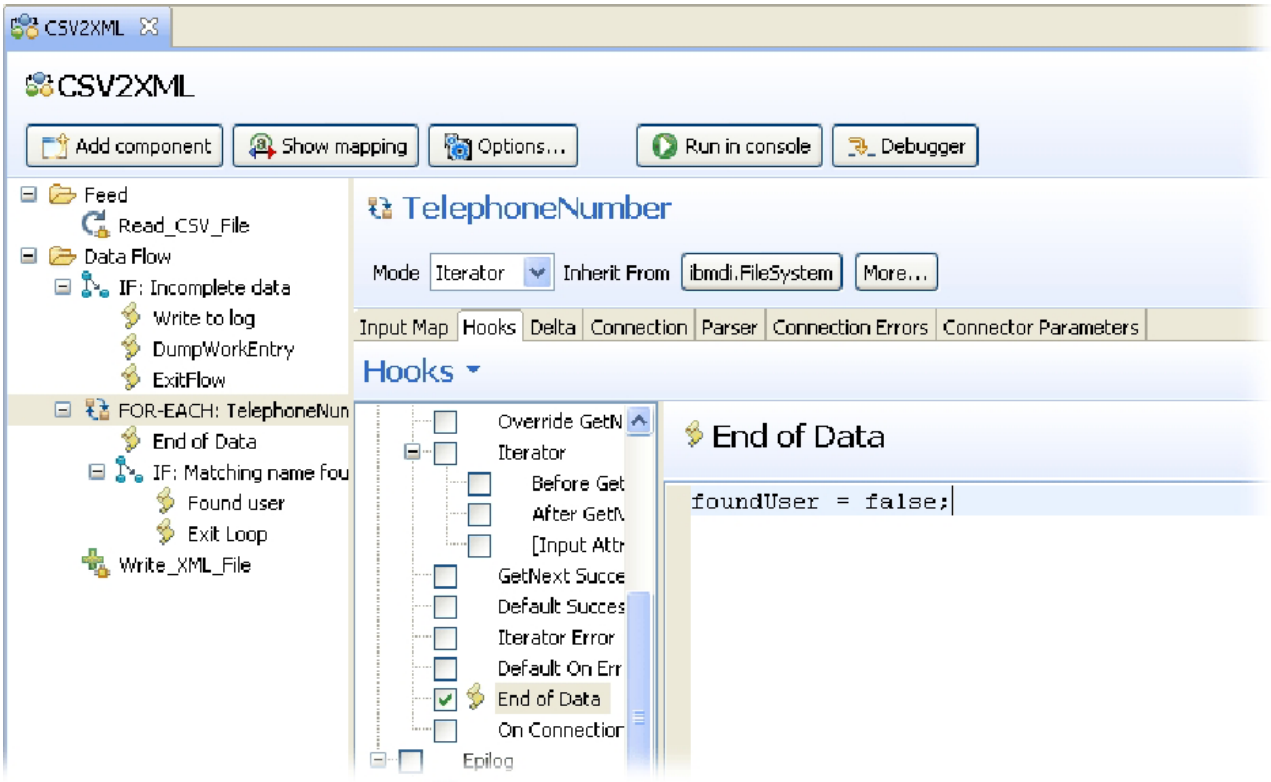

图 59. 编写数据末尾挂钩的脚本

选择名为“数据末尾”的挂钩，并输入以下脚本。

```
foundUser = false;
```

仅当连接器尝试在其已连接的源中的最后一个条目之后读取时，才会到达“数据末尾”挂钩。在这种情况下，找不到任何匹配项。

现在您的 AssemblyLine 应该具有以下组件:

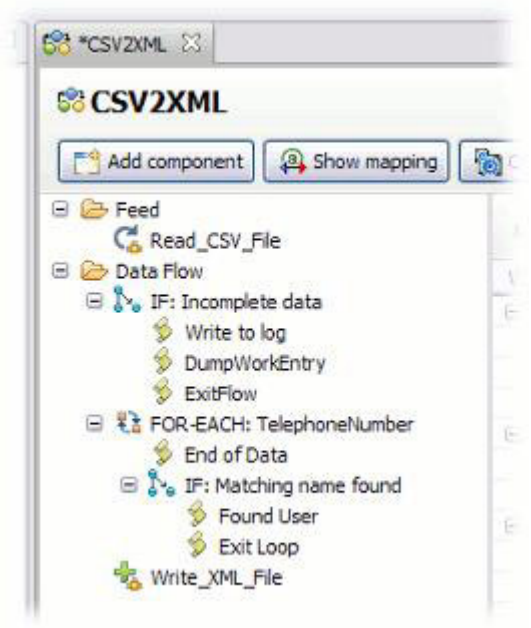

图 60. "AssemblyLine 数据流"部分中的组件列表

现在您应能够通过检查脚本变量来确定搜索是否成功。该操作就非常重要，因为只要找不到匹配项，那么您还必须设置"telephoneNo"属性的缺省值；否则，ConnectorLoop 将仍然读入最后一个值。

因此请紧接 ConnectorLoop 之后添加另一个 IF 分支，并将它命名为"NOT foundUser"。单击 **IF 分支**详细信息区域中的**脚本**按钮，并输入以下脚本来检查脚本变量的值:

```
! foundUser
```

感叹号对 foundUser 的值求反，因此如果 foundUser 在 ConnectorLoop 的"数据末尾"挂钩中设置为 *false*，那么该分支条件的求值结果将是 *true*。

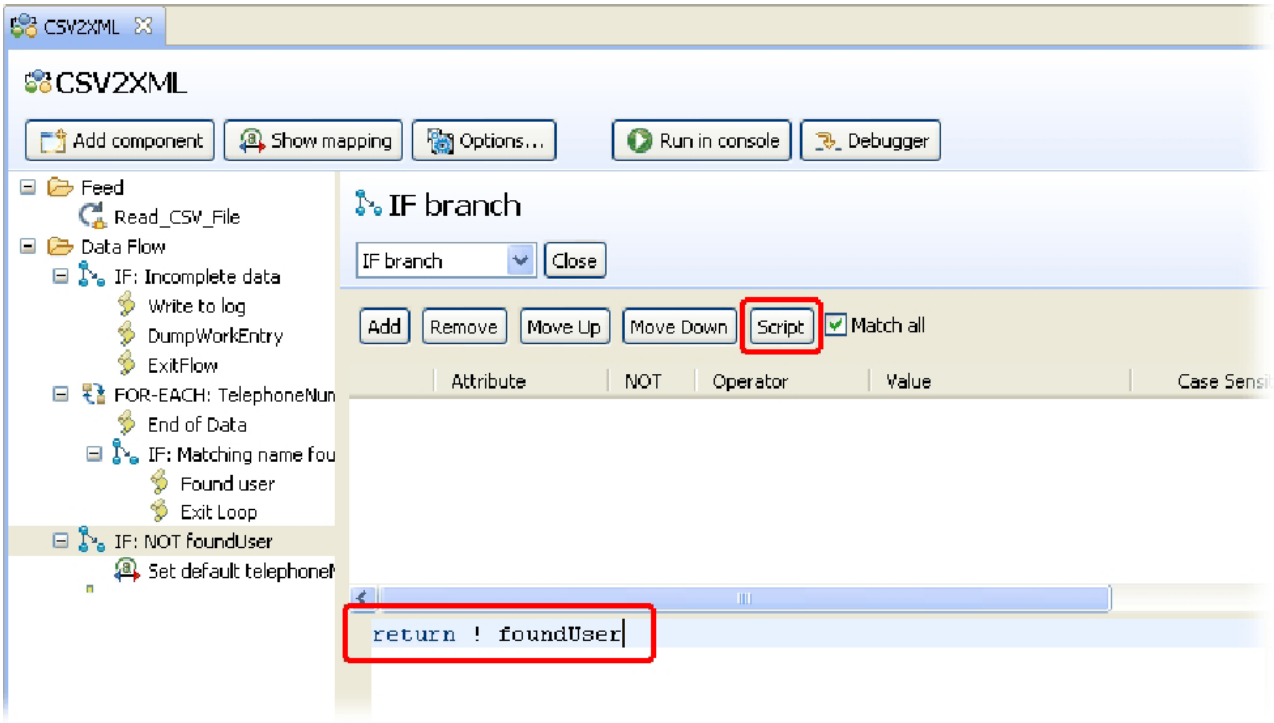

图 61. 为 IF 分支的条件编写脚本

将类型为“属性映射”的新组件插入其下。将此“属性映射”组件命名为“设置缺省 telephoneNo”以使其功能在 AssemblyLine 上下文中更加清晰。现在使用**添加属性**按钮以创建名为“telephoneNo”的单个属性，该名称与 ConnectorLoop 返回的名称相同。双击该属性以设置赋值脚本:

```
"N/A"
```

这表示对于从 CSV 输入中读取且在 PhoneNumbers.xml 中找不到的任何人，其“telephoneNo”的值将是“N/A”[26]。

最后，通过将此新的“telephoneNo”属性拖至“Write_XML_File”输出映射中，然后确保赋值为以下内容，可将该属性包含在该输出映射中:

```
work.telephoneNo
```

此时 AssemblyLine 看起来应类似于下图。

26. 如果宁愿这些用户根本不具有任何“telephoneNo”属性，只需使用不返回值的赋值即可。通过返回特殊值 NULL 可以执行此操作:

    ```
    null
    ```

    这样将使缺省 NULL 行为将从 Work 条目中除去该属性。因此，不会到达“Write_XML_File”连接器的输出映射，从而也不会出现在所产生的 XML 文档中。

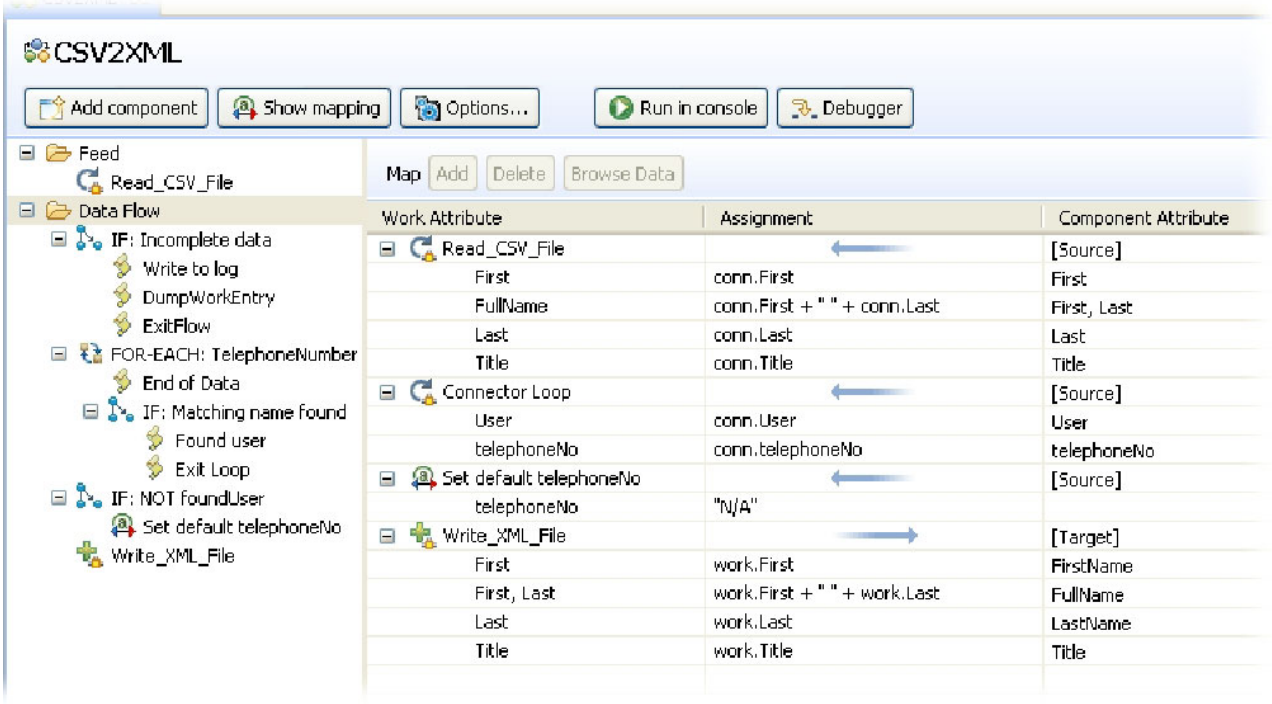

图 62. 使用 FOR-EACH 循环完成 AssemblyLine

现在再次运行 AL，并检查日志输出。“NOT foundUser”分支应该已两次为 *true*，且对于其他四个条目为 *false*。

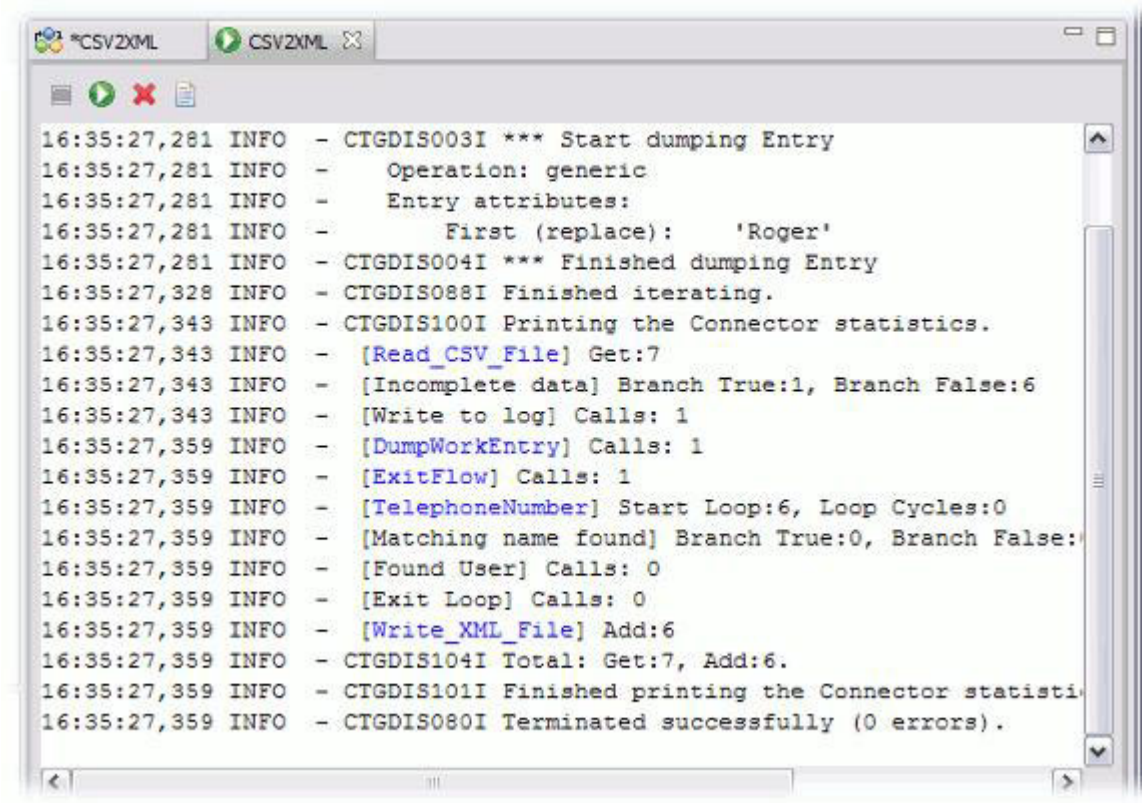

图 63. 包含 IF 分支统计信息的日志输出

请注意，某些组件名称在日志输出的 AL 统计信息中以蓝色突出显示。如果按住 Ctrl 键同时使用鼠标左键单击某个组件名称，将在 AssemblyLine 编辑器中打开选定组件。

这样，XML 输出看起来应该类似于下图:

图 64. 包含“telephoneNo”属性的 XML 输出

到目前为止，一切都很好。现在应该尝试使用*查找*方式来执行连接了。

使用 Lookup 方式

与其进行修改并面临正在运行的 AssemblyLine 变得混乱的可能性，不如制作副本，然后修改该副本。通过右键单击“CSV2XML.assemblyline”并选择**复制**来执行此操作。

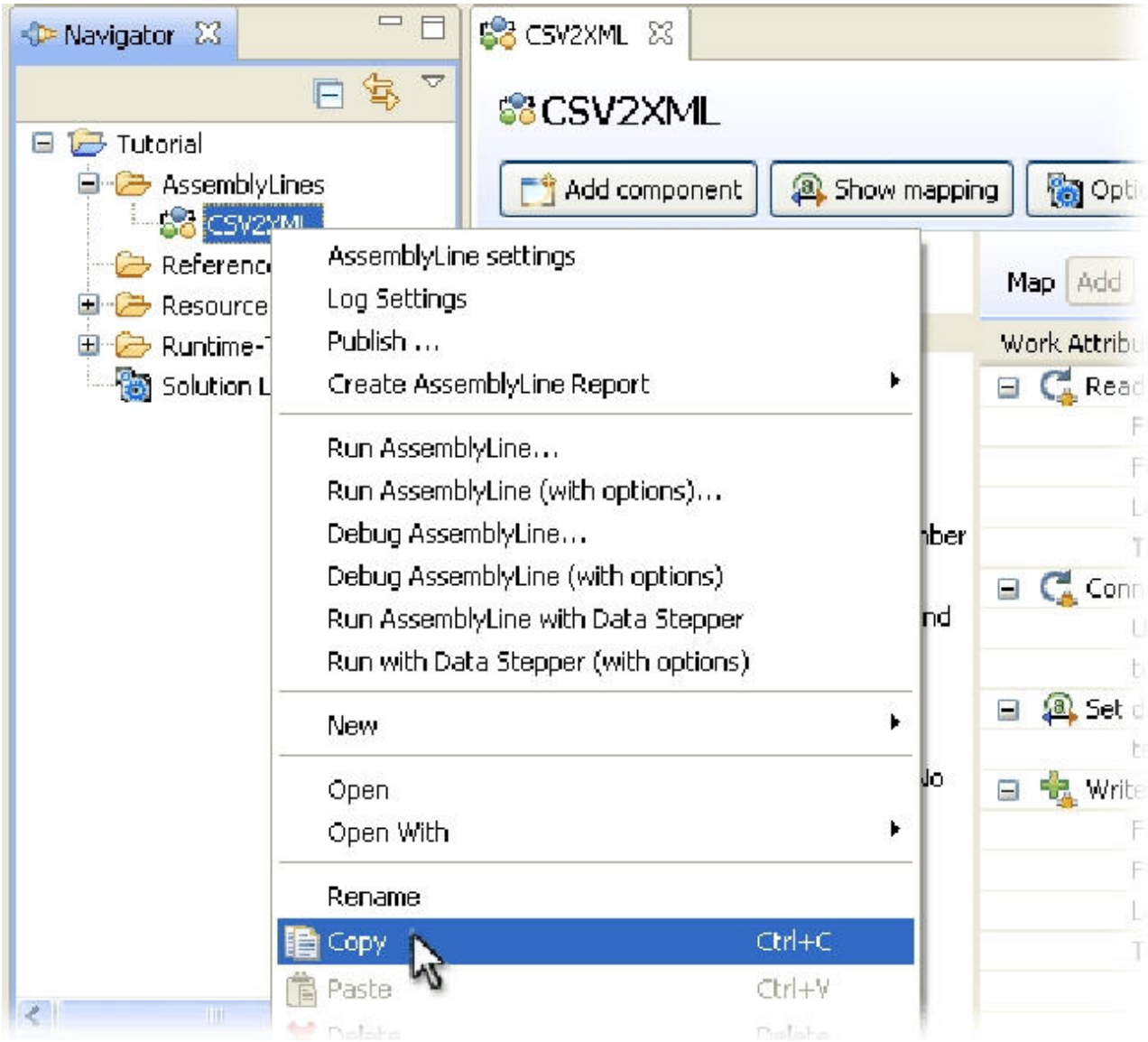

图 65. AssemblyLine 复制功能

现在右键单击“导航器”中的“AssemblyLines”文件夹，然后选择**粘贴**选项。将此新 AssemblyLine 命名为“CSV2XML_LookupMode”，然后双击它以打开 AL 编辑器。

现在您可以除去 ConnectorLoop 以及它下面的所有组件。选择它并按**删除**键。您将在其位置处拖入从另一个 AssemblyLine 复制的 JDBC 连接器。

但是，首先您必须构建此连接器要读取的数据库表；或者，必须运行预构建的 AssemblyLine 来构建该表。要执行此操作，请使用文件浏览器浏览到“Tutorials”文件夹，并查找名为 `CreatePhoneDB.assemblyline` 的文件。将其拖入“导航器”面板中“AssemblyLines”文件夹顶部的 CE 窗口中。

