Novell Identity Manager Driver for SOAP

1.0.2

www.novell.com

实施指南

2005 年 11 月 23 日

Novell®

法律声明

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

联机文档：要访问本产品和其它 Novell 产品的联机文档并获取产品的更新资料，请参见 www.novell.com/documentation。

Novell 商标

DirXML 是 Novell, Inc. 在美国和其它国家 / 地区的注册商标。
eDirectory 是 Novell, Inc. 在美国和其它国家 / 地区的商标。
NetWare 是 Novell, Inc. 在美国和其它国家 / 地区的注册商标。
Novell 是 Novell, Inc. 在美国和其它国家 / 地区的注册商标。
Nsure 是 Novell, Inc. 在美国和其它国家 / 地区的注册商标。
SUSE 是 Novell, Inc. 在美国和其它国家 / 地区的注册商标。

第三方材料

所有第三方商标是其相应拥有者的财产。

目录

关于本指南

本指南介绍了如何安装和配置 Identity Manager Driver 1.0 for SOAP （也称作 SOAP 驱动程序）。

读者

本指南面向实施 Identity Manager 的 eDirectory™ 管理员、应用程序服务器开发者、万维网服务管理员以及咨询人员。您还应对 DSML/SPML、 SOAP 和 HTML 有所了解。

反馈

我们希望听到您对本手册和本产品中包含的其它文档的意见和建议。使用联机文档中每页底部的 《用户意见》功能，或访问 www.novell.com/documentation/feedback.html 并输入您的意见。

文档更新

有关本文档的最新版本，请参见 《驱动程序实施指南》(http://www.novell.com/documentation/dirxmldrivers/index.html) 中的 Identity Manager Driver for SOAP 部分。

其它文档

有关 Identity Manager 的信息，请访问 Identity Manager 文档万维网站点 (http://www.novell.com/documentation/dirxml20/index.html)。有关其它 Identity Manager 驱动程序的信息，请参见 《驱动程序实施指南》(http://www.novell.com/documentation/dirxmldrivers/index.html)。

文档约定

在 Novell® 文档中，大于号 (>) 用于分隔同一步骤中的各项操作，以及交叉参照路径中的各个项目。

商标符号 （® 、 ™ 等）表示 Novell 商标。星号 (*) 表示第三方商标。

如果某个路径名的书写对某些平台需使用反斜线而对另一些平台需使用正斜线，则使用反斜线表示该路径名。要求使用前斜线的平台 （例如 Linux* 或 UNIX*）上的用户应该按软件的要求使用正斜线。

概述

1

SOAP （简单对象访问协议）是基于 XML 的协议，用于在不同应用程序和操作系统之间进行因特网通讯。

SOAP 驱动程序使用一组语言和协议在具有 Identity Manager 的 Identity Vault 和使用 HTTP 的应用程序 （如使用 SOAP 的万维网服务）之间实现身份分配和数据同步。

此驱动程序不面向特定的万维网服务，它仅是一个处理 Identity Vault 和万维网服务之间数据的 HTTP 传输的通用 Shim。对于此驱动程序，万维网服务被定义为将 XML 和 HTTP 用作传输协议的应用程序。应用程序也可以使用 SOAP 对讯息进行编码。

本节提供有关 Identity Manager Driver 1.0 for SOAP 的以下信息：

1.1 驱动程序概念

本节包含下列内容：

1.1.1 数据管理

驱动程序使用各种因特网协议和语言在 Identity Manager 和万维网服务之间交换数据。

SOAP

SOAP （简单对象访问协议）是一种基于 XML 的协议，用于交换位于 Identity Manager 中的讯息。它定义讯息交换，但不定义讯息内容。此驱动程序支持 SOAP 1.1。

SOAP 文档分为以下三个元素：

- 封套： XML 根节点。
- 标题： 提供环境信息 （如，事务 ID 和安全信息）。
- 正文： 特定于方法的信息。

SOAP 遵循 HTTP 请求 / 响应讯息模型，该模型提供 HTTP 请求中的 SOAP 请求参数和 HTTP 响应中的 SOAP 响应参数。

SPML 和 DSML

SOAP 驱动程序包括两种协议的样本配置：SPML 1.0 和 DSML 2.0。

- **SPML 1.0**：服务供应标记语言是基于 XML 的供应请求和响应协议。客户端向服务器发出 SPML 请求。该请求描述了要在给定服务点执行的操作。服务点执行必要的操作来实施所请求的服务。完成操作后，服务点向客户端返回 SPML 响应，其中详细描述了与该请求有关的所有结果或错误。