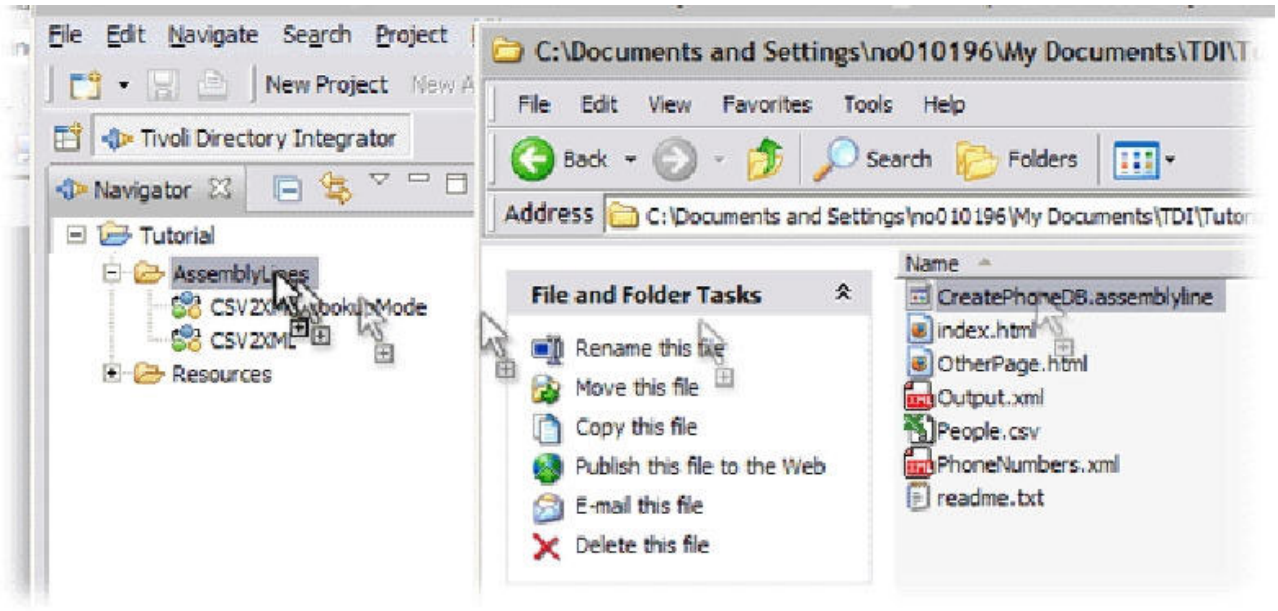

图 66. 将 AssemblyLine 复制到您的项目中

该 AssemblyLine 将导入到您的项目中。现在右键单击它，然后选择**运行 AssemblyLine...**

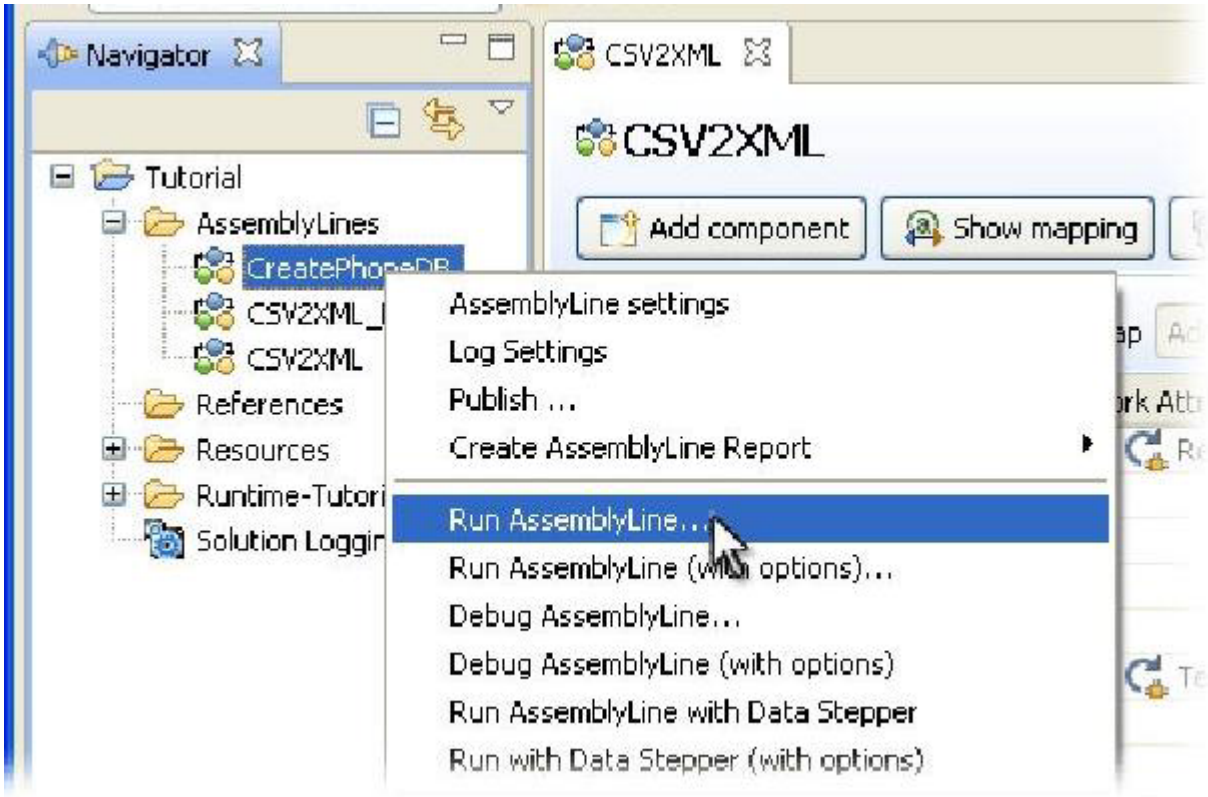

图 67. 运行 CreatePhoneDB AL

此 AssemblyLine 将首先在您的解决方案目录“TutorialDB”下创建一个 Derby[27] 数据库，然后创建“PhoneDB”表。然后它将在 `PhoneNumbers.xml` 中循环，并将此信息装入新表中[28]。

如果一切进展顺利，那么您的日志输出看起来应该类似于以下内容:

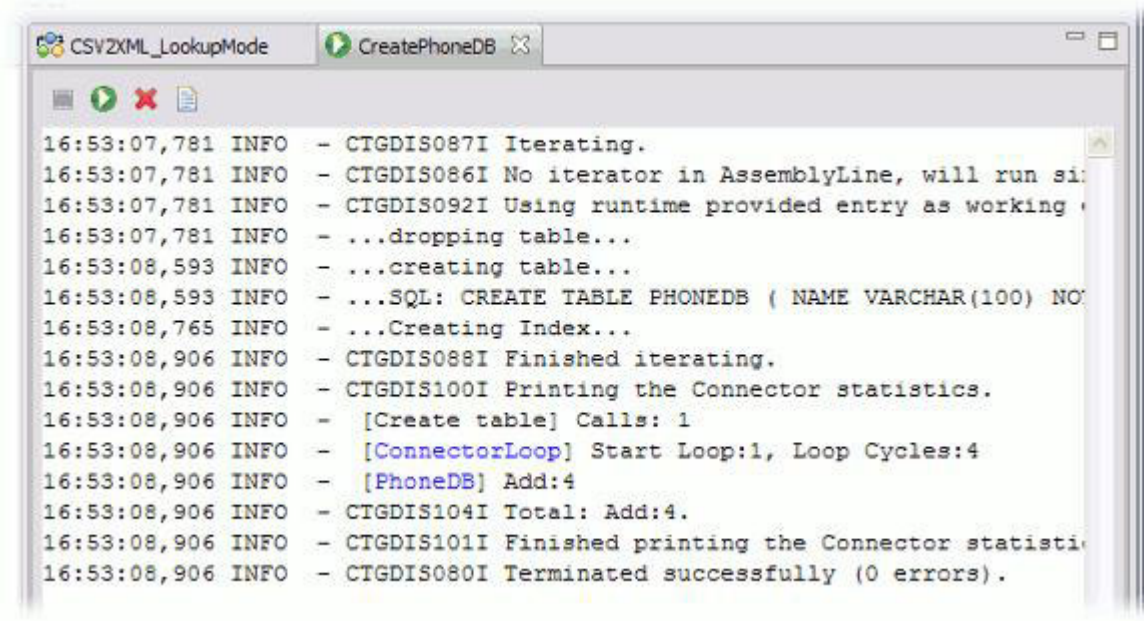

图 68. “CreatePhoneDB”AssemblyLine 的日志输出

“CreatePhoneDB”AssemblyLine 具有已配置并可以使用的 JDBC 连接器。您将把此组件复制到您的项目资源库（具体而言，为“Connectors”文件夹）中，然后在您的 AL 中进行复用。

打开“CreatePhoneDB”AssemblyLine，抓取连接器“PhoneDB”，并将其拖动到“导航器”树中“资源”下的“连接器”文件夹中。

27. Apache Derby 是开放式源代码关系数据库，与 IBM Security Directory Integrator 捆绑在一起。

28. 虽然本指南不会介绍该 AL，但是对于使用高级脚本编制技术开发在大多数连接器接口中可找到的特定于数据源的功能，这是一个很好的示例。您可随意查看和使用此 AssemblyLine。

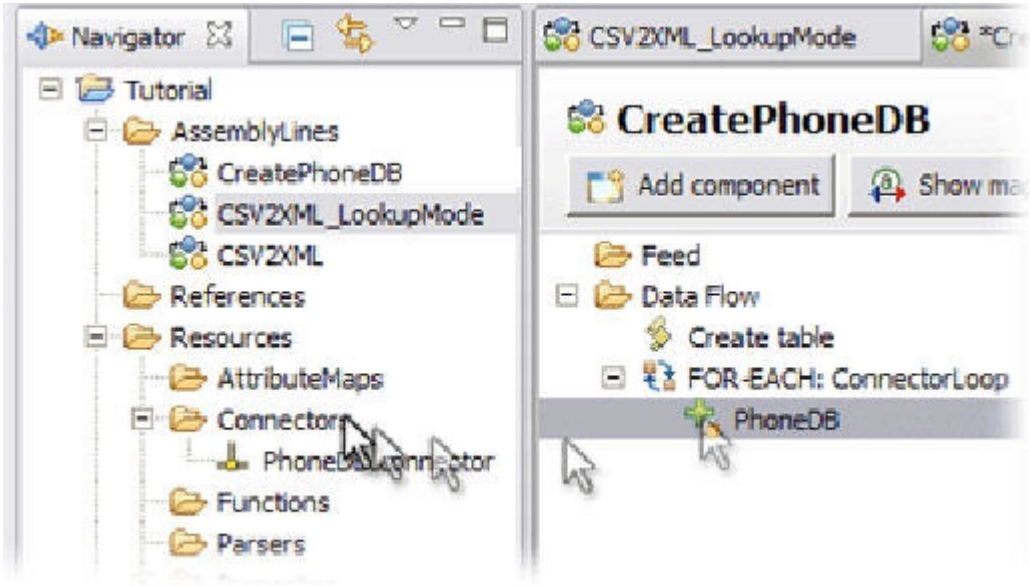

图 69. 将连接器拖动到“资源”中

关闭“CreatePhoneDB”以使您自己的 AssemblyLine 重新可见。然后将新的“PhoneDB.connector”资源拖动到 ConnectorLoop 以前所在的位置。

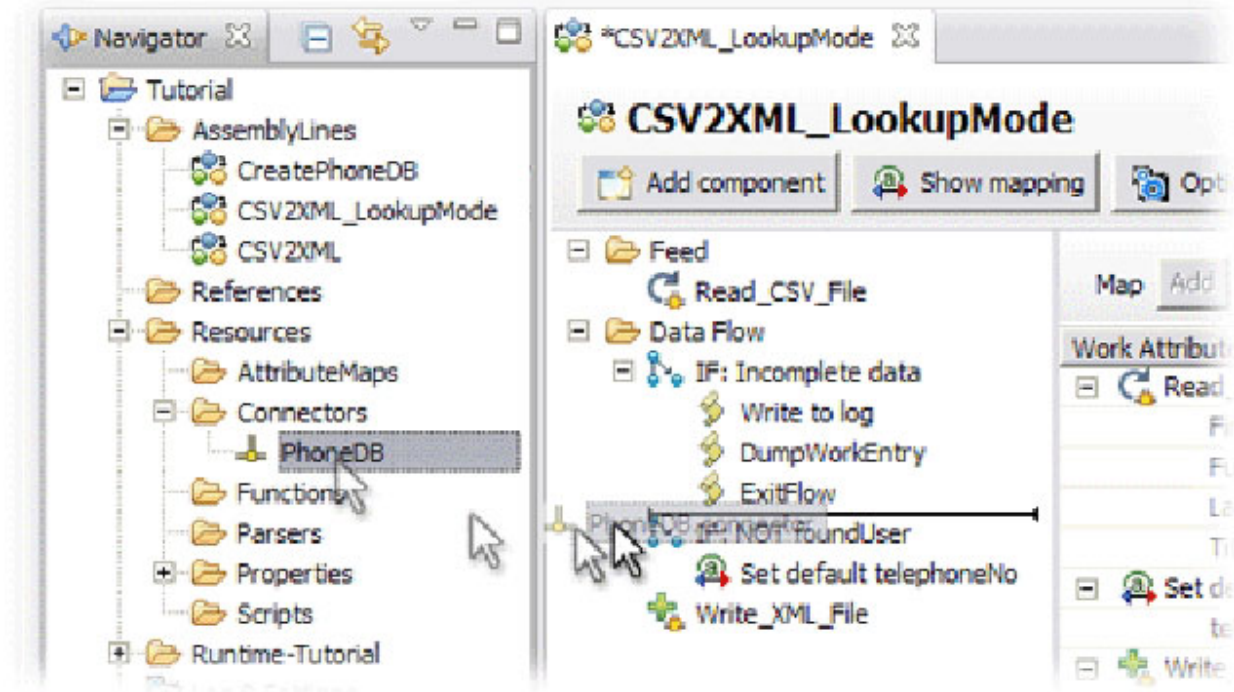

图 70. 将新资源拖动到您的 AssemblyLine 中

继承

您是否注意到此连接器在 AssemblyLine 中以蓝色显示？这是因为它此时继承自资源库。这表示它在运行时将从您拖动的连接器中动态检索配置设置。使用此继承功能可方便地在多个 AssemblyLine 之间复用资源，如已配置的组件和已编写脚本的逻辑。

可使用组件编辑器面板顶部的**继承自**按钮更改组件的祖代。组件选项卡（如“配置”、“变化量”、“输入/输出映射”和“挂钩”）还将提供继承选项，这允许您设置与该组件本身的祖代不同的祖代。

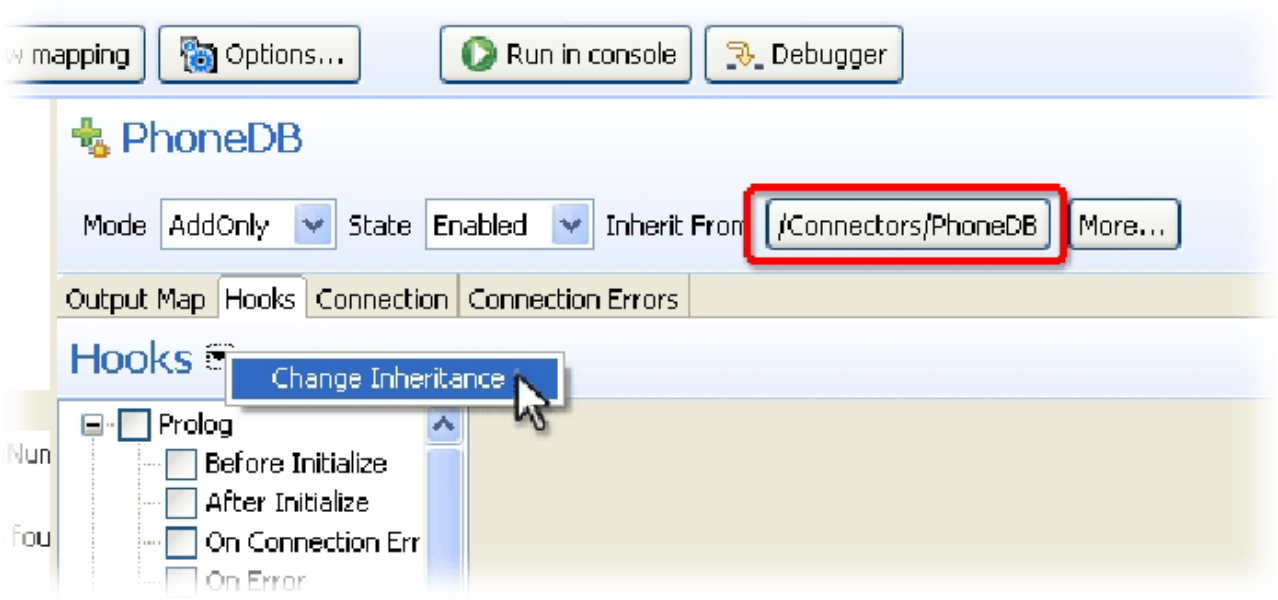

图 71. “设置挂钩的继承”选项卡

继承的值以蓝色形式显示，如果更改它，那么继承将中断。通过使用属性映射规则和挂钩的“上下文”菜单中的**还原为继承的值**选项恢复继承。

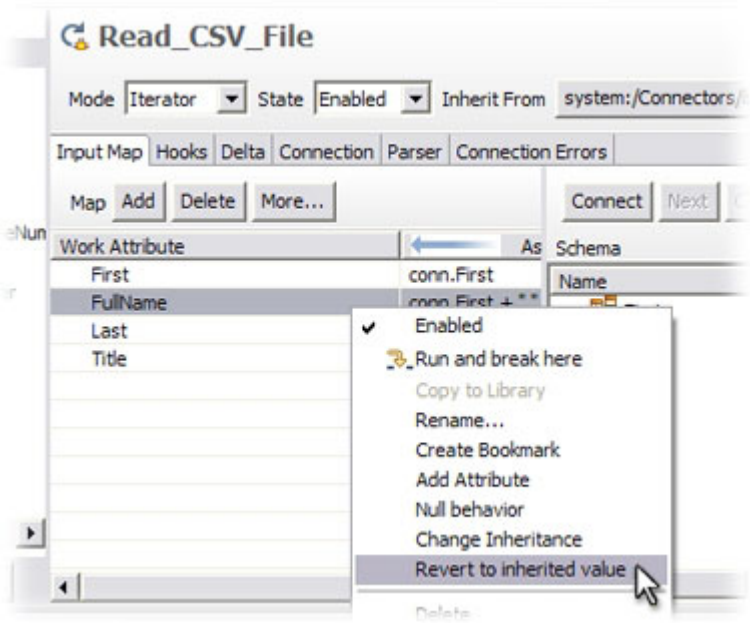

图 72. 恢复映射规则的继承

再次返回至练习，现在首先将新“PhoneDB”连接器的**方式**设置从**仅添加**更改为**查找**。

第二步，因为属性映射最初是与上一个方式相关联的*输出映射*，所以必须通过按“连接器模式”上方的**连接**和**下一步**按钮发现输入模式。第三步（即最后一步）是将“PHONE”属性从“模式”拖动至“输入映射”，以向您提供该值的简单映射。

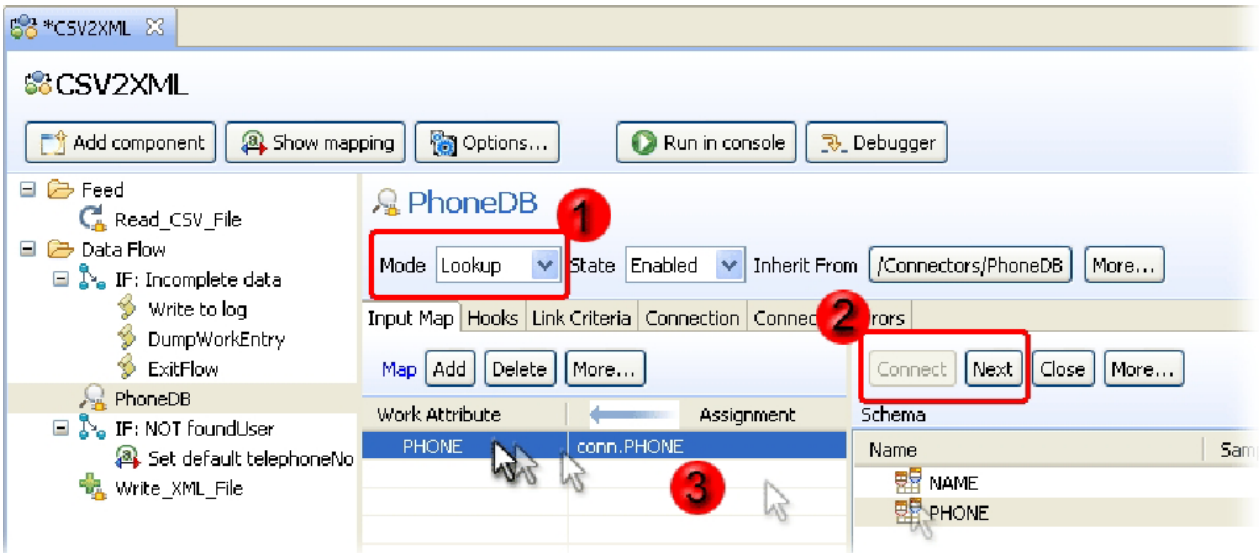

图 73. 更改方式，发现和映射属性

一旦使“输入映射”规则就位，将该规则从“PHONE”重命名为“telephoneNo”就很重要，这样才能使之符合“Write_XML_File”连接器的输出映射的赋值。

如果您感到疑惑，不必在“NAME”中进行映射，因为您已经具有该用户名。而在定义搜索规则时将使用该字段。

查询搜索规则 = 链接条件

选择*查询*方式时，连接器编辑器中将出现标有“链接条件”的新选项卡。链接条件用于定义搜索的匹配规则。

您可以将这些规则定义为简单的链接条件，方法是：使用下拉列表，根据需要添加新的链接条件，并按需要为条件设置**任意匹配**复选框。或者，可以选中**使用定制脚本构建条件**复选框，然后编写计算实际搜索规则的脚本片段，如 LDAP 搜索过滤器的以下示例：

```
"(cn=" + work.FullName + ")"
```

请注意，由于您必须编写已连接的系统所预期的实际语法，因此这会将您的解决方案与要搜索的数据源更紧密地联系起来。在此情况下，意味着需要创建 WHERE 子句（而不使用“WHERE”关键字本身）。

相反，连接器会将简单链接条件转换为本机搜索语法，因此您可以切换连接器界面，而不必重新设置链接条件。

简单链接条件看起来类似于条件。第一个下拉列表包含您发现的模式，第二个下拉列表显示 AL 中此时可用的 Work 条目属性。此外，就像条件一样，此处使用美元符号来表示应该在运行时替换命名属性的值以便创建搜索过滤器。

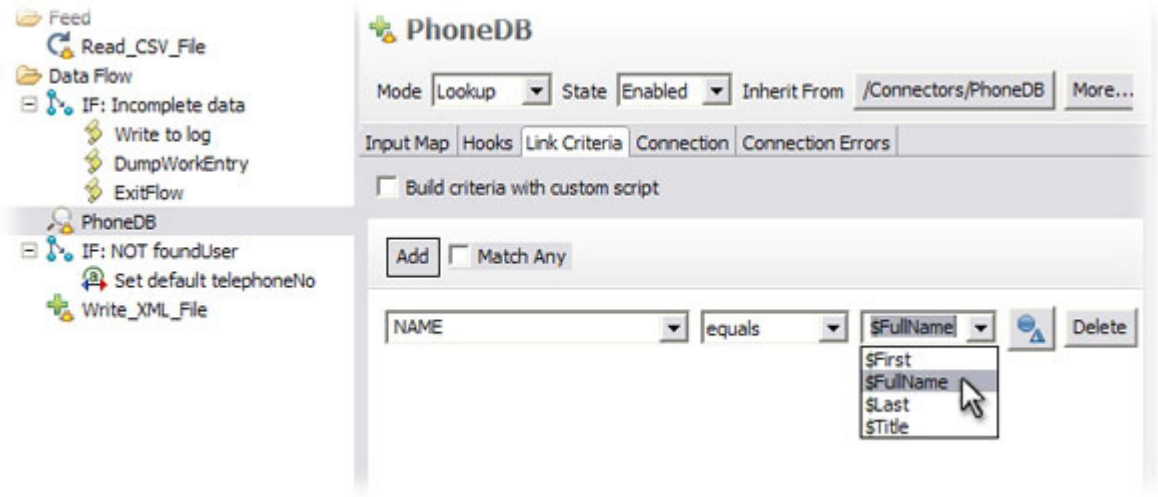

图 74. 简单链接条件

请记住保存您的工作，例如，按 CTRL-S[29]。您可能希望定期执行此操作以避免丢失任何工作。

现在可以重新**运行**您的 AssemblyLine 了。

29. 如果您已删除了 AssemblyLine 或资源并且希望撤销此操作，那么请在“导航器”面板中右键单击您的项目，然后选择**从本地历史记录恢复...**，将会向您显示可用来恢复的资产版本的列表。当然，您将必须先保存某些内容，然后该内容才能显示在您的本地历史记录中。

解释运行错误

请勿惊慌。您应该是遇到了日志输出中显示的堆栈转储所指示的错误。滚动到第一个堆栈转储的顶部。您将在此处看到关于问题发生的*地点*以及导致问题的*原因*的信息。

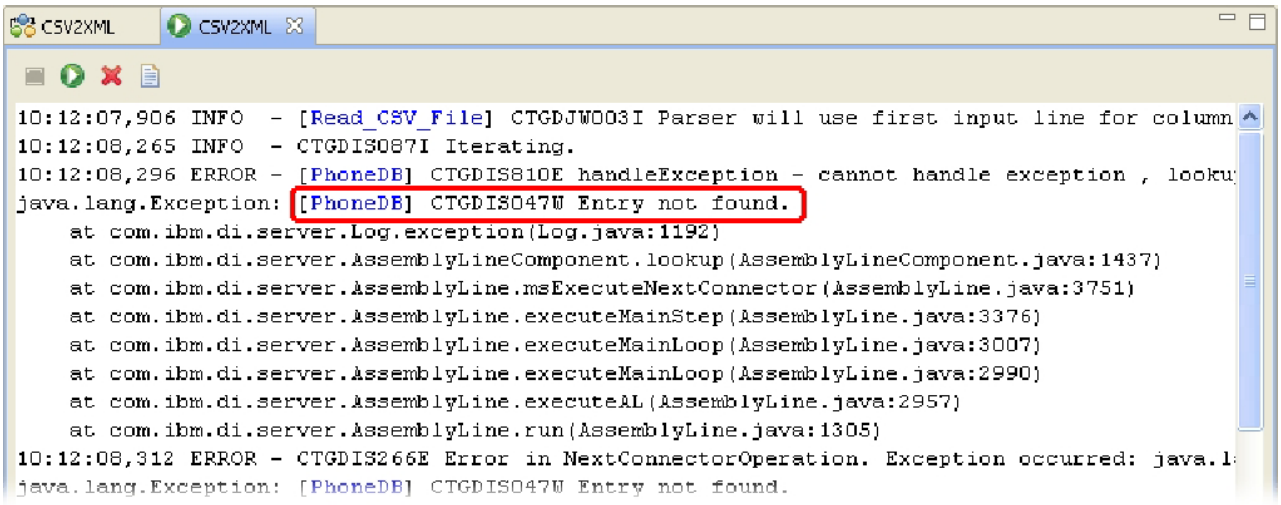

图 75. 日志输出中的错误消息

组件名称位于方括号中（“PhoneDB”），后跟指示找不到条目（换句话说，查找失败）的错误描述。

当连接器以*查找*方式配置时，系统在执行搜索时应该会找到一个且只有一个匹配的记录。如果未找到任何记录或者多个记录与“链接条件”匹配，那么必须至少启用*挂钩*以阻止 AL 停止，并以特殊的挂钩结束。在作为 *引用* 一部分的“数据流”图中，此行为清晰可见。下面是来自详述*查找*方式的页面的摘录:

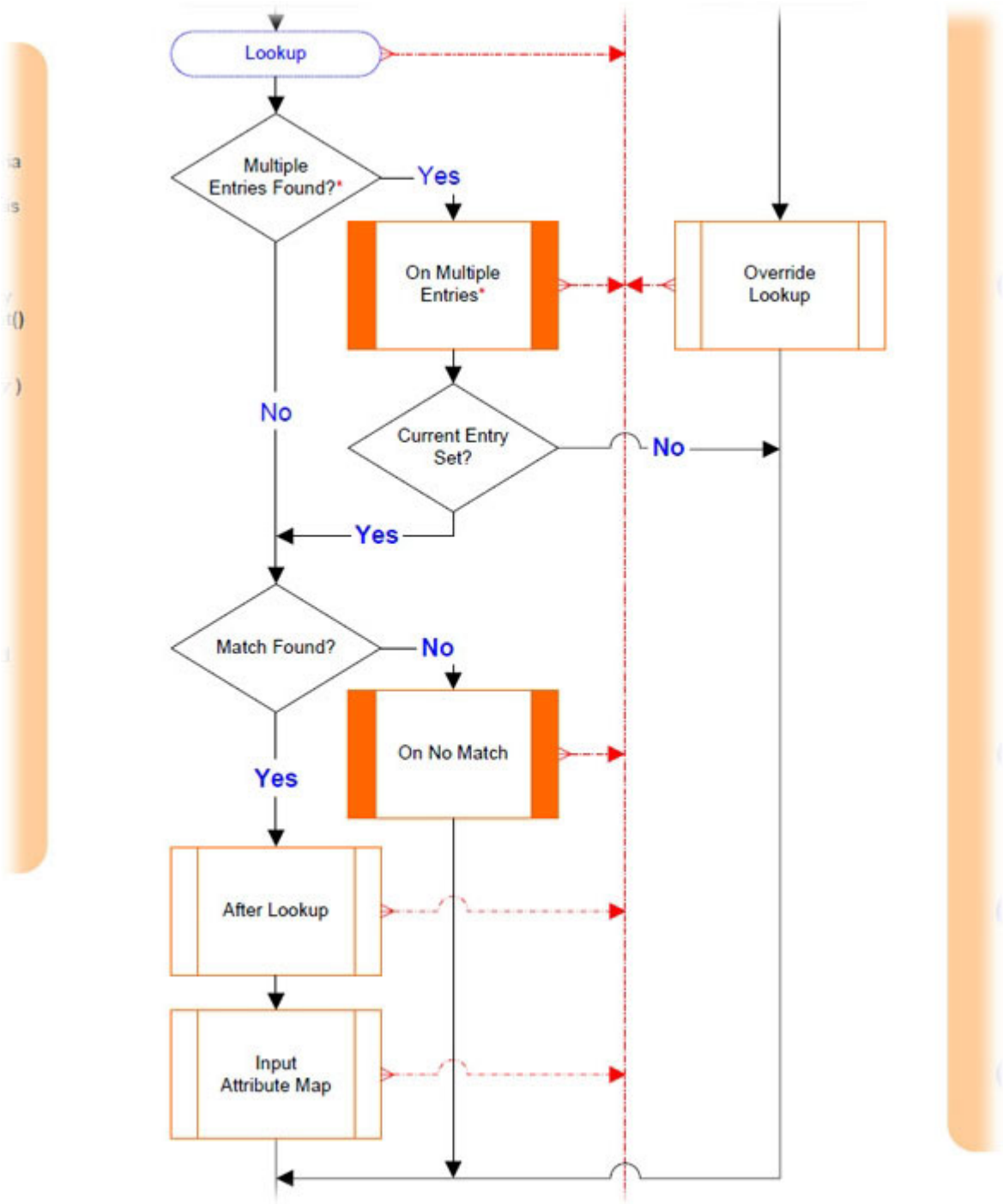

图 76. Lookup 方式的部分流程图

您可以利用该行为来设置 `foundUser` 标志变量。右键单击“PhoneDB”连接器并选择**挂钩...**以打开挂钩编辑器。选择“不匹配时”挂钩并输入脚本代码以将 `foundUser` 设置为 *false*。

```
foundUser = false;
```

现在选择“查找后”并输入以下补充挂钩脚本[30]:

30. 正如您在流程图中所看到的，仅当搜索结果为单个匹配时才会执行“查找后”挂钩。

```
foundUser = true;
```

您的 AssemblyLine 现在应该与此屏幕快照中的 AssemblyLine 相似:

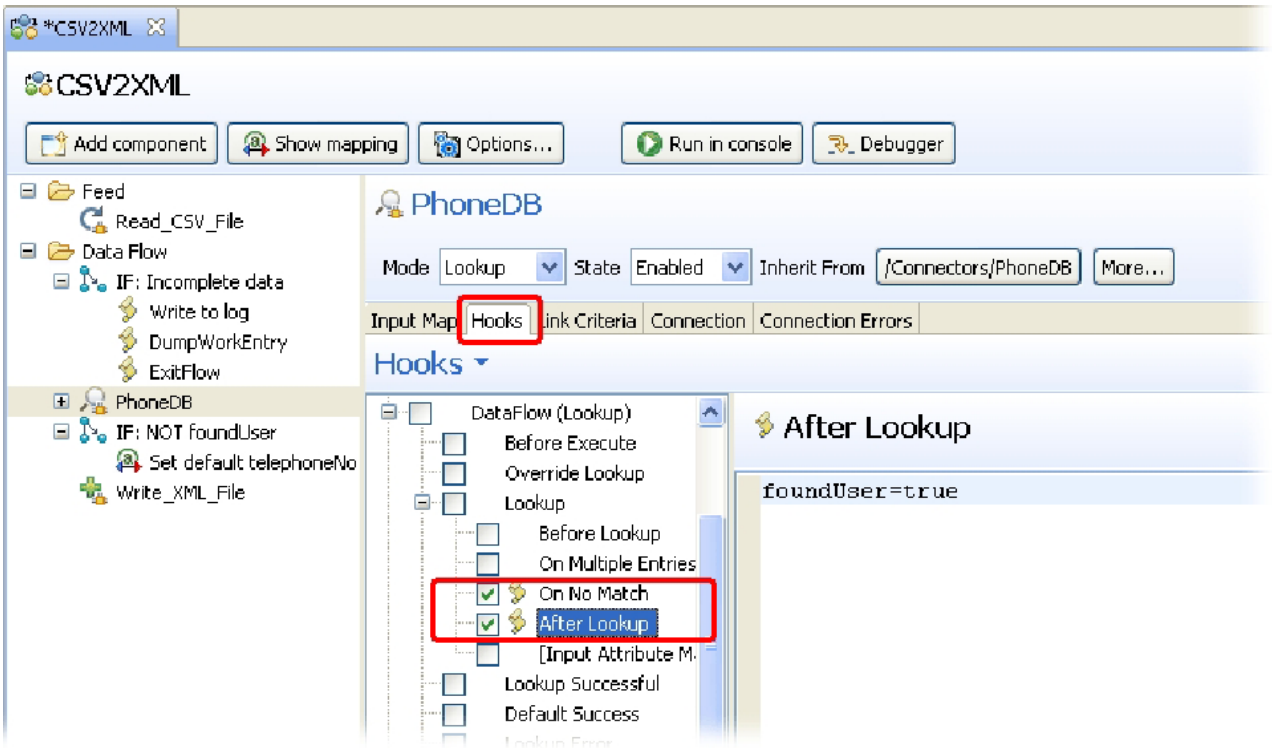

图 77. 第一个教程练习已完成

现在可以再次**运行**该 AssemblyLine ， 这次它应该会无错完成。Output.xml 文件应该与从基于循环的 AL 获取的结果一致。

祝贺您！您刚刚完成了您的第一个 IBM Security Directory Integrator 教程练习。现在我们开始讨论实时事件如何触发 AssemblyLine 。

第 3 章 事件驱动的集成

到目前为止，您一直将 AssemblyLine 作为批处理进程运行，从而每次想要数据流动时就手动启动这些进程。还可以如下通过调用 IBM Security Directory Integrator 服务器从命令行来运行 AssemblyLine[31]:

```
ibmdisrv -c examples/Tutorial/Tutorial1.xml -r CSVtoXML
```

使用这种方法，可以很容易地使用调度工具（例如，*crontab*）调度其操作，或很容易地从外部应用程序中启动 AssemblyLine 。

IBM Security Directory Integrator 提供了大量的功能，用于使 AssemblyLine 知悉事件、允许解决方案进行处理并对各种实时触发器作出响应。

这些触发器的示例有:

- 通过 IP 端口的协议请求，例如 REST 调用、SNMP 和 Web service;
- 在队列中出现的新消息;
- 到达收件箱的电子邮件;
- 对数据的更改，例如文件、数据库、目录和 Notes® 数据库中数据的更改;
- 基于调度或计时器的 AssemblyLine 操作。

这并不是一个完整列表，您将在其他 IBM Security Directory Integrator 文献、社区 Web 站点和新闻组中找到启发和指导信息。

可以用几种方法来处理解决方案中的这些事件:

迭代器方式的连接器

某些连接器允许您为 Iterator 方式配置超时参数。FileSystem 连接器就是一个示例，该连接器可以设置为通读文件至结尾，然后等待新信息出现，这就是所谓的“尾部读取”。

其他连接器，如用于 RDBMS 更改检测和 LDAP 更改日志的连接器都是以类似方式工作。这些连接器允许您构建持续运行等待要在连接的系统中出现的新更改的 AssemblyLine 。

还有一个计时器连接器，它以“迭代器”方式运行，可以配置为根据调度参数按设定的时间间隔驱动您的 AssemblyLine。您很快就要测试该连接器。

IBM Security Directory Integrator 包含作为标准安装一部分的 Web 管理控制台。这一称为管理和监视控制台 (AMC) 的基于浏览器的应用程序可用来监视 AssemblyLine 的运行状况、将配置热装入至正在运行的服务器、启动和停止 AL 以及配置故障/响应行为以保持集成解决方案高度可用。它还可用于设置 AssemblyLine 应该何时运行的调度。但是，该 Web 管理工具超出了本指南的讨论范围。

注: 不推荐使用 AMC 功能部件，也将从 IBM Security Directory Integrator 的未来版本中移除该部件。

31. 不带命令行参数调用 `ibmdisrv` 时，IBM Security Directory Integrator 服务器将提供用法消息

服务器方式的连接器

有一些专用的连接器（如 HTTP Server 连接器和 LDAP Server 连接器）可用来构建解决方案以处理来自外部客户机的传入请求、执行所请求的操作并通过适当的响应进行回复。您将在下一个练习中使用 HTTP Server 连接器。

通知和属性

就像还有用于发送 IBM Security Directory Integrator 通知事件（甚至是在不同平台上运行的不同 IBM Security Directory Integrator 服务器之间发送事件）的组件（和脚本调用）一样，IBM Security Directory Integrator 具有可预订这些事件的组件。

本指南将引导您进一步了解 AL 调度和“连接器服务器”方式。

调度 AssemblyLine

使用“配置编辑器”中的用户界面，您可以创建“调度程序”以在指定的时间运行 AssemblyLine。例如，您可以创建一个“调度程序”以在每天的 3:05 AM、每个星期六的 7 AM 或更复杂的调度来运行 AssemblyLine。

要创建“调度程序”，在“配置编辑器”中，单击**文件 > 新建 > 调度程序**。或者，右键单击 AssemblyLine 并选择**创建调度**。

IBM Security Directory Integrator 服务器装入配置文件时，包含在配置文件中的启用的“调度程序”将自动启动。用于装入配置文件的命令为 `ibmdisrv -c myconfig.xml -d`，其中 `myconfig.xml` 是导出的配置文件。停止配置实例将停止所有相关的“调度程序”。

图 78. IBM Security Directory Integrator 调度程序

引用 的 *IBM Security Directory Integrator 调度程序*部分对 IBM Security Directory Integrator 调度程序进行了详细的描述。

服务请求 AssemblyLine

在最后的这个练习中，您将构建 Web 服务器 AssemblyLine 以提供用于启动“CSV2XML_LookupMode”AL 的极简单的用户接口；换句话说就是用于启动数据传输的 HTTP 服务。

首先创建新的 AL 并将它命名为“TINA_WebServer”[32]。然后插入新组件，选择“HTTP Server 连接器”，然后按**完成**结束向导。现在打开它的**输入映射**选项卡。

32. 此处 TINA 表示“TINA 不是 Apache”:-)

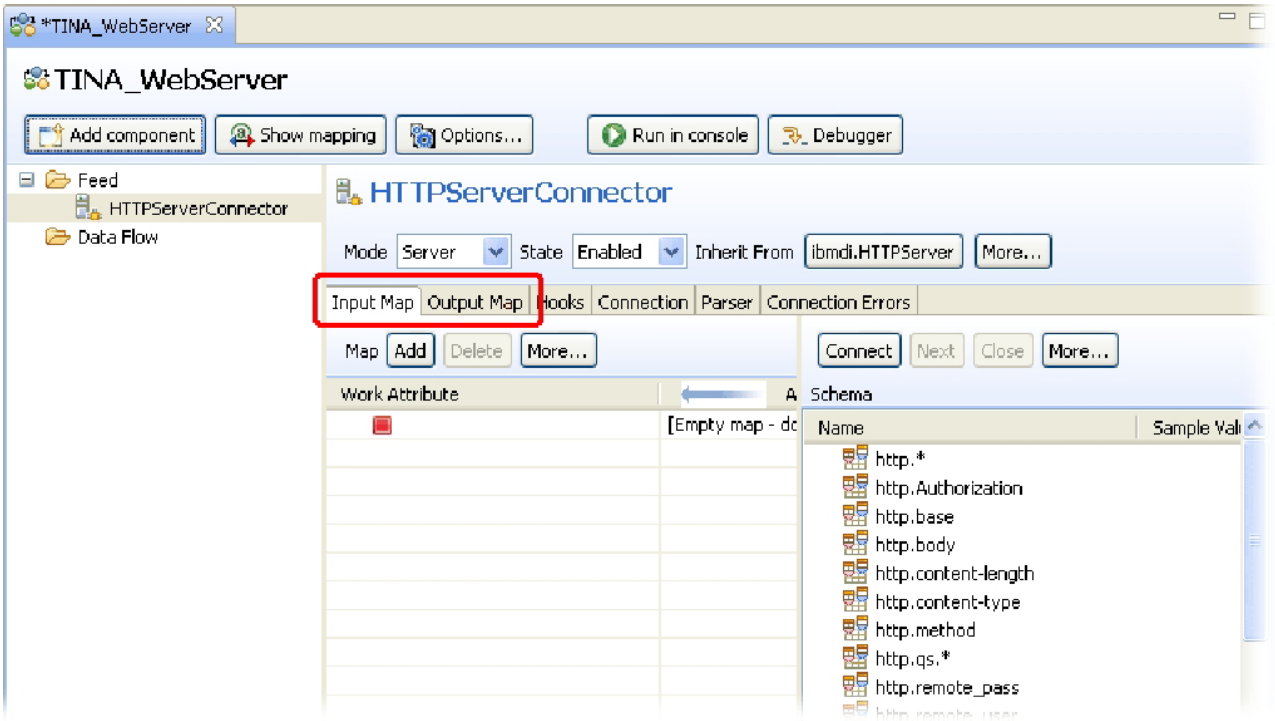

图 79. "HTTP Server 连接器属性映射"面板

服务器连接器是对功能组件的补充。但 FC 作出服务请求，服务器连接器提供服务并支持服务。因此，服务器连接器的输入映射用于接收来自作出请求的客户机的属性，输出映射则用于提供回复。您还可以在"属性映射"屏幕的右边看到已存在输入模式。对于输出模式同样如此。服务器连接器提供此信息以帮助您进行映射。但请注意，其中一些模式属性包含通配符，如"http.qs.*"。这是设计时信息，告诉您需要名称以"http.qs"开头的任意数量的传入属性。[33]

最后，所有服务器连接器的**方式**下拉列表都提供"服务器"和"迭代器"方式。方式总数为三个，此处未显示第三个方式（响应）。服务器连接器在其操作的不同阶段会在这些方式之间进行切换:

1. 服务器连接器首先以*服务器*方式启动，连接到某个资源（如 IP 端口），并等待传入的客户机连接;
2. 一旦建立连接，连接器就切换到*迭代器*方式以基于输入映射检索客户机数据，并将数据传递给"数据流"组件;
3. 最后，*数据流*组件完成执行时，连接器转到"响应"方式，使用输出映射对客户机作出回复。