  此驱动程序支持 SPML 1.0。 SPML 与 SOAP 1.1 相联结，并使用 HTTP 和 HTTPS 1.1 作为传输协议。

- **DSML 2.0**：目录服务标记语言以 XML 文档的形式表示目录结构信息、目录查询和更新以及这些操作的结果。

DSML 与 SOAP 1.1 相联结，并使用 HTTP 和 HTTPS 1.1 作为传输协议。

有关驱动程序中包括的样本 SPML 和 DSML 配置的更多信息，请参见 “使用样本驱动程序配置”在第 11 页 。

XML

XML （可扩展标记语言）是允许在因特网上交换结构化数据的标准通用标记语言 (SGML) 的通用子集。

HTTP

HTTP 是用于在因特网或其它计算机网络上请求和传输数据的协议。此协议最适用于因特网基础结构和防火墙。

由于通常只对当前请求保持连接，因此 HTTP 是无状态请求 / 响应系统。客户端与服务器建立 TCP 连接，并向其发送请求命令，然后服务器发回响应。

1.1.2 驱动程序的工作方式

以下图表说明了 Identity Manager 和万维网服务之间的数据流：

图 ***1-1*** *SOAP* 驱动程序数据流

Identity Manager 引擎使用 XDS （XML 的专用格式）表示 Identity Vault 中的事件。 Identity Manager 将 XDS 传递给驱动程序策略 （由基本策略、 DirXML® 底稿和 XSLT 样式表组成）。

在订购者通道上，驱动程序策略将 XDS 转换为 XML （如 SOAP）。在发布者通道上，驱动程序策略将其它形式的 XML （如 SOAP）转换为 XDS。

驱动程序 Shim 接收来自驱动程序策略的 XML。驱动程序 Shim 使用 HTTP 与万维网服务通讯。通常，驱动程序 Shim 和应用程序之间的转接是序列化的 XML。

例如，假定驱动程序使用 DSML 样本配置与仅配置为订购者的 DSML 服务器进行通讯。当 Identity Vault 中发生事件时， Identity Manager 会创建一个表示该事件的 XDS 命令。 Identity Manager 将该 XDS 命令传递给驱动程序策略。

驱动程序策略用输出转换样式表转换该 XDS 命令。XSLT 样式表将 XDS 转换为包含 DSML 的 SOAP 封套。将该 SOAP 封套传递给驱动程序 Shim。驱动程序 Shim 将 SOAP 封套转换为字节数组，建立相应的 HTTP 连接并执行 HTTP POST 操作，以便将数据提交到万维网服务。

万维网服务或应用程序处理请求，并向驱动程序 Shim 返回 SOAP 响应。Shim 将响应作为字节数组接收，并在将其传递回驱动程序策略之前将其转换为 XML 文档。输入转换样式表处理响应，将其转换为报告回 Identity Manager 引擎的相应 XDS。

1.1.3 了解操作数据

驱动程序 Shim 根据嵌入命令中的 XML 元素 （在 Shim 中显示为 <operation-data>），对订购者命令进行特殊处理。 <operation-data> 元素有两种用途。首先，它可以用于将命令与其生成的响应相匹配，这对于创建关联很有用。其次，它可以用于覆盖默认的订购者通道连接特性。

将 <operation-data> 元素从某一订购者通道策略添加到命令中。驱动程序 Shim 从命令中去除 <operation-data> 元素，再将命令发送到应用程序，然后将 <operation-data> 元素恢复到得到的响应。

默认情况下，响应之后恢复 <operation-data> 元素时，会将其追加为根节点的子元素。通过向 <operation-data> 元素提供一个或多个 parent-node-*n* 特性可以改变这种情况，其中 *n* 是从 1 开始并按每个要提供的父限定词递增的数字。驱动程序 Shim 检查操作数据节点以查找 parent-node-*n* 特性。如果找到了特性，则依次尝试每个特性；如果存在已命名节点，则将节点用作响应中操作数据的父节点。