您不必担心此切换操作，因为系统会自动处理。但是，如果您编写任何挂钩脚本，那么您将看到存在三组挂钩: *服务器*、*迭代器*和*响应*。

添加输入映射属性项，继续教程练习。

33. 此特定属性集（http.qs.*）将包含客户机传递到 HTTP 调用中的所有查询字符串参数。

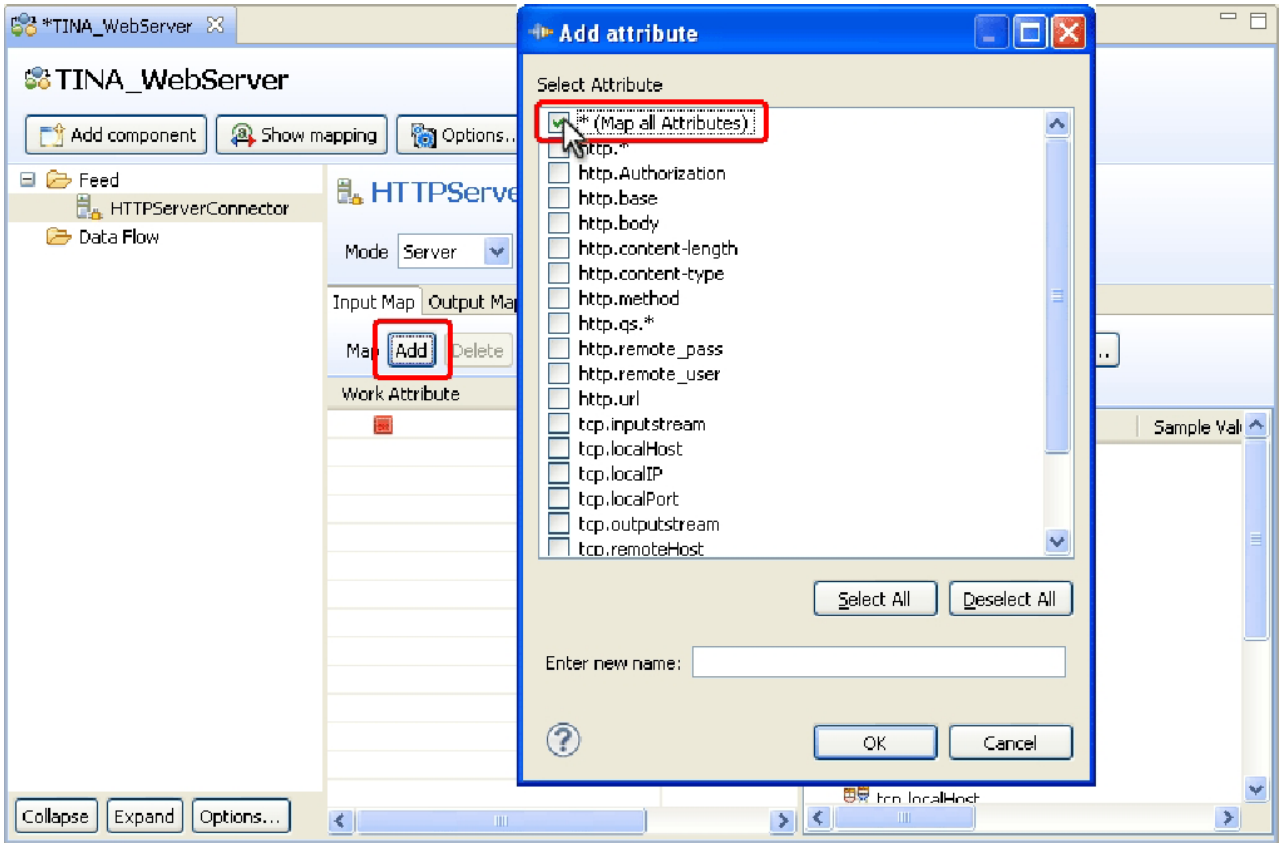

图 80. 添加输入映射属性项

只需选择呈现的对话框中最上方的选项。请注意，您可能已在“输入新名称”字段中输入星号字符 (*)。这是特殊通配符映射规则，用于告知 IBM Security Directory Integrator 对连接器所读取的所有属性进行映射。

您的输入映射看起来将类似于以下内容:

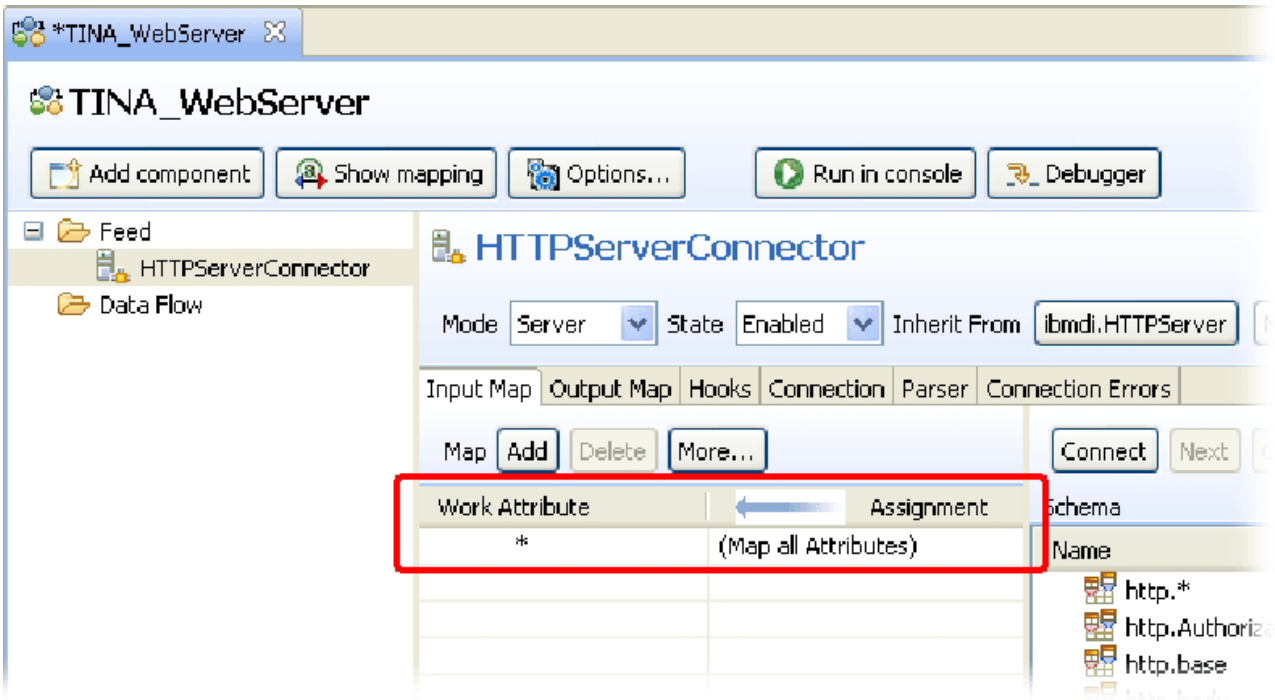

图 81. 通配符映射项

同时将通配符映射项添加到输出映射，可确保在响应消息的 Work 条目中设置的任何属性都会映射回客户机。

要测试此组件，请将“转储 Work 条目”脚本组件放入“流”部分，然后**运行** AssemblyLine。日志输出应该显示消息，表明您的 HTTP Server 连接器正在端口 80 上侦听 HTTP 连接[34]。这意味着您的 AL 正在等待客户机连接（这是您现在正通过打开浏览器并浏览到以下 URL 所执行的操作）：

```
http://localhost:80
```

现在查看您的日志输出。在此处您将看到已作为属性返回的大量 TCP 和 HTTP 头属性。

34. 如果由于某种原因，此端口已在使用，那么只需打开 HTTP Server 连接器的“配置”面板，然后选择其他端口。

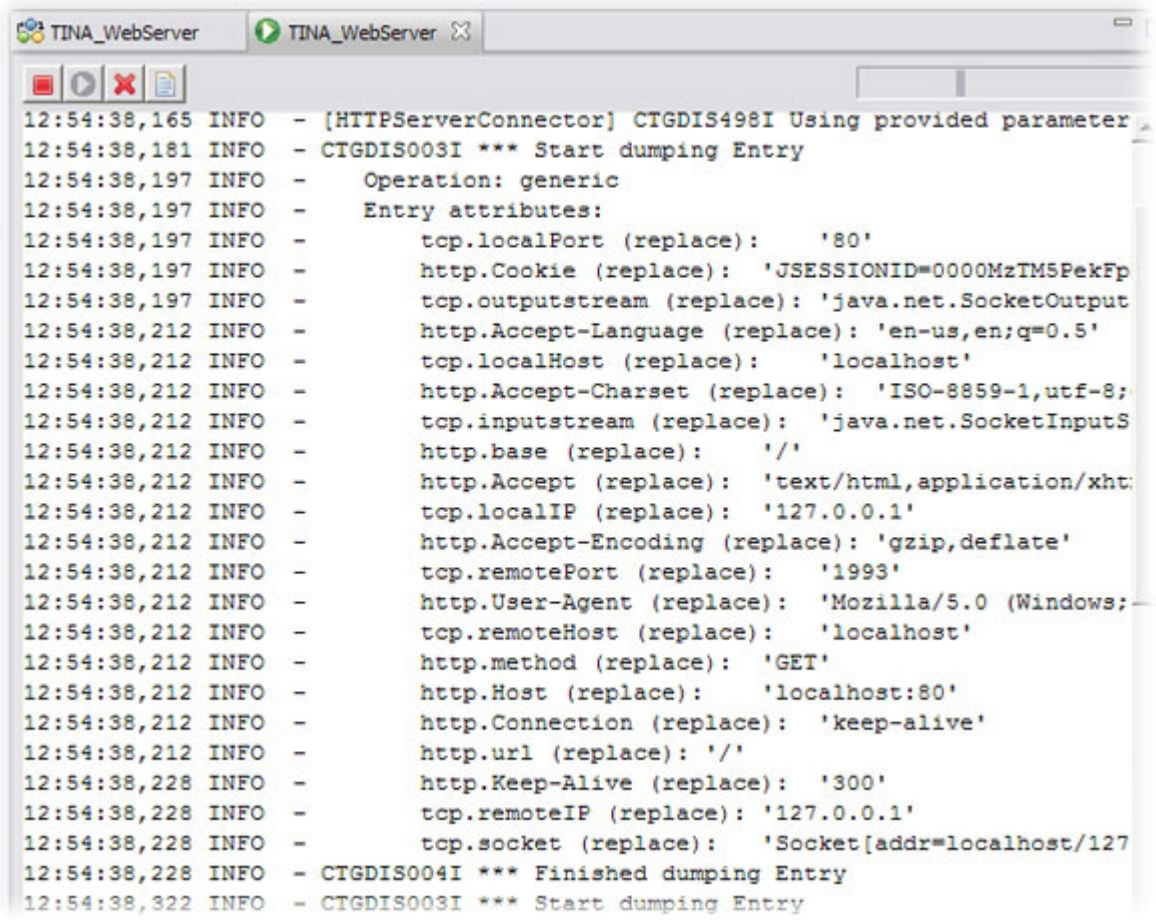

图 82. 作为属性返回的 TCP 和 HTTP 头属性

此练习中您将感兴趣的唯一属性是“http.base”，它包含在主机和套接字后出现的那部分 URL。具体而言，您将要检查它是否包含文本“RunAL”。

要检查此文本，请将 IF 分支添加到 AL 的“数据流”部分。将它命名为“RunAL detected”，并将条件设置为: http.base *contains* 'RunAL'。

如果此分支条件的求值结果为 *true*，那么您将希望启动“CSV2XML_LookupMode”AL。要执行此操作，您将复用作为“调度程序”的 AL 中的 AssemblyLine 功能组件，方法是: 首先将其拖动到“导航器”面板的**资源 > 功能**中，然后返回到此 AssemblyLine 中，将其拖动到新 IF 分支的顶部。

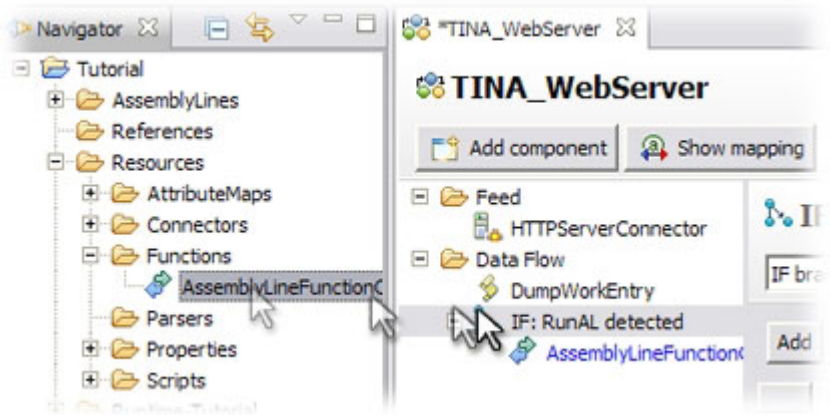

图 83. 拖入 AssemblyLine 功能组件（AL FC）

现在重新运行您的 AssemblyLine，并在浏览器的地址字段中输入以下文本:

```
http://localhost/RunAL
```

现在您将在日志输出中看到 Work 条目转储，后跟被调 AssemblyLine 的统计信息。

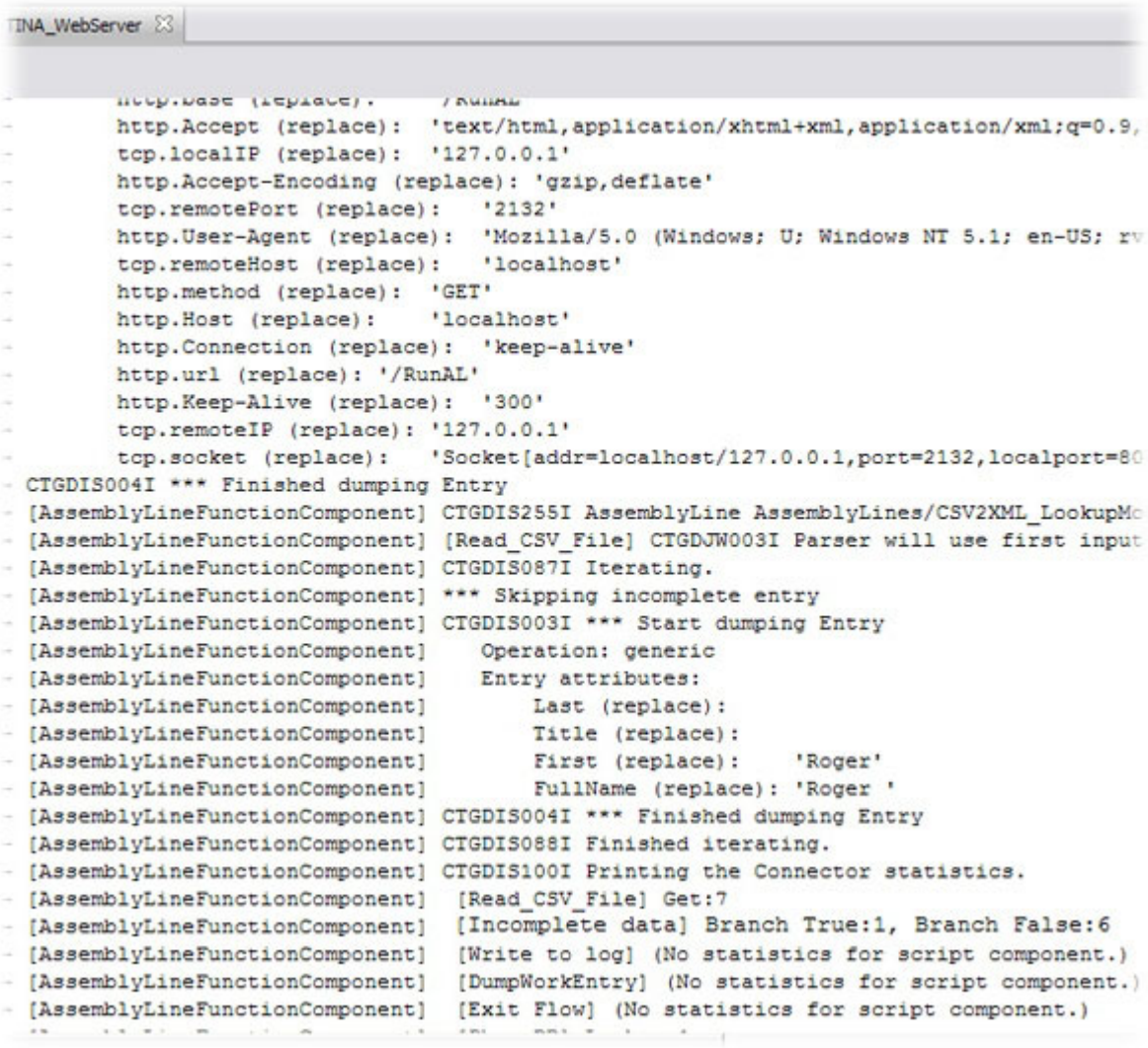

图 84. 后跟 AL 统计信息的 Work 条目转储

您的服务已在工作，但是练习尚未结束。首先，通过使您的 Web 服务器 AssemblyLine 返回一些 HTML 页面，其美观性将只是略有改善，但易用性则大大提升。这通常需要编写大量脚本。幸运的是为您提供了一些教程文件，因此您只需执行拖放操作即可。

开始前，请禁用“转储 Work 条目”脚本以最小化日志输出。然后紧接在“IF RunAL detected”之后添加 ELSE 分支。将其命名为“Return web page”。现在使用文件浏览器在 Tutorial 目录中查找名为“`Return web page.script`”的脚本文件，并将其拖动到**资源 > 脚本**中。您可以将该脚本从该处拖到您的 AL 中，放到 ELSE 分支的顶部。现在您的 AL 看起来应该类似于以下内容:

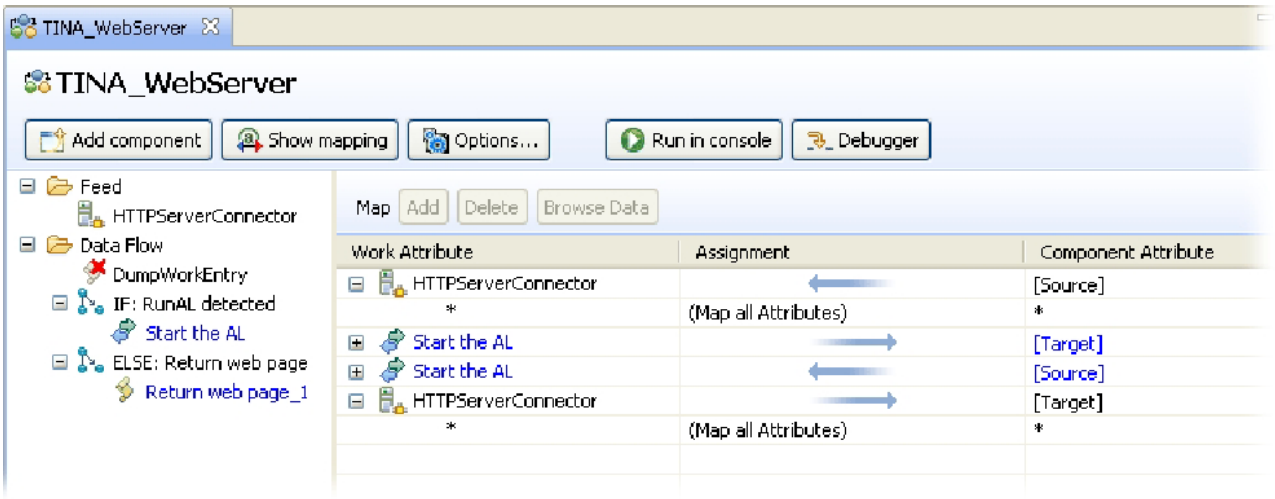

图 85. 已完成的 TINA_WebServer AssemblyLine

再次运行 AssemblyLine，当您访问 http://localhost 时，应该会看到以下 Web 页面:

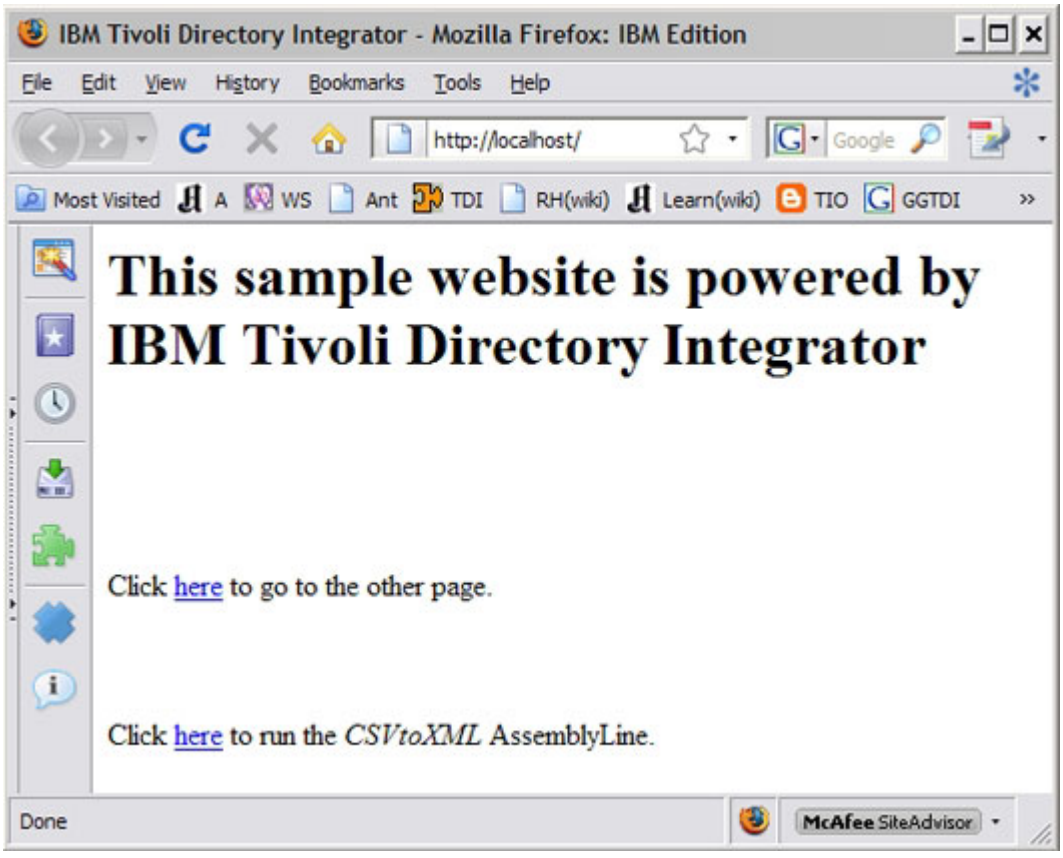

图 86. 解决方案的简单 Web 界面

顶端的链接将导航到 `OtherPage.html`，而底部的链接应该发送 http.base 中的所需“RunAL”文本以启动您的 AssemblyLine。

这样就完成了本指南中的手动操作部分。

第 4 章 加强集成解决方案

正如您所看到的，创建 AssemblyLine 可以是相当迅速的一件事。但是，AL 运行完成并不表示您的解决方案已准备进入“黄金时段”。即使是简单的集成任务也应保证预想和结果分析都为最少。

要反思的一些相关问题是:

- 是否所有的源数据都按预期进行处理？您如何确认这一点？
- 是否在数据内容和/或质量中检测到异常情况？是否已处理这些异常情况？
- 处理是否会影响其他数据集、系统或 AL？如何影响？
- 集成是否带来审计负担？
- 谁将部署您的解决方案？谁将使用该解决方案，谁将管理该解决方案以及如何进行管理？

对于长时间运行的 AssemblyLine （如用于同步和支持服务的 AssemblyLine ），可以附加其他的注意事项，例如可用性和容错性、性能、可伸缩性和安全性。

最后这部分的目标在于使您意识到这些问题，并提供了可用于解决这些问题的大量 IBM Security Directory Integrator 功能和技术。

注: 如果您对使用 IBM Security Directory Integrator 是新手，请不要担心这部分看起来复杂而难以理解。随着您的经验逐步丰富和对系统日益熟悉，请回头重新阅读这些部分。

易读性、复用和可配置性

所有开发工作都需要进行故障诊断、维护和扩展。IBM Security Directory Integrator 解决方案也不例外。

您可以遵循几条基本准则以使这些操作更便利。

1. 编写 AssemblyLine 时请记住，其他人必须能够理解、使用和维护这些 AssemblyLine。这表示要使 AL 尽可能简短并清楚明了地命名组件。对于不是程序员的人来说，读取和调试 AL 流中通过分支和循环实现的逻辑比在挂钩中“隐藏”的或打包到脚本组件中的脚本更简单。
2. *简短 AL* 规则的一个推论是应同时使脚本片段保持简短。将这些脚本片段分成较小的单元，甚至将它们放入单独的脚本组件中，而不是编写单块的代码块，从而提高易读性和可调试性。然后可以禁用 SC 以跳过代码。

   提高易读性和避免代码重复的另一种方法是使用脚本组件继承（从“导航器”树形视图中的“脚本”文件夹）并定义执行一般任务的函数。AssemblyLine 在其自身的脚本引擎的上下文中执行，因此在一个地方声明的所有变量和函数可以全局使用。通常在 AL Prolog 挂钩中或在已选择为“附加 Prolog”[35] 的脚本中定义这些变量和函数。

35. 附加 Prolog 将在调用任何 AssemblyLine 自身的 Prolog 挂钩之前执行。这些 Prolog 在“AssemblyLine 设置”面板中进行选择。可通过右键单击 AL，然后选择 **AssemblyLine 设置...**来访问该面板

3. 在算法中遇到易读性和美观性冲突的情况时，请优先选择易读性，记住，当您选择您自己六个月后的工作时，它可能看起来像其他某个人的。多为同事考虑对自己也有好处。
4. 请注意，没有配置编辑器使用技巧的人员可能需要修改设置并运行您的 AssemblyLine。可以从命令行方便地启动 AL:

   ```
   ibmdisrv –c myConfig.xml –r myAssemblyLine
   ```

   这意味着您可以准备脚本或批处理文件来帮助启动 AL。
5. 使用“属性”来使参数设置更具体化，从而使重新配置 AL 更简单。属性是可以存储在文件或数据库中的键值对，且将允许您从配置编辑器外部重新配置您的解决方案。通过单击参数标签，然后按**添加属性**，可将属性与组件参数联系起来。

   还可以通过 `system.getTDIProperty()` 和 `system.setTDIProperty()` 调用在脚本中查询和修改属性，这样您还可以通过外部属性设置对定制逻辑轻松地进行切换。

   可使用命令行实用程序 `bin/tdisrvctl` 在正在运行的服务器中进一步更改属性，这样您还可以启动和停止 AssemblyLine、查询状态并（重新）装入配置，且执行所有操作时都不必停止服务器。
6. 前面已提到，使用文件的相对路径可以更轻松地将您的解决方案移动到新安装中。建议使您的路径成为用来从中装入 Config XML 文件的目录的相对路径。可以通过 `{config.$directory}` 属性访问该路径，然后可以将该路径用于使用 **Text w/substitution** 选项的特定路径参数；例如:

   ```
   {config.$directory}/html
   ```
7. 前面已提过，但是尤其是构建将由其他人部署和运行的解决方案时，不要期望这些用户具有 IBM Security Directory Integrator 使用技巧。请提供用来启动您的 AssemblyLine（包括测试和验证 AL）的批处理文件/脚本。这些批处理文件/脚本可能很简单，例如，只是连接到数据源并返回报告以指示成功或失败。请注意，为了在控制台命令行打印消息，以便您的批处理文件/脚本返回状态信息，请使用服务器方法 `main.logmsg()`，而不是在教程练习中使用的 AL 版本: `task.logmsg()`。后一个调用只会将您的消息发送到日志。

以上只是一点指示信息。在其他 IBM Security Directory Integrator 文献和新闻组中可找到更多信息。

日志记录和审计

IBM Security Directory Integrator 使用 *log4j* 提供灵活的日志管理。可以在大量标准*附加程序*（包括用于 Unix syslog、Windows eventlog、日常文件和滚动日志的附加程序）之间进行选择。可以创建或下载以及使用新的附加程序。

缺省情况下，仅启用最少的服务器日志记录。启用最少的日志记录时，应该使用 FileRoller 附加程序来定义 AssemblyLine 的日志记录，方法是: 写入解决方案目录的 `"logs"` 子文件夹并将日志文件命名为与 AssemblyLine 相同的名称。因此，对于“CSV2XML”AL，将基于以下文件路径定义一组滚动日志文件: `logs/CSV2XML.log`。

请注意，您可以选择使用 `logmsg()` 方法定义消息的日志级别，具体做法是: 将以下某个关键字作为日志消息前面的第一个参数进行传递:

DEBUG、INFO、WARN、ERROR 和 FATAL。日志级别是向下包含的，因此 WARN 将包含 ERROR 和 FATAL，而 DEBUG 则表示将记录所有级别的消息。例如，类似以下内容的消息:

```
task.logmsg("DEBUG", "Updated: " + conn);
```

将只能由设置进行 DEBUG 级别的日志记录的附加程序发出。

通过在 task.logmsg() 调用之前添加用于检查属性值的 IF 语句调用，可以将审计消息添加至可从正在运行的服务器之外打开和关闭的解决方案。例如，以下脚本片段可能会出现在“数据流 – 更新成功”挂钩中:

```
if (system.getTDIProperty("MyProps","audit").equalsIgnoreCase("true"))
     task.logmsg("DEBUG", "Updated the following data: " + conn);
```

通过使用 tdisrvctl 命令行实用程序更改“MyProps”属性存储中“审计”属性的值，可以动态地打开或关闭正在运行的服务器的此类审计消息。

一般来说，记录过多信息好过记录过少信息。尽管您也不应使日志输出被过多信息拥塞。在混乱的日志输出中查找您感兴趣的消息可能会很困难。

连接问题

连接问题通常可分为两类: 初始化错误和连接丢失。

缺省情况下，在 AL 操作初始化阶段，所有组件都在 AL 操作一开始时就建立连接。如果由于某种原因连接在此时失败，那么将调用“Prolog – 发生连接错误时”挂钩，从而向您提供可用来发送警报甚至更改参数设置的脚本容器，并尝试再次连接。同样，如果在 AL 循环期间发生连接错误，那么请使用“数据流 – 在连接丢失时”挂钩处理此情况。

除了此定制处理外，连接器和功能组件还通过**连接错误**选项卡提供内置的*重新连接*功能。可在此处指示组件尝试重新建立丢失的连接，或者在发生初始化问题时继续尝试设置连接。

在发生初始化错误的情况下，除非遇到重复出现的问题（如 SSL 协商期间偶发性的超时，或者类似的偶发性连接问题），否则通常的做法是不启用重新连接。

但是，建议启用**丢失连接时自动重新连接**，这样可允许组件重新建立其连接，然后像没发生任何事情一样继续。请注意，对于*迭代器*方式的连接器，重新连接还将意味着迭代“光标”可能会复位，这样，循环将在结果集中的第一个条目处再次开始。当然，这对于知悉状态的迭代器（如“更改检测连接器”）来说不是问题，因为这些组件会自动使用状态信息在它们停止的位置进行恢复。

AssemblyLine 可用性

提高 AL 可用性意味着两点: 1）尽量确保 AssemblyLine 不会停止，以及 2）尽快重新启动已停止的 AL。对于像长时间运行迁移和同步这样的情况， AssemblyLine 在重新启动后可能还需要从发生故障时停止的位置继续。

如果发生未处理的异常， AssemblyLine 将停止。您可以通过处理所有的错误来避免这种可能性，对于 IBM Security Directory Integrator，这表示至少启用每个连接器和功能组件的“发生错误时采用缺省值”挂钩。通过启用“错误挂钩”，指示服务器在发生异常时仍继续。

正如您在教程练习期间所看到的，如果 AssemblyLine 由于发生错误而停止，那么您将获得堆栈跟踪，它的前面有关于出错地点和原因的信息。如果您通过启用“错误挂钩”防止 AL 停止，那么您有责任通过使用可用于脚本的特殊对象报告错误状态。

```
task.logmsg("ERROR", "[" + thisConnector.getName() + "] - " +
             error);
```

以上脚本示例使用了以下两个对象：始终引用正在执行的脚本所关联组件的 thisConnector 以及名为 error 的预定义变量。error 对象是一个条目，就像 work 和 conn 一样，并且它保存了像“status”、“connectorname”之类的属性[36]和“message”这样的属性，以及其他关于任何最近的错误情况的相关详细信息。由于该对象是“条目”对象，所以您可以使用 task.dumpEntry(error) 来显示其内容以及对属性名称的直接引用（例如: work.message），或者，因为所有的“条目”对象都可以根据需要将自身转变为字符串表示，您也可简单地像上面示例中一样将 error 附加到字符串消息之后。

要处理脚本中（例如在“属性映射”赋值以及“挂钩和脚本”组件中）发生的异常，请将代码合并在 try-catch 块中。这使您可以捕获异常并自己处理这些异常:

```
try {
 res = myLib.callToSomeFunctionThatMightFail();
} catch (excptn) {
 task.logmsg("Call failed with error: " + excptn);
}
```

除了处理错误之外，您将希望使用前一部分中描述的“自动重新连接”功能。这将使 AssemblyLine 避免由于暂时的连接问题（如由数据源或防火墙导致的超时）而发生故障。

如果由于某种原因导致 AssemblyLine 仍然过早地停止，那么您下一步要做的是重新启动该流水线。一个方法是使用 Web 管理工具定义故障/响应行为。

另一个更值得推荐的方法是创建“启动者”AL，并利用前一部分教程练习中使用的组装流水线功能组件。通过将该 AL FC 放入永不停止的条件循环（换句话说，其条件始终为 *true*）中，然后配置 AL FC 以调用所需服务 AL 并等待其完成，确保只要控制权返回到“启动程序 AL”，无限循环就将只需再次重新启动数据流即可。

如果基于某个更改检测连接器（或变化量引擎，这两者都在其他地方进行了相应描述）对同步 AL 应用上面列出的重新启动循环方法，那么 AssemblyLine 将自动从发生故障的位置继续。如果未这样做，那么您将肩负进行状态处理的重任。常用的方法是使用系统存储来保存状态信息（如时间戳记或已排序结果集的其他键值）。然后，只要初始化 AssemblyLine ，该状态信息就会应用于迭代器连接器，以在前一个已处理的条目之后立即恢复处理。

36. 您可能已注意到，“连接器”这个词有时与“组件”是同义的，这使得像 thisConnector 这样的变量也可以引用 FC 的组件、SC 的组件和属性映射组件。这对于 error 对象的“connectorname”属性同样适用，它也可以保存任何类型的组件的名称。

如果 AL 的迭代数据源不支持已排序返回，那么可能需要从最开始再次启动迭代。在这种情况下，请注意，“连接器更新”方式提供了将“输出映射属性”与目标系统中当前发现的属性进行比较的**计算更改**功能，如果未检测到差别，那么将跳过修改操作。

您可以通过在多个 IBM Security Directory Integrator 服务器（如 IBM MQ）之间引入安全传输来防止出现单点故障。在这种方式下，通过将数据甚至是处理指示信息放置到队列中，可使用任意数量的服务器（以及 AssemblyLine ）启动处理。在接收端，多服务器/AL 基于先来先服务原则选择这些数据和指示信息，并执行请求的工作。这不仅会产生更为健壮的解决方案，还允许通过添加发送方和接收方 AssemblyLine 方便地进行伸缩。

这是对大得多的主题的简短讨论。但是，我们的目标偏重于给予启发而不是指示特定方法。建议您查看社区 Web 站点和讨论组以获取更详细的建议和示例。

伸缩和性能

这又是一个值得我们花费比此处更长的篇幅讨论的主题。但是，即使是概括性的讨论也应当提及几点。

控制数据流速度的主要因素是 I/O；从源检索数据或将更改输出到目标花费的时间使 AL 中的处理时间显得不值一提。此外，协调连接也是非常耗时的，涉及安全握手时此问题尤其突出。设计和构建 AssemblyLine 时记住这一点将直接影响其性能。

例如，想像从客户机接收传入协议请求的基于服务器方式的一个 AL，如您在上次练习中构建的 HTTP Server 解决方案。为了执行实际的工作，接收到的每个请求都会使服务 AssemblyLine 启动。如果此处的 AL 必须初始化组件，那么每个请求转向需要的时间将至少为所有连接时间之和。

以下三种方法可以缓解这种情况:

- 让主要服务器方式的 AL 执行实际处理而不是进行分派，从而不需要针对每个请求建立和断开连接;
- 设计服务 AssemblyLine，以使其可用*手动/循环方式*调用。使用手动/循环方式可在初始化 AL FC 时初始化服务 AL。此外，AssemblyLine 功能组件仅对服务 AL 的每次调用驱动一次循环。使用此方式时，服务 AL 就像主调 AssemblyLine 的组件一样操作，必须构建它以适应此行为;
- 使用全局连接器池。此功能在 *引用* 中进行了描述。您可用该功能来定义在服务器启动时初始化并根据需要在 AssemblyLine 之间共享的连接器池。

提高性能的另一种方法是将繁重的处理任务分成多个同时进行的 AssemblyLine。以迁移任务为例，您可以启动多个 AL，其中每个AL 都处理源数据集的一个子集，而不是让一个 AssemblyLine 处理整个源数据集。由于您在调用 AssemblyLine 时可向它传递初始化参数，因此可以开发多次启动的单个 AL，该 AL 用一个过滤参数来控制实例应该处理的数据范围。

另一个方法是在您的解决方案中合并使用消息总线，如前面的部分中所述。在 IBM 的一些大型客户机中使用此方法已取得极大成功。

在网络链接不稳定或系统可用性很低从而影响处理速度的情况下，您可以将更多 AL 部署为后台任务，以将难以访问的数据与本地高速存储同步。尤其是在实施实时服务时，此方法可以确保对客户机请求具有令人满意的响应速度。

监视

使用 Web 管理工具可以快速简单地管理现成的 IBM Security Directory Integrator 解决方案。此集成服务控制台 (ISC) 插件是一个 AppServer 应用程序，可以监视在您的基础结构中任何数量的服务器上运行的任何数量的解决方案。除了用于查看 AL 统计信息的现成功能之外，您还可通过日志和启动/停止时间、Web 管理工具来定制运行状况和事件控制台以及定义调度和故障/响应行为，从而使您的 AssemblyLine 正常运行。

您还可以配置 IBM Security Directory Integrator 解决方案以发送状态事件，例如作为 SNMP 陷阱或使用系统的定制事件格式。可以很容易地配置系统以进行 JMX 管理和监视，例如使用 ITM。

AssemblyLine 调试器

虽然前面已提过，但在这里还要重复保证：使您自己熟练使用 AssemblyLine 调试器所花的时间会在加速解决方案设计、故障诊断和部署时得到十倍的回报。

附录. EasyETL 指南

“ETL”代表“抽取、变换和装入”，简言之就是从一个位置获取数据，根据需要予以更改，然后将其置于其他位置。“EasyETL”是 IBM Security Directory Integrator 的功能部件，它使您只需数次击键就能以交互方式快速完成此操作。

公共 ETL 示例为:

- 将数据库记录、Notes 文档、目录条目甚至待收邮件或 MQ 消息导出到文件;
- 将来自文件的数据装入系统或数据存储器;
- 将数据直接从一个系统迁移到另一系统，或在软件/模式更新的情况下在同一系统的各个版本之间迁移。

IBM Security Directory Integrator EasyETL 让您以数个直观步骤就解决这些和其他场景，产生适合一次性数据移动需求以及您基础结构中任务关键型数据流的解决方案。

新建 EasyETL 任务的第一步是选择输入源并选择要转移的属性。已处于该步骤，IBM Security Directory Integrator 让您可以运行 EasyETL 作业并收集已读取到复制缓冲区中用于粘贴的数据。如果需要变换或计算数据，那么 EasyETL 会让您添加变换，重新运行 ETL 作业并复制/粘贴已变换的数据。您还可以选择输出目标并让 EasyETL 作业将数据直接写入目标。

一旦 EasyETL 解决方案在按需要工作，IBM Security Directory Integrator 就可以生成用于启动和调度集成任务的命令行资产（批处理文件或脚本）。最后，EasyETL 利用 IBM Security Directory Integrator 的变更检测功能来快速将 ETL 作业变为数据同步任务。

注: 对 IBM Security Directory Integrator 用户而言的好消息是，每个 EasyETL 解决方案都是具有单个 AssemblyLine 的 IBM Security Directory Integrator 项目，而且可以在全功能的开发环境中将其打开。但是，一旦您在其中予以更改，它就不再可用作 EasyETL。

使用 EasyETL

启动 IBM Security Directory Integrator 配置编辑器并选择要使用的工作空间。 IBM Security Directory Integrator 首次启动时，将打开欢迎页面[37]。

37. 您随时可以通过从主菜单选择**帮助** > **欢迎**来返回到欢迎页面。

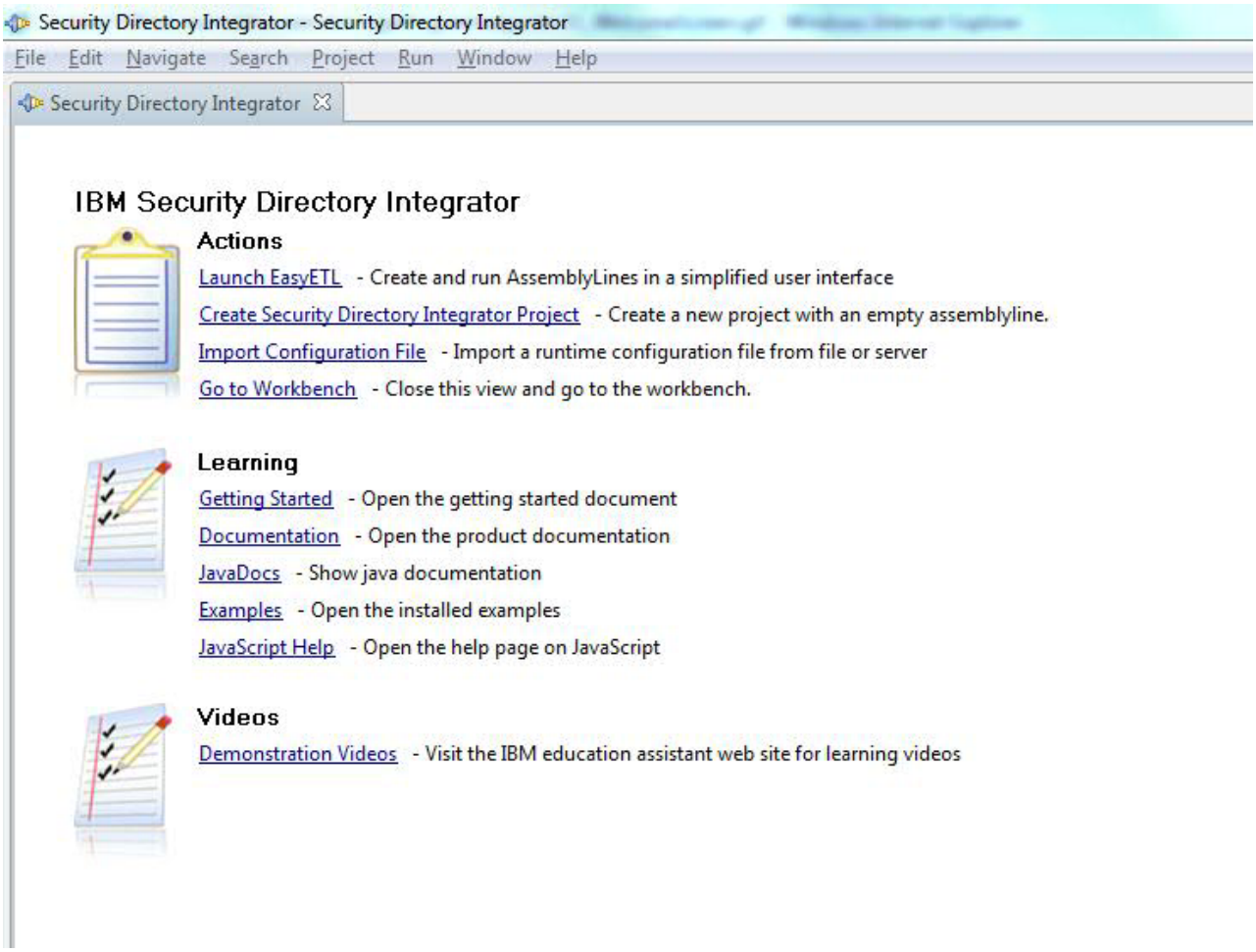

图 87. 欢迎屏幕

如以上屏幕快照中所示，最上方的链接在此打开 EasyETL[38] 工作台。现在，单击该链接以打开 Security Directory Integrator EasyETL 工作台。