要了解 <operation-data> 元素如何与样式表结合使用，请参见 “操作数据”在第 25 页 。

1.2 驱动程序功能

驱动程序包含以下功能：

- Identity Vault 和万维网服务之间数据的 HTTP 传输
- SPML 和 DSML 的样本配置
- 自定义 HTTP 请求标题字段

  默认情况下，为订购者通道提供带有 ID 和口令的基本授权请求标题。有关更多信息，请参见 “使用驱动程序配置文件创建驱动程序对象”在第 11 页 。
- 使用 HTTPS 协议的 SSL 连接
- 订购者 HTTP 和 HTTPS 代理服务器
- 运行时策略中多个订购者连接的定义和选择
- 可充当 HTTP 或 HTTPS 侦听器以监听发布者通道上的传入连接
- 使用自定义 Java* 代码的潜在可扩展性

  有关更多信息，请参见附录 A “使用 Java 扩展”在第 35 页 。

安装驱动程序

2

本节包含以下有关安装驱动程序的信息：

2.1 驱动程序前提条件

- ❑ 下列操作系统之一：
  - 带有最新 Support Pack 的 NetWare® 6 或 6.5
  - 带有最新 Support Pack 的 Novell® Open Enterprise Server
  - 带有最新 Service Pack 的 Windows* NT*、 2000 或 2003
  - Linux Red Hat*AS、 ES 2.1 或 AS 3.0
  - SUSE® LINUX Enterprise Server 8 或 9 （包括 SP1）
  - Solaris* 8 或 9
  - AIX* 5.2L
- ❑ 带有最新 Support Pack 的 Novell eDirectory™ 8.7.3 或 Novell eDirectory 8.8
- ❑ Novell Identity Manager 3.0
- ❑ Novell iManager 2.5 或更高版本

2.2 新增功能

- 能够覆盖 SOAP 操作
- Cookie 处理

2.3 安装驱动程序

将驱动程序作为 Novell Identity Manager 3 安装程序的一部分进行安装。有关安装说明，请参考 《*Identity Manager 3.0* 安装指南》中的 《安装 Identity Manager》和 《升级》这两章。

导入驱动程序配置会创建驱动程序对象。导入配置后，即可使用 iManager 配置和管理驱动程序。有关如何配置驱动程序的说明，请参见第 3 章 “使用样本驱动程序配置”在第 11 页 。

2.4 升级

如果要升级到 Identity Manager 3.0，请按照 《*Identity Manager 3.0* 安装指南》中 《升级》一章中的说明进行操作。

使用样本驱动程序配置

3

Identity Manager Driver for SOAP 包括两个可以用作创建驱动程序对象的起始点的样本配置。

本节包含以下主题：

3.1 使用驱动程序配置文件创建驱动程序对象

SOAP 驱动程序带有两个可以用于创建驱动程序对象的配置文件：

- SOAP-SPML.xml：服务供应标记语言 (SPML) 配置文件
- SOAP-DSML.xml：目录服务标记语言 (DSML) 配置文件

有关样本文件的更多信息，请参见 “了解 SPML 配置”在第 16 页 和 “了解 DSML 配置”在第 16 页 。

3.1.1 在 Designer 中导入驱动程序配置文件

使用 Designer 可以导入 SOAP 驱动程序的驱动程序配置文件。这些文件可创建和配置驱动程序正常运行所需的对象和策略。以下说明解释了如何创建驱动程序和导入驱动程序的配置。