38. EasyETL 是 Security Directory Integrator *透视图*，您可以使用菜单选择在透视图之间切换：**窗口** > **打开透视图** > **其他...** 如果您已对透视图进行更改且希望将其重置回缺省值，只需选择**窗口** > **重置透视图**。

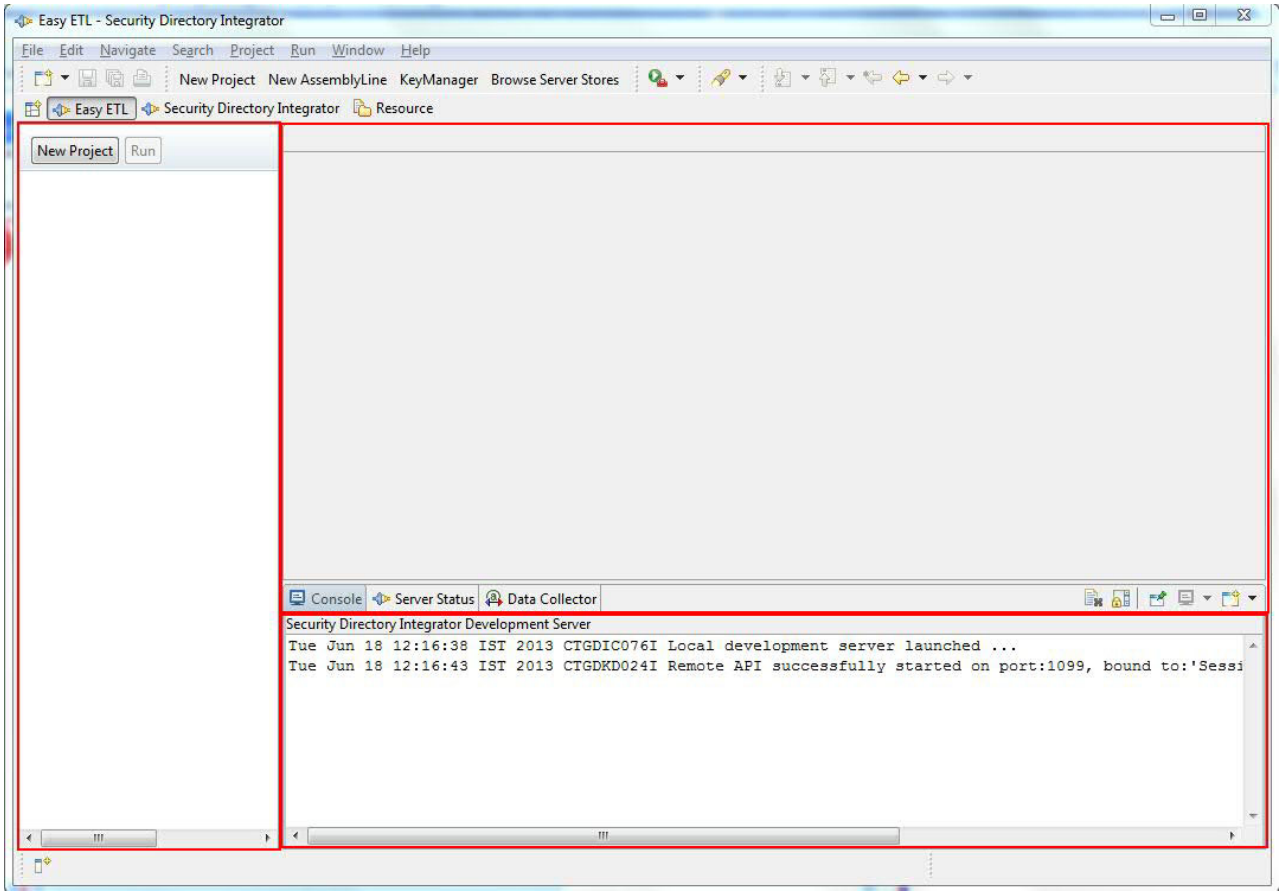

图 88. EasyETL 工作台

EasyETL 工作台向您显示三个项:

- 项目导航器（列出您的 ETL 作业）。您可以右键单击任何项目以执行操作，例如运行项目或创建命令行资产来启动项目;
- 每个打开的项目的简单 AssemblyLine[39]编辑器;
- 各种视图（作为屏幕底部的选项卡）。缺省情况下，您获得三个视图:
  - 来自测试 IBM Security Directory Integrator 服务器的控制台输出;
  - 服务器状态视图，其中您可以监视服务器和任何运行的 EasyETL 项目;
  - 数据收集器，其中在表格式列表中显示每个周期的结果数据。

在您通读本文档时，将发现关于每个部分的更多信息。

创建项目

通过在“项目导航器”的顶部按**新建项目**按钮来新建 EasyETL 项目。

39. “ AssemblyLine ”是 IBM Security Directory Integrator 中的数据流实施，所以创建或打开 EasyETL 项目时，在编辑器中将呈现底层 AssemblyLine 。

另请注意，术语“AssemblyLine”在本文献和其他 IBM Security Directory Integrator 文献中缩写为“AL”。

图 89. “新建项目”按钮

将该项目命名为“CSVtoXML”，然后按**完成**。该操作将在简单 AL 编辑器中打开新项目。

图 90. 简单 AssemblyLine 编辑器

该编辑器提供两个下拉列表：一个用于选择输入源，而另一个用于选择目标。在您选择源之前，以下区域为空（除协助文本之外）。

设置 ETL AL 的输入

通过单击源信息的下拉列表并选择**文件连接器**来配置 EasyETL AssemblyLine 的输入。

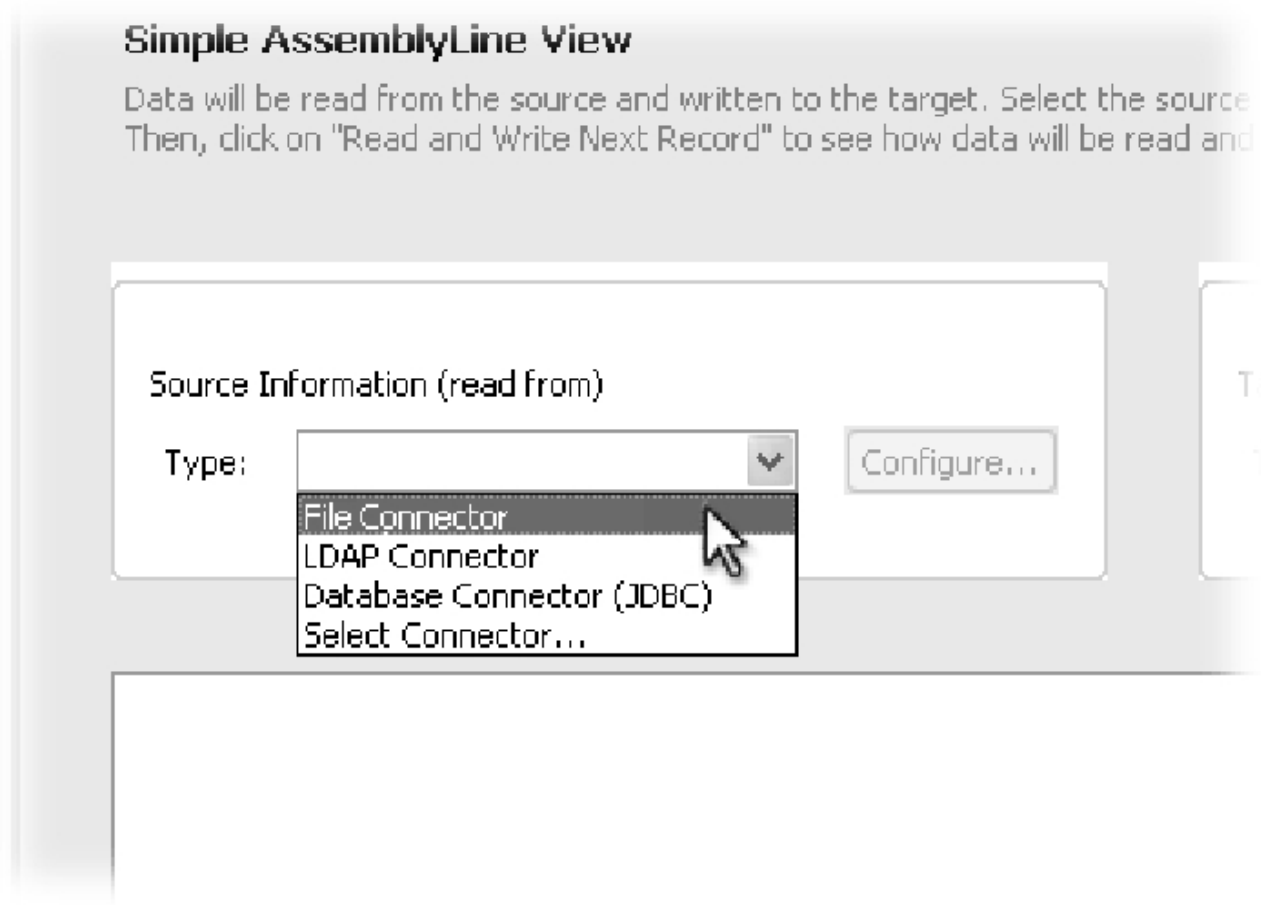

图 91. 选择源信息

然后，将向您呈现该连接器的配置对话框。

将 `File Path` 参数指向位于以下目录中的“`People.csv`”文件:

`TDI_HOME``/examples/Tutorial`

其中 *TDI_HOME* 替换为您机器上的 IBM Security Directory Integrator 安装目录[40]。

40. 请注意，IBM Security Directory Integrator 在 Windows 上运行时同时支持将正斜杠和反斜杠用作路径分隔符。如果使用正斜杠，IBM Security Directory Integrator 解决方案将更易于在运行 IBM Security Directory Integrator 的 Windows 及其他所有平台之间移植。

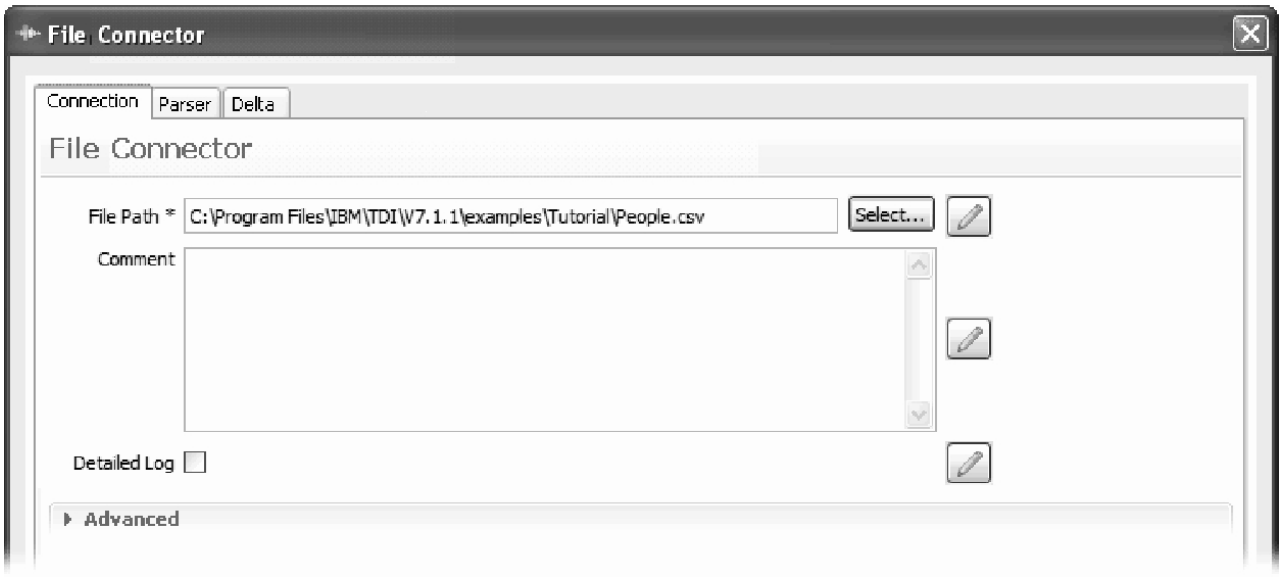

图 92. 设置 File Path 参数

现在，单击标记为**解析器**的选项卡并选择 **CSV 解析器**，将缺省配置参数保持原样。最后，按对话框底部的**连接**按钮以测试连接并发现可用的属性。

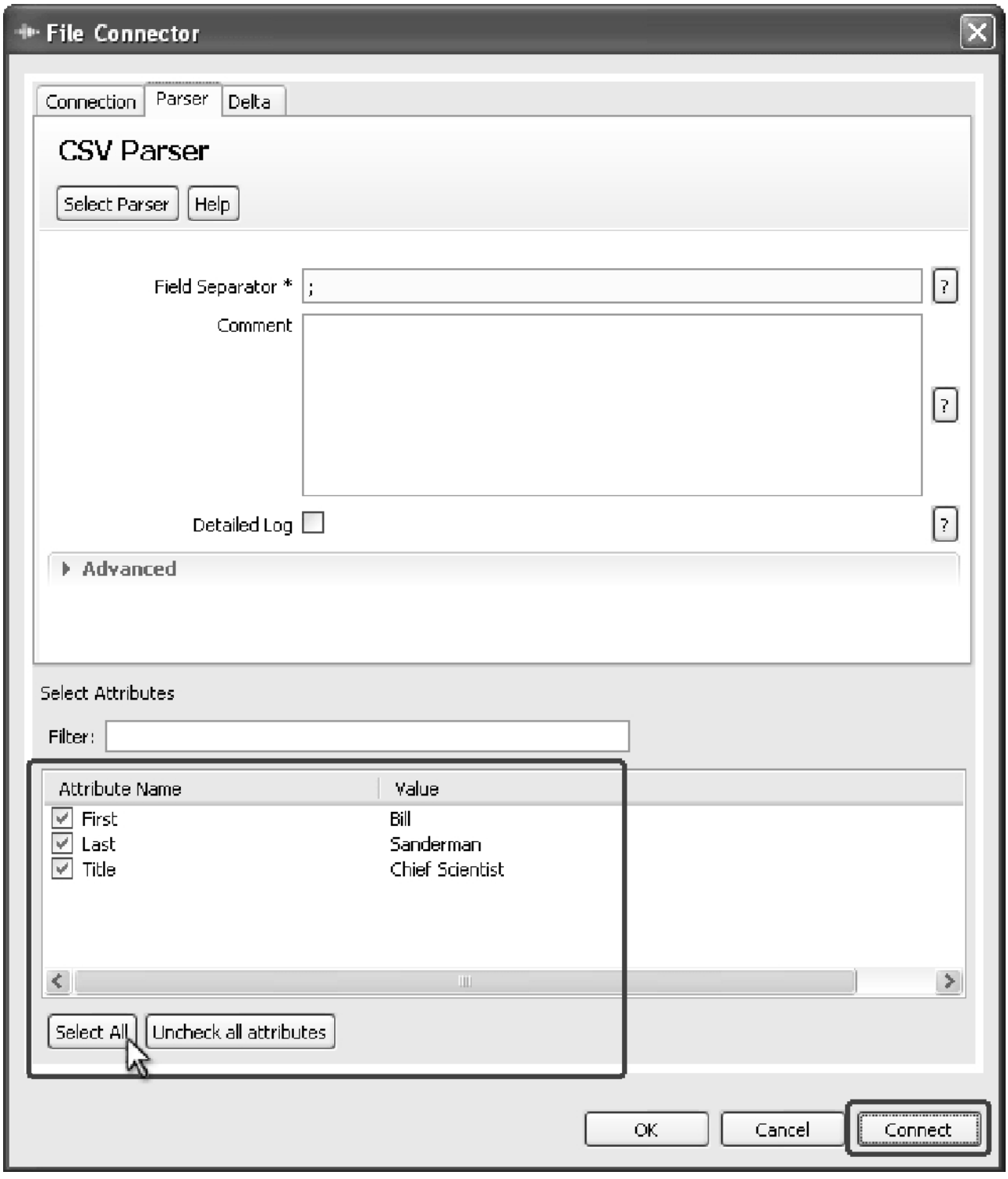

图 93. 测试连接和发现模式

此时会显示已连接系统的模式，从此处，您可以选择其中哪些要用于数据流。对于本示例，使用**全选**按钮，然后按**确定**以关闭配置对话框。

回到 EasyETL 工作台中，您将看到简单 AL 编辑器的下半部分已更改，表明您现在已配置输入源。现在，IBM Security Directory Integrator 向您提供用于步进浏览该信息（一

次一条记录）的按钮，以及用于将 ETL 任务运行到完成和将其停止的按钮。

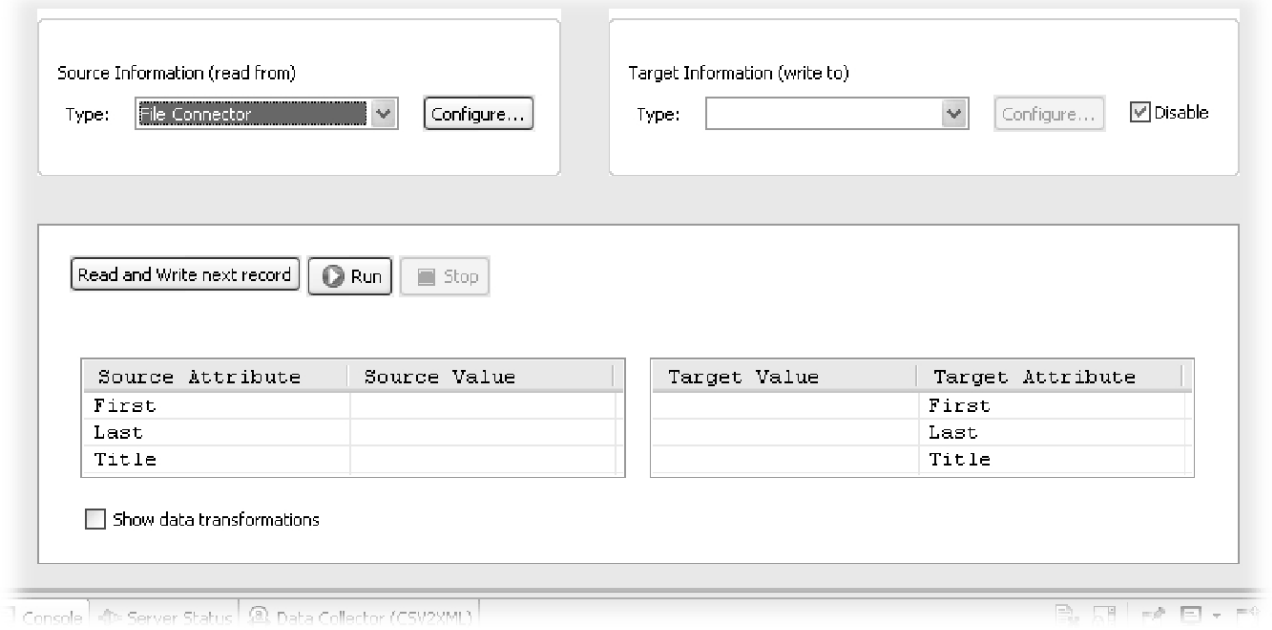

图 94. 配置的输入源

这些按钮之下是称为*数据查看器*的两个网格框，这些查看器列出由数据流处理的属性。左侧的数据查看器显示输入属性。上一步中选择用于读取的属性以粗体显示在顶部，称为*输入映射*。以下是以灰色显示的所有未选择属性，而您可以通过对其双击来包含这些属性以供输入映射。类似地，通过对其双击或将其删除来从映射除去属性[41]。

右侧的框是输出查看器，其中显示要编写的属性集。在 IBM Security Directory Integrator 术语中，这是*输出映射*，缺省情况下，它与您为“输入映射”所选的列表相同。请注意，可以通过单击右侧列，然后编辑值来更改任何输出属性的名称。使用该技术将称为“Title”的属性重命名为“JobTitle”。

41. 您还可以右键单击并选择**添加属性**或**除去属性**。

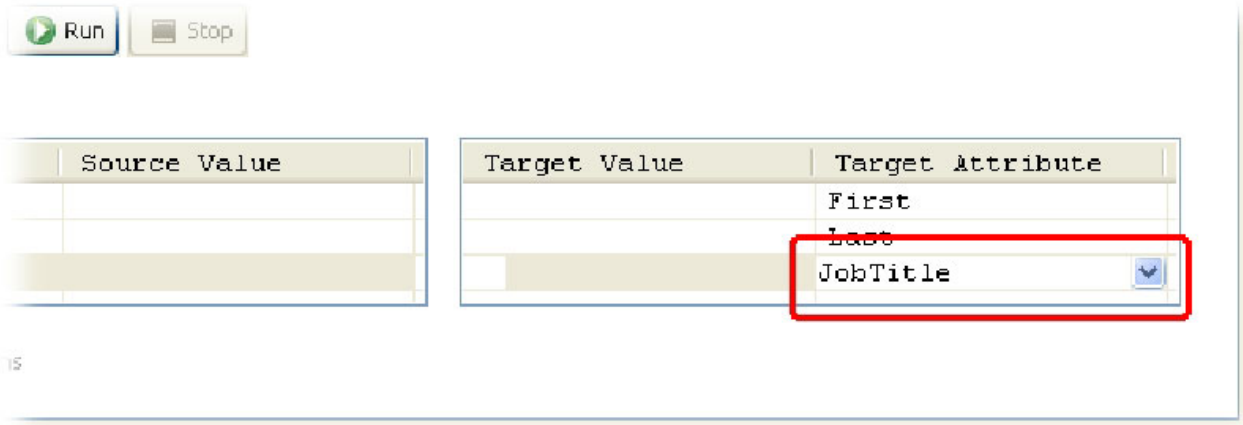

图 95. 重命名输出属性

现在，EasyETL 项目已经准备好进行其第一次测试。

运行 EasyETL AssemblyLine

选择位于屏幕底部的**数据收集器**视图，然后按**读取和编写下一条记录**按钮。该操作导致发生以下情况:

1. 由于 EasyETL AssemblyLine 转移到运行中的服务器并启动而存在延迟;
2. 从 CSV 输入源读取和解析第一条记录，而在输入和输出网格显示屏中显示数据;
3. 在“数据收集器”视图中编写和收集为输出选择的属性[42].