可以采用多种方式导入驱动程序配置文件。以下步骤只是其中的一种方式。

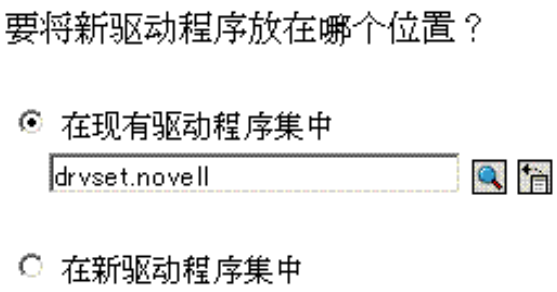
1. 在 Designer 中打开一个项目，在建模器中，右键单击 《驱动程序集》对象，然后选择 *Add Connected Application*（添加连接的应用程序）。
2. 在下拉列表中，选择 *SOAP-DSML.xml* 或 *SOAP-SPML.xml*，然后单击 《运行》。
3. 在 Perform Prompt Validation （执行提示验证）窗口中单击 《是》。
4. 填充各个字段以配置驱动程序。指定特定于所在环境的信息。有关设置的信息，请参见表 3-1 在第 13 页 和表 3-2 在第 14 页。
5. 在指定参数后，单击 《确定》以导入驱动程序。
6. 导入驱动程序后，自定义并测试驱动程序。
7. 全面测试驱动程序后，将驱动程序部署到 Identity Vault 中。请参见 《*Designer for Identity Manager 3*：管理指南》中的 《将策略部署到 Identity Vault 中》。

3.1.2 在 iManager 中导入驱动程序配置文件

SOAP 预配置文件是一个示例配置文件。在 iManager 服务器上安装 Identity Manager 万维网部件时已经安装了该文件。可以将预配置文件看作为所在环境导入、自定义或配置的模板。

1. 在 iManager 中，选择 《Identity Manager 实用程序》 > 《导入驱动程序》。

2 选择驱动程序集，然后单击 《下一步》。



如果将此驱动程序放置在新的驱动程序集中，则必须指定驱动程序集名、环境和相关的服务器。

3 选择 *SOAP DSML* 或 *SOAP SPML*，然后单击 《下一步》。

4 填写各个配置参数以配置驱动程序。有关设置的信息，请参见表 3-1 在第 13 页 和表 3-2 在第 14 页。

5 使用用户对象定义安全性等效，该对象具有驱动程序需要对服务器拥有的权限

执行此任务时，最常使用的是 Admin 用户对象。但是，可能要创建 DriversUser （举例）并向该用户指派安全性等效。无论驱动程序需要对服务器具有何种权限，DriversUser 对象都必须具有相同的安全性权限。

6 标识代表管理职能的所有对象，并将其从复制中排除。

排除在步骤 2 中指定的安全性等效对象 （如 DriversUser）。如果删除了安全性等效对象，则表明已从驱动程序中去除了权限。因此，驱动程序不能对 Identity Manager 进行更改。

7 单击 《完成》。

8 配置驱动程序的其它设置。

有关更多信息，请参见 “配置驱动程序”在第 19 页 。

3.1.3 配置参数

下表解释了在初始驱动程序配置过程中必须提供的参数。

注释：参数显示在多个屏幕上，而某些参数仅在对前一提示的回答需要更多信息以正确配置策略时才显示。

表 **3-1** *SOAP DSML* 驱动程序的配置参数

字段	说明
驱动程序名	在 Identity Manager 中指定驱动程序对象的名称。
Configure Data Flow（配置数据流）	指定要激活的驱动程序通道。 *eDirectory to DSML*（eDirectory 到 DSML）：将 Identity Vault 事件发送到应用程序。 *DSML to eDirectory*（DSML 到 eDirectory）：从应用程序接收事件。 《双向》：同时激活 eDirectory™ 和 DSML 通道。
<nds>、*<input>*、*<output> element handling*（<nds>、<input>、<output> 元素处理）	请选择以下选项之一： *Remove/Add Elements*（去除 / 添加元素）：驱动程序 Shim 去除和添加所需的 nds、输入和输出的 XML 元素。将文档发送到 Metadirectory 引擎前，将这些必需的元素从发送到应用程序的 XML 文档中去除，并添加到从应用程序接收的 XML 文档中。 对于 SOAP 驱动程序，这是优先选项。 *Pass Elements Through*（传递元素）：关闭元素处理。不必将所需的 nds、输入和输出的 XML 元素添加到 XML 文档中或从中去除。
Driver is Local/Remote（本地 / 远程驱动程序）	请选择以下选项之一： 《本地》：在拥有驱动程序集的服务器上运行驱动程序 Shim。 《远程》：使用远程装载程序在远程服务器上运行驱动程序。如果指定此选项，请单击 《下一步》，然后指定远程装载程序的配置信息。有关更多信息，请参见 *《Novell Identity Manager 3.0 管理指南》*中的 《设置连接的系统》。
URL of the remote DSML server:（远程 DSML 服务器的 URL：） （视情况而定）订购者通道字段 注释：只有在 《配置数据流》字段中选择 《eDirectory 到 DSML》或 《双向》时，才显示这些字段。	指定 《远程 DSML 服务器的 URL》和服务器监听的端口号。 例如：http://137.66.10.13:18180/soap 服务器是监听、处理有效的 DSML 请求并返回其结果的软件部件。 提示：如果将驱动程序配置为使用 SSL，则 URL 必须以 https 而不是 http 开头。
鉴定 ID （视情况而定）订购者通道字段	如果远程服务器要求 《鉴定 ID》，则在该字段中指定它。否则，请将该字段保留为空。
鉴定口令 （视情况而定）订购者通道字段	如果在上面指定了 《鉴定 ID》，则为远程服务器指定 《鉴定口令》。否则，请将这些字段保留为空。