所以即使尚未选择输出目标，您仍然可以运行和测试 ETL 项目，在数据流过 AssemblyLine 时查看数据。

42. 位于**数据收集器**视图顶部最左侧的按钮打开配置对话框，您可在其中增加**数据收集器缓冲区**大小。请注意，如果计划收集大量数据，那么可能需要增大可用于 IBM Security Directory Integrator 的内存。通过找到 IBM Security Directory Integrator 安装文件夹中的“ibmditk”批处理文件/脚本并在编辑器中将其打开，可实现该操作。底部附近是启动“miadmin”的行，您可以在 -vmargs 选项后插入以下文本: -Xmx512M 该示例将允许 IBM Security Directory Integrator 数据内存增加到 512 兆字节。

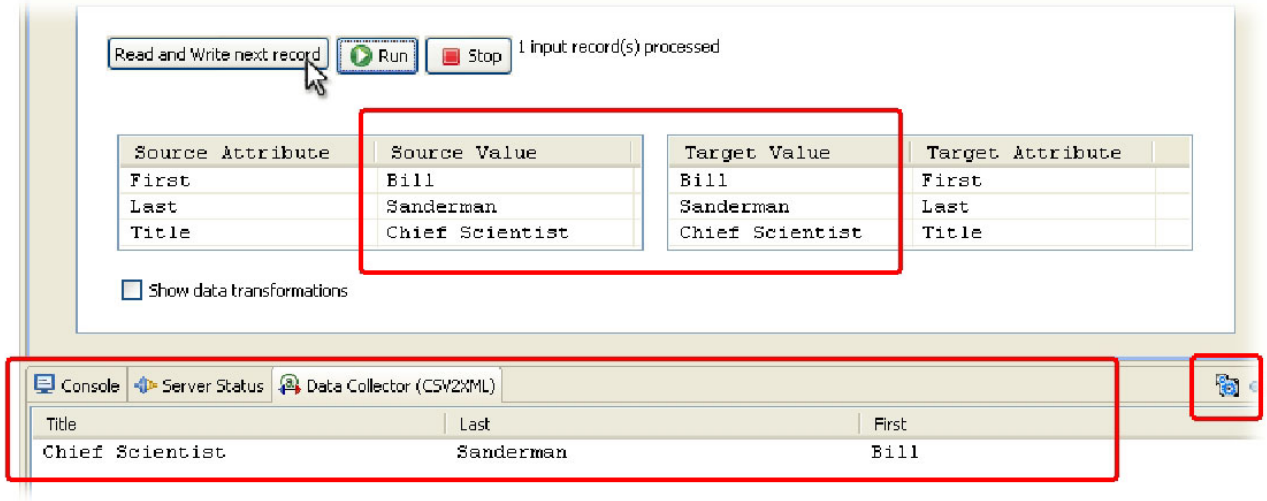

图 96. 读取和收集的一条记录

每次按**读取和编写...** 按钮，就读取、显示和收集另一条记录。如果现在按“运行”按钮，那么 ETL 作业将运行到完成，而您将看到此 AL 报告。

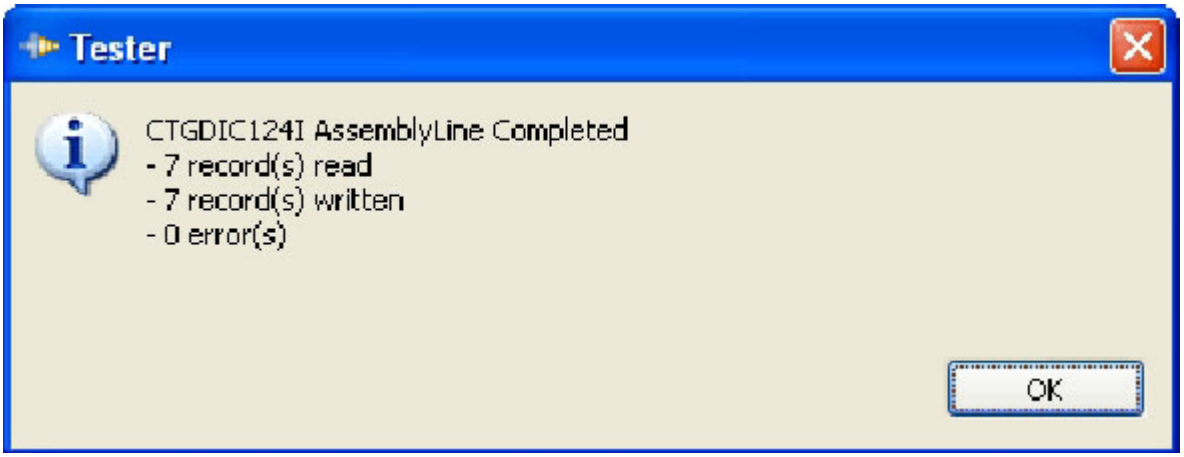

图 97. 完成的 EasyETL AssemblyLine

如以上对话框中所示，在任何位置均未实际编写任何记录。但是，**数据收集器**仍然提供关于所抽取和转移信息的方便可见的反馈。

而且，您可以选择所收集数据中的行并将该信息复制/粘贴到文件或其他目标[43]。

变换

到目前为止，输出值与输入完全相同。但是，在有些情况下，您希望处理或者甚至计算这些基于读取的数据。通过用 JavaScript 编写变换可在 IBM Security Directory Integrator EasyETL 中执行该操作。

43. 用 CSV 格式复制数据以简化将值导入电子表格和报告表的过程。“CSV”代表字符分隔值，而 IBM Security Directory Integrator 所用的分隔符是分号 (;)。

为了使用变换，您必须首先通过选中**显示数据变换**复选框来启用变换。

图 98. 启用变换

这导致在数据查看器之间显示新的网格框：变换查看器。在此您看到箭头，其指示所有三个输出属性直接从输入属性获取其值 – 换言之，没有进行变换。您将定义新的输出属性，然后添加“变换”脚本以计算其值。现在，通过在输出映射中右键单击，选择“添加属性”并将其命名为“FullName”，可执行该操作。现在，双击此新输出属性左侧的“变换”，然后输入以下脚本片段：[44]

```
return First + " " + Last
```

44. 请注意，您可以按 Ctrl + 空格来获取可输入内容的建议列表。该列表包含一些特殊对象（如“task”和“main”）并列出从底部的输入源读取的属性。

    另请注意，在此显示的用于访问输入属性的表示法仅在 EasyETL 中有效。在完整 IBM Security Directory Integrator 工作台中，您必须在属性名称前面添加具有该属性名称的条目对象 - 例如，Work 条目 (work)。因此，以上示例形如以下内容：

    ```
    return work.First + " " + work.Last
    ```

    在本文档的第一部分中讨论了 Work 条目和其他 IBM Security Directory Integrator 概念。

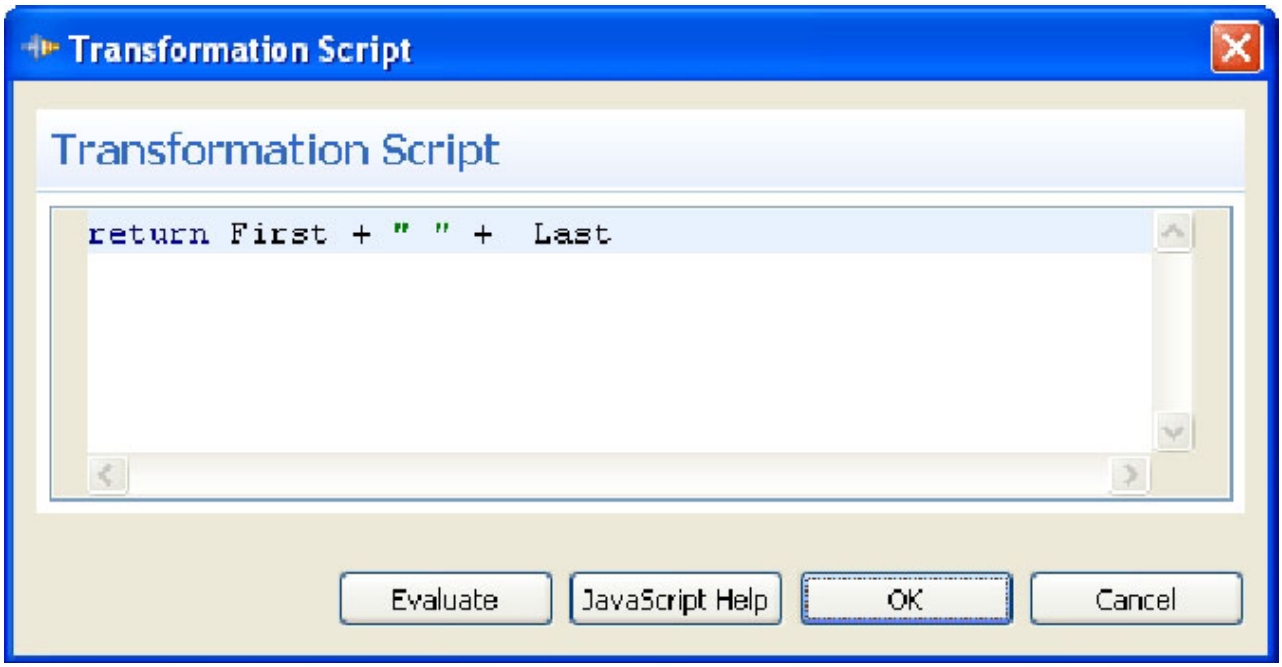

图 99. 显示变换脚本

变换脚本编辑器对话框中有一个用于测试脚本的**评估**按钮，以及一个用于显示若干 JavaScript 提示和示例的按钮。

现在，按**评估**以了解您的变换的工作情况。

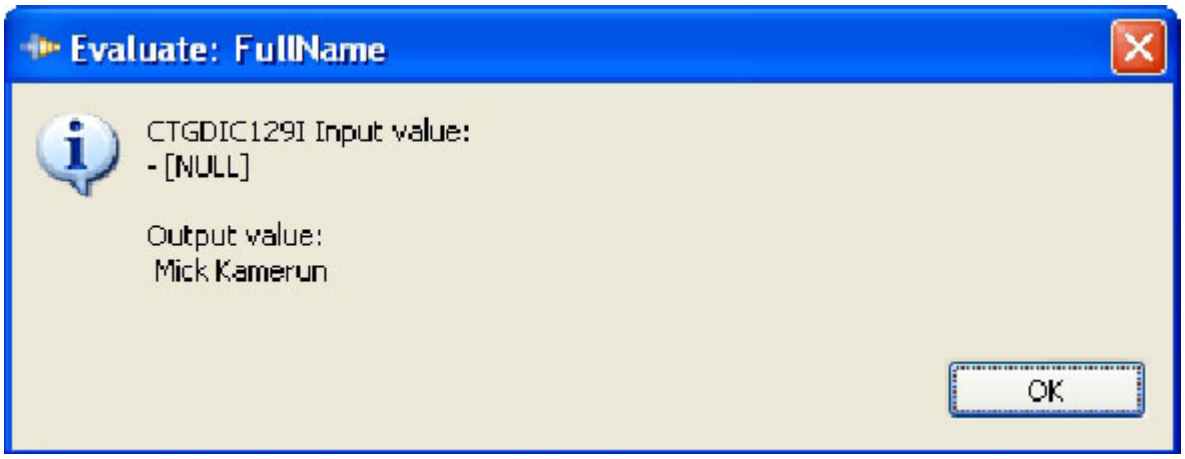

图 100. 评估表达式

所显示的输出值是使用运行 AssemblyLine 时收集的数据来计算的。通过按**确定**来关闭评估结果对话框。

现在，按**确定**以接受该脚本并关闭变换脚本编辑器，通过按**运行**并查看收集的条目来重新运行 EasyETL AssemblyLine。请注意，现在**数据收集器**为您提供两个组件集合进行选择：*输出*和*输入*。选择**输出**并查看变换脚本如何为每个条目生成“FullName”值。

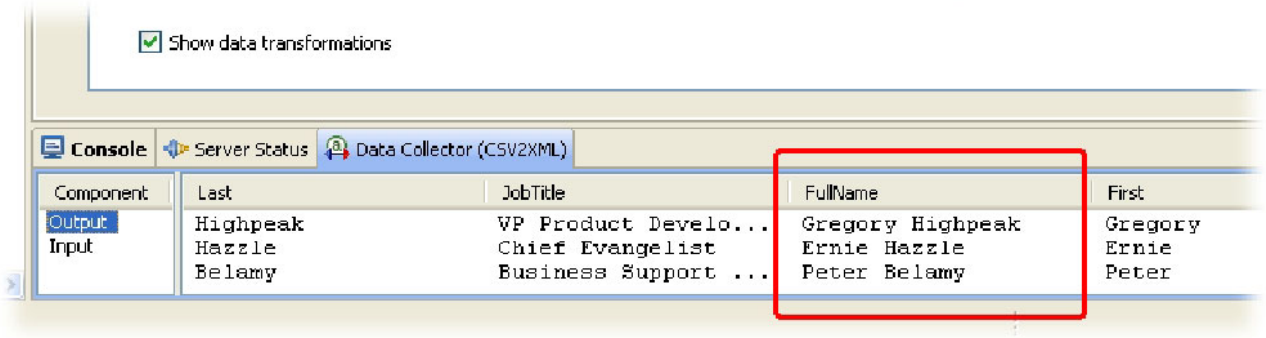

图 101. 带有计算的 FullName 属性的输出集合

所以，现在您知道如何设置输入源并选择要抽取的属性，以及如何变换该数据以适合您的输出需求。下一步是选择输出目标并将数据驱动到目标处。

选择输出目标

使用输出视图之上的下拉列表为输出目标选择**文件连接器**，将其写入名为“Output.xml”的文件[45]。选择 **XML 解析器**，然后按**确定**。请注意，您可以使用连接按钮来确保输入的文件路径有效。但是，没有要发现的数据 – 当然，除非您将连接器指向现有 XML 文件。

一旦配置了输出，请再次运行 ETL AssemblyLine 。一旦完成，您可以打开输出文件并验证结果。

45. 如果没有输入完整路径或相对路径，那么 IBM Security Directory Integrator 将以安装期间指定的“解决方案目录”的路径为基础。

```
- <DocRoot>
  - <Entry>
      <First>Bill</First>
      <Last>Sanderman</Last>
      <Title>Chief Scientist</Title>
    </Entry>
  - <Entry>
      <First>Mick</First>
      <Last>Kamerun</Last>
      <Title>CEO</Title>
    </Entry>
  - <Entry>
      <First>Jill</First>
      <Last>Vox</Last>
      <Title>CTO</Title>
    </Entry>
  - <Entry>
      <First>Roger</First>
    </Entry>
  - <Entry>
      <First>Gregory</First>
      <Last>Highpeak</Last>
      <Title>VP Product Development</Title>
    </Entry>
  - <Entry>
      <First>Ernie</First>
      <Last>Hazzle</Last>
      <Title>Chief Evangelist</Title>
    </Entry>
  - <Entry>
      <First>Peter</First>
      <Last>Belamy</Last>
      <Title>Business Support Manager</Title>
    </Entry>
</DocRoot>
```

图 102. XML 输出

您的输出目标可能就是数据库表而已，就像您的输入可以来自 Lotus Notes 应用程序或 LDAP 目录。无论所使用的系统或数据存储器如何，创建该示例 ETL 作业所执行的步骤都是相同的。

检测变更

IBM Security Directory Integrator 提供若干功能来检测输入数据中的变更。不仅提供了一组“变更检测连接器”，您还可以选择为输入源启用*变化量引擎*。

读取数据时，变化量引擎生成数据的快照，然后将这些快照与上次运行期间生成的快照相比较，以确定哪些已变更。将跳过未变更的条目，只检索已修改的条目以便在 EasyETL AssemblyLine 中处理。

按输入源的**配置**按钮，然后选择**变化量**选项卡。

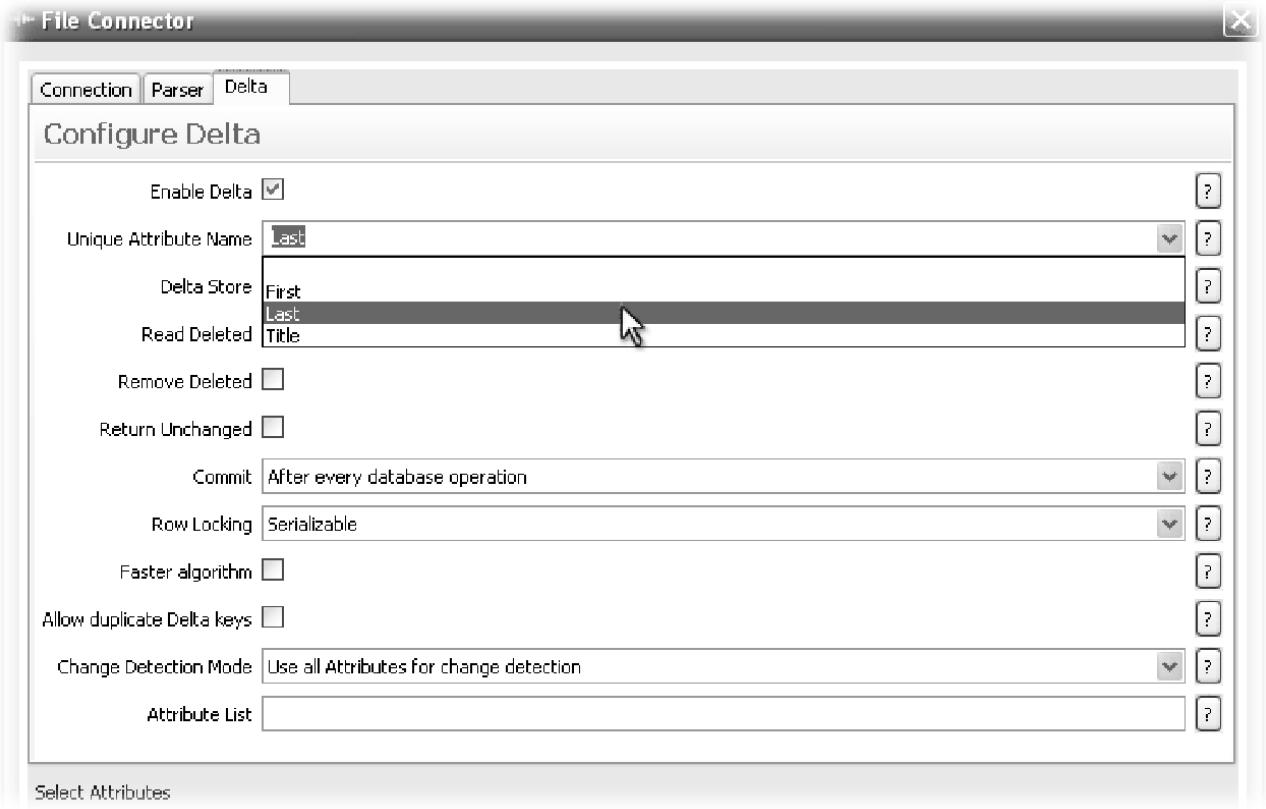

图 103. 变化量配置

您必须通过选中配置面板顶部的复选框来首先启用“变化量引擎”。然后使用下拉列表来选择“第一”作为**唯一属性名**[46]。

此处提供若干其他参数，其中一些参数更适合在标准 IBM Security Directory Integrator 工作台中使用，而不适合在 EasyETL 中使用。例如，虽然 EasyETL AL 可以检测并转移新的和修改的条目，但是它不会处理从数据库删除行或目录中的条目。但是，它会

46. 如您所推论，“变化量引擎”使用输入属性之一来对快照进行唯一标识。如果在输入数据中没有可用的唯一值，那么您可以指定多个属性，这些属性将并置在一起以形成快照标识。您可以通过输入以加号 (+) 分隔的多个属性名来完成该操作。例如: `First + Last`

将该信息写入输出目标，例如带有 LDIF 解析器的文件连接器。 LDIF 文件可以包含变更操作标记，而一些系统支持 LDIF 导入。

您可以在此处了解关于 IBM Security Directory Integrator 的完整变化量处理功能的更多信息:

http://www.tdi-users.org/twiki/pub/Integrator/HowTo/HowTo_SyncData_6.1.1070523.pdf

您可能希望对**落实**参数进行变更。这控制新的快照和已变更的快照落实到 IBM Security Directory Integrator 系统存储数据库的时间。缺省情况下，这设置为“每个数据库操作后”，因而在读取阶段内发生。

但是，如果希望确保在落实快照之前已成功转移变更，请改为将该下拉项设置为“在 AL 周期结束时”，使之在更新了输出目标后发生。

为了让“变化量引擎”执行其工作，它需要基线快照集。您通过在首次启用变化量后运行 ETL 作业来予以创建。一旦完成，您将注意到，弹出报告的次数是写入的发生次数的两倍。这是因为 IBM Security Directory Integrator 也对写入系统存储的快照计数，所以对于处理的每个条目，您获取两次写入。

重试运行 EasyETL AssemblyLine，而您将发现此次没有写入任何条目。“变化量引擎”检测到输入记录均未变更而将其跳过。

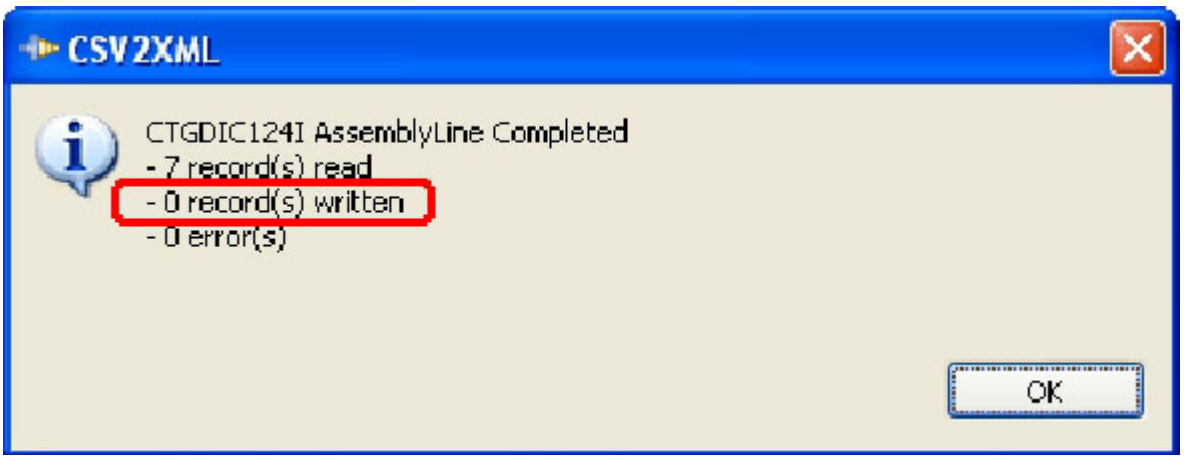

图 104. 未变更并跳过的所有条目

作为最终测试，提出输入 CSV 文件并更改任何字段值 – 除“最后一个”之外[47]。保存变更，然后重新运行 ETL 作业，而您将发现只处理了已修改的条目。

配置更新的输出目标

当前设置对于到文件的输出很有效。但是，如果您将这些变更驱动到目录、RDBMS 或类似的数据存储器，那么您将希望添加新数据并更新现有记录。为了让 EasyETL 作业执行该操作，您必须首先选择将哪个输出属性用作条件以查找要修改的记录。

这是通过在期望的输出属性上右键单击并选择**用作链接条件**选项来完成的。

47. 由于这是用于标识快照的属性，所以对条目的该属性值的任何变更将使其对“变化量引擎”显示为*新*记录。

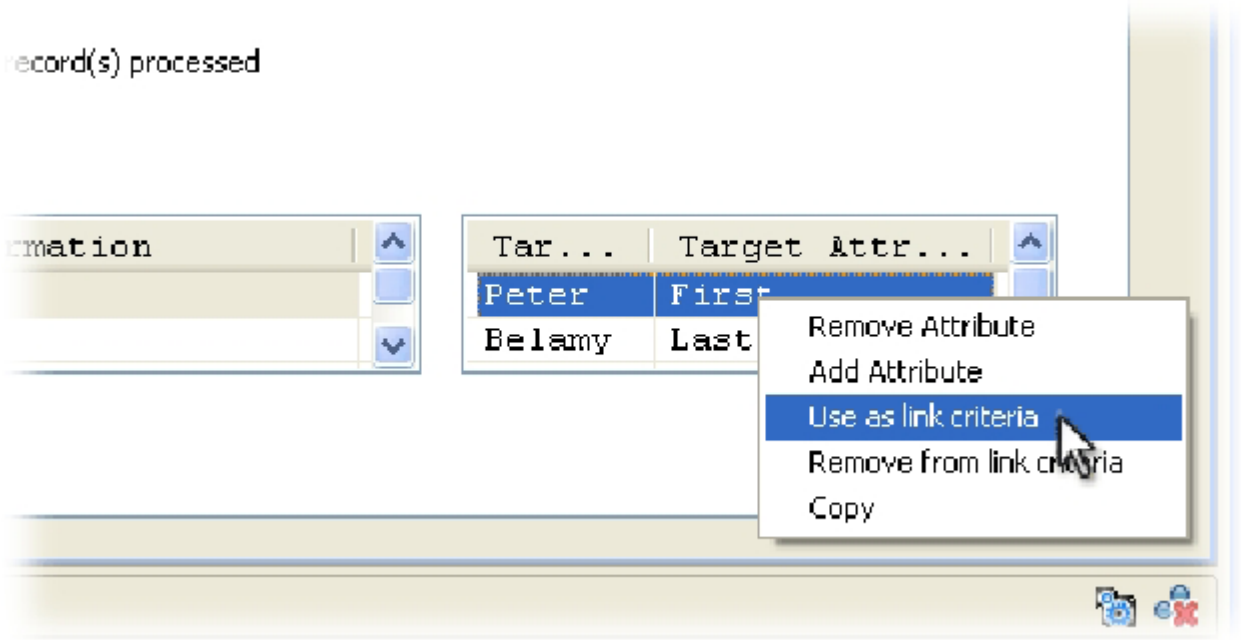

图 105. 选择链接条件

现在，输出连接器写入目标时，它首先使用指定的“链接条件”属性来搜索记录。如果找不到匹配项，那么将添加新条目。如果匹配成功，那么将更新该记录。

操作就是这样简单：现在，已配置您的 ETL 作业来提供输入源与输出目标之间的持续同步。

用于运行和调度 ETL 作业的命令行资产

一旦 ETL AssemblyLine 准备好部署，您就可以在导航器中的项目上右键单击，并选择 **创建所需文件...** 选项。

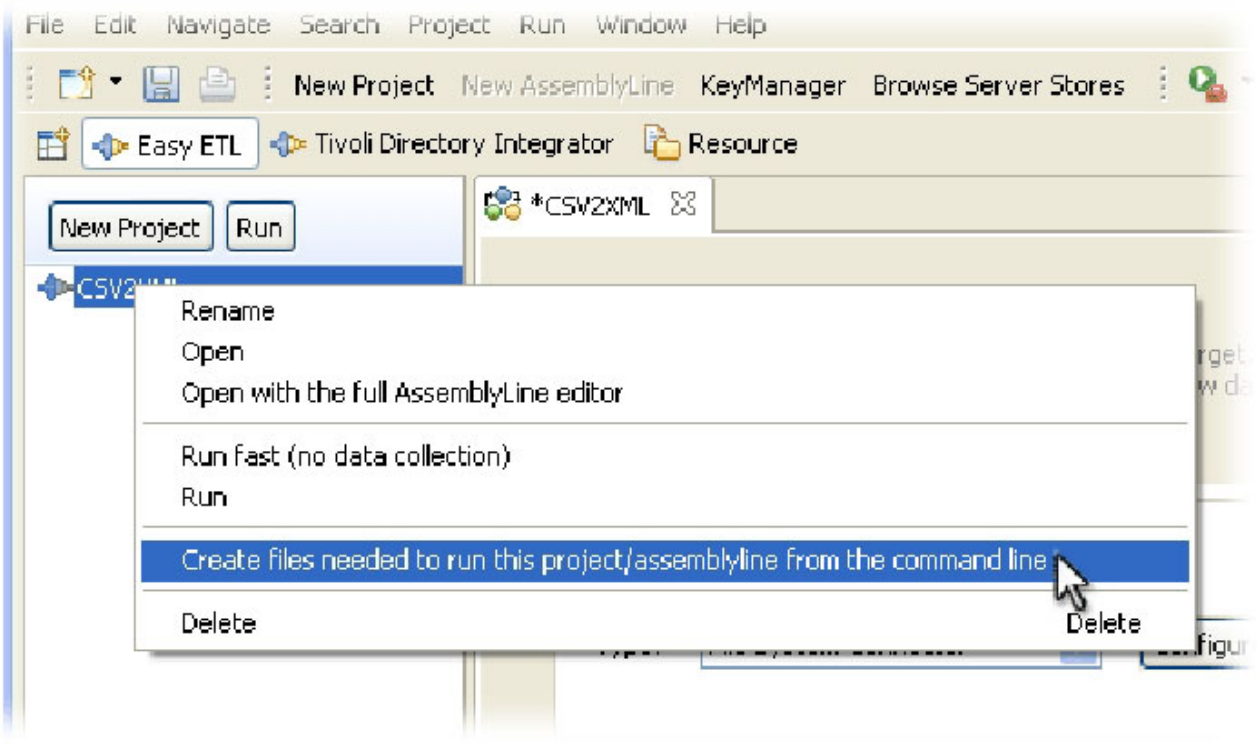

图 106. 创建命令行资产以运行 ETL 作业

这将弹出“导出文件”对话框，在其中编写该脚本/批处理文件。请注意，将给予其与项目相同的名称，因此对于在 Windows 上运行的本教程练习而言，会将其称为“CSV2XML.bat”。从命令行执行 EasyETL 项目为您的解决方案提供了最佳的性能。

您还将在相同位置创建 XML 文件。这称为 IBM Security Directory Integrator *配置*文件，其中包含 EasyETL AssemblyLine 的详细信息，IBM Security Directory Integrator 服务器运行该 AssemblyLine 需要这些详细信息。如果在文本编辑器中打开生成的脚本，那么将看到启动 IBM Security Directory Integrator 服务器所需的单行代码，请将其指向 Config，然后指定要运行的 AssemblyLine。现在，您只需设置调度任务或 cronjob 来定期调用该脚本，这样您的同步/迁移服务就会生效。

其他选项

高速 ETL

虽然数据收集器这一工具的功能很强大，但是由于进行数据收集并在屏幕上呈现，ETL AssemblyLine 运行会变慢。如果您转而希望 EasyETL AL 处理尽可能快，那么可以选择项目并按导航器顶部的“运行”按钮，或者右键单击项目并选择**快速运行...** 选项。

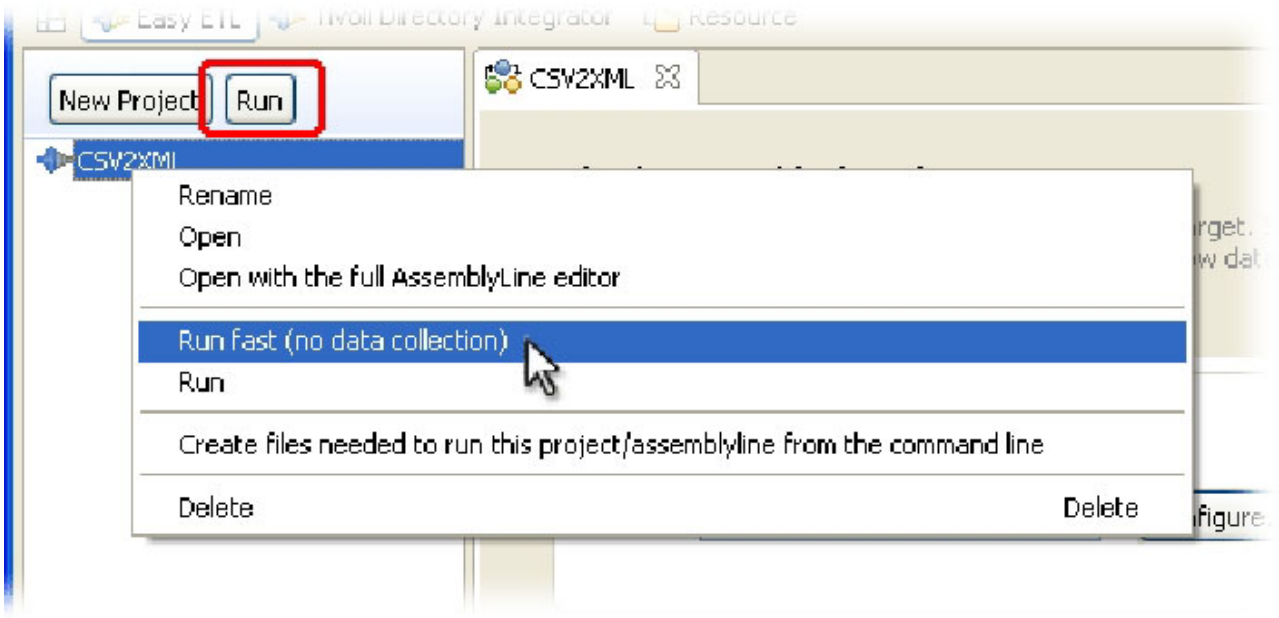

图 107. 全速运行 ETL 作业

任一选项将打开控制台显示器，在 AL 全速执行时，该显示器中将显示来自 AssemblyLine 的日志消息。

请注意，项目上下文菜单中的**运行**选项将 ETL 作业与数据收集一起运行。

过滤输入数据集

另一强大的功能是控制输入数据集的内容的能力。只要输入源是数据库或目录，该功能就可用。

例如，将“LDAP 连接器”选为输入并查看该组件的配置对话框。**搜索过滤器**参数旁是标记了三个点 (...) 的按钮。该按钮打开“链接条件”编辑器，在其中您可以定义搜索规则，将应用这些规则来构建结果集以供该连接器读取。

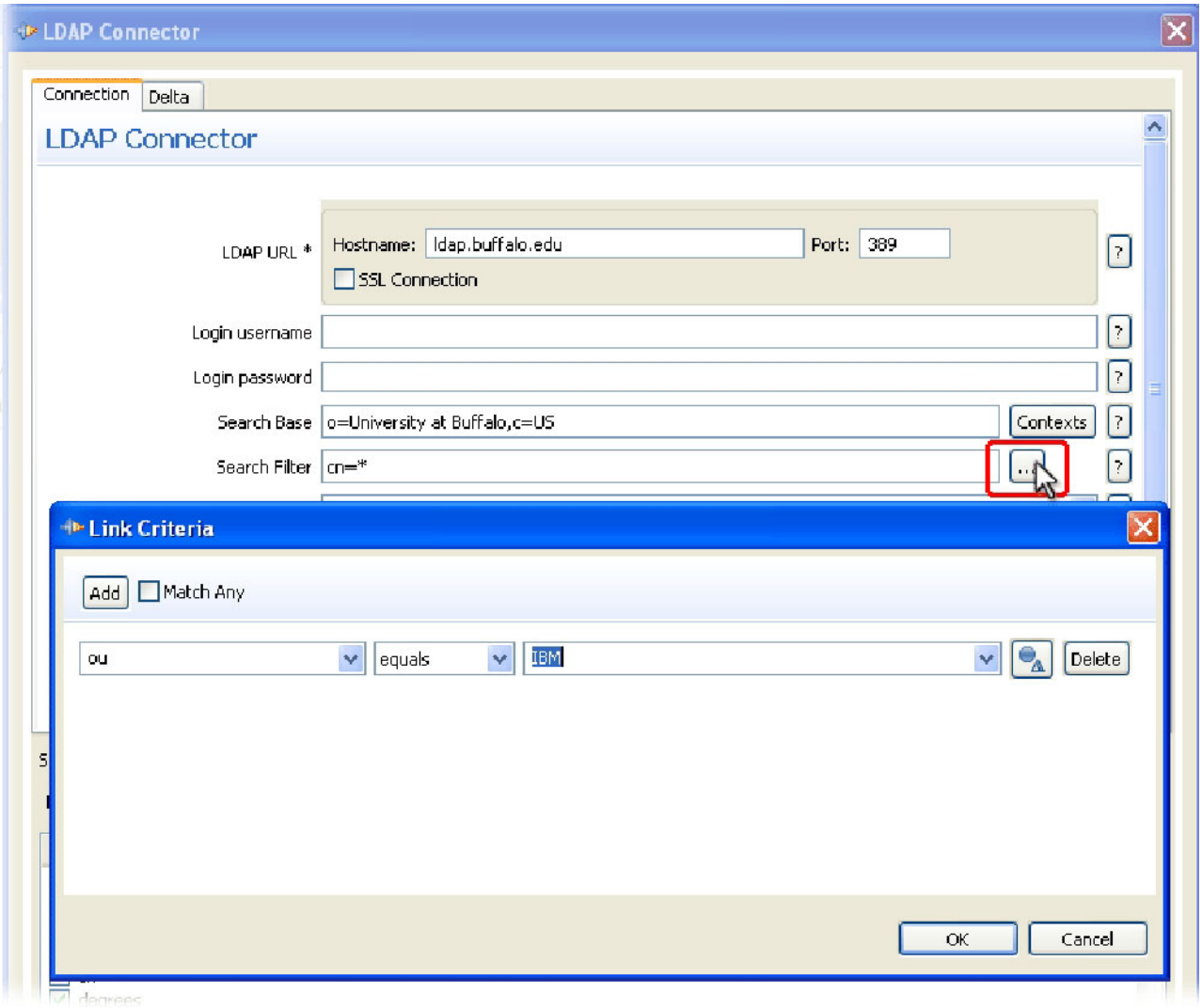

图 108. 定义输入连接器的链接条件

该相同功能可用于数据库和 JDBC 连接器，其中您将在**选择**参数旁找到 (...) 按钮。

虽然您自己可以在搜索参数中直接输入 LDAP 搜索语法，但是这需要您了解 LDAP 搜索过滤器或 JDBC Select 语句的语法。表达期望的选择的更简单方法通常是，使用“链接条件”并让连接器处理底层语法。

将 EasyETL AssemblyLine 带入下一级别

在全功能的 IBM Security Directory Integrator AssemblyLine 编辑器中打开 ETL 项目，可让您将定制记录和审计、错误处理、故障转移逻辑、自动重新连接、数据增加（连接）以及更多项添加到迁移或同步解决方案中。您可以通过右键单击项目并选择**以完整 AssemblyLine 编辑器打开**选项来执行此操作。您将仍然在 EasyETL 工作台中工作，但是可以访问可用于 AssemblyLine 的其他功能。

如果您对这种方式感到满意而准备尝试，那么切换到 Security Directory Integrator 透视图（**窗口** > **打开透视图** > **Security Directory Integrator**），并

开始在完整 IBM Security Directory Integrator 工作台中工作。更好的是 - 既然您已掌握 EasyETL，请返回到第一部分并开始挖掘 IBM Security Directory Integrator 的完整功能。

声明

本信息是为在美国提供的产品和服务编写的。IBM 可能在其他国家或地区不提供本文档中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

有关双字节字符集 (DBCS) 信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区:

International Business Machines Corporation“按现状”提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。

某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本出版物的新版本中。IBM 可以随时对本出版物中描述的产品和/或程序进行改进和/或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的: （i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系:

IBM Corporation
2Z4A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文档中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的。实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

所有显示的 IBM 的价格均是 IBM 的当前建议零售价，可随时更改而不另行通知。经销商的价格可与此不同。

本信息仅用于规划的目的。在所描述的产品上市之前，此处的信息可随时更改。

这些信息包含日常业务操作中使用的数据和报告示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称均是虚构的，如与实际的商业企业使用的名称和地址有任何相似之处，纯属巧合。

版权许可:

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口 (API) 进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。如果是为按照 IBM 应用程序编程接口 (API) 来以应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。

凡这些样本程序的每份拷贝或其任何部分或任何衍生产品，都必须包括如下版权声明:

© (贵公司名称) (年份). 此部分代码是根据 IBM 公司的样本程序衍生出来的。© Copyright IBM Corp.（输入年份）。All rights reserved.

如果您正以软拷贝格式查看本信息，图片和彩色图例可能无法显示。

Trademarks

IBM、IBM 徽标和 ibm.com® 是 International Business Machines Corp.，在全球许多管辖区域的商标或注册商标。其他产品和服务名称可能是 IBM 或其他公司的商标。IBM 的最新商标列表，可从 Web 站点 www.ibm.com/legal/copytrade.shtml 上的“版权和商标信息”部分获取。

Adobe、Acrobat、PostScript 和所有基于 Adobe 的商标是 Adobe Systems Incorporated 在美国和/或其他国家或地区的商标或注册商标。

IT Infrastructure Library 是英国中央计算机与电信局（现在隶属于英国商务部）的注册商标。

Intel、Intel 徽标、Intel Inside、Intel Inside 徽标、Intel Centrino、Intel Centrino 徽标、Celeron、Intel Xeon、Intel SpeedStep、Itanium 和 Pentium 都是 Intel Corporation 或其子公司在美国和其他国家或地区的商标或注册商标。

Linux 是 Linus Torvalds 在美国和/或其他国家或地区的商标。

Microsoft、Windows、Windows NT 和 Windows 徽标是 Microsoft Corporation 在美国和/或其他国家或地区的商标。

ITIL 是英国商务部的注册商标和注册的欧盟商标，并在美国专利和商标局注册。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Java 和所有基于 Java 的商标和徽标是 Oracle 和/或其子公司的商标或注册商标。

Cell Broadband Engine 是 Sony Computer Entertainment,Inc. 在美国和/或其他国家或地区的商标，并依据许可证使用。

Linear Tape-Open、LTO、LTO 徽标、Ultrium 和 Ultrium 徽标都是 HP、IBM Corp. 以及 Quantum 在美国和其他国家或地区的商标。

索引

N

T

W

Printed in China

G151-1408-03