字段	说明
Listening IP address and port（监听 IP 地址和端口） （视情况而定）发布者通道字段 注释：只有在 《配置数据流》字段中选择 《DSML 到 eDirectory》或 《双向》时，才显示这些字段。	指定安装 SOAP 驱动程序的服务器的 IP 地址和此驱动程序监听的端口号。如果服务器中只安装了一个网卡，则可以指定 127.0.0.1。选择服务器上未使用的端口号 （例如 127.0.0.1:18180）。驱动程序监听此地址的请求，处理这些请求并返回结果。
鉴定 ID （视情况而定）发布者通道字段	指定远程 DSML 服务器的 《鉴定 ID》以验证传入的请求。如果远程服务器不发送 《鉴定 ID》，可将该字段保留为空。
鉴定口令 （视情况而定）发布者通道字段	如果在上面指定了 《鉴定 ID》，则指定远程服务器的 《鉴定口令》以验证传入的请求。否则，请将这些字段保留为空。
Remote Host Name and Port（远程主机名和端口） （视情况而定）远程装载程序字段 注释：只有在 《本地 / 远程驱动程序》字段中选择 《远程》时，才显示这些字段。	输入运行远程装载程序服务器的服务器主机名或 IP 地址以及端口。 示例：137.66.10.13:8090 端口 8090 是远程装载程序服务监听的默认端口。
驱动程序口令 （视情况而定）远程装载程序字段	驱动程序口令由远程装载程序用于将自身鉴定到 Identity Manager 服务器。它必须与在远程装载程序服务器的 《驱动程序对象口令》中指定的口令相同。
远程口令 （视情况而定）远程装载程序字段	远程口令用于控制对远程装载程序的访问。它必须与指定的远程装载程序服务器的远程装载程序口令相同。

表 ***3-2*** 配置 *SOAP SPML* 驱动程序的参数

字段	说明
驱动程序名	在 Identity Manager 中指定驱动程序对象的名称。
配置数据流	指定要激活的驱动程序通道。 eDirectory 到 SPML：将 Identity Vault 事件发送到应用程序。 SPML 到 eDirectory：从应用程序接收事件。 双向：同时激活 eDirectory 和 SPML 通道。

字段	说明
<nds>、 <input>、 <output> 元素处理	请选择以下选项之一： 去除 / 添加元素：驱动程序 Shim 去除和添加所需的 nds、输入和输出的 XML 元素。将文档发送到 Metadirectory (Identity Manager) 引擎前，将这些必需的元素从发送到应用程序的 XML 文档中去除，并添加到从应用程序接收的 XML 文档中。 对于 SOAP 驱动程序，这是优先选项。 传递元素：关闭元素处理。不必将所需的 nds、输入和输出的 XML 元素添加到 XML 文档中或从中去除。
本地 / 远程驱动程序	请选择以下选项之一： 《本地》：在拥有驱动程序集的服务器上运行驱动程序 Shim。 《远程》：使用远程装载程序在远程服务器上运行驱动程序。如果指定此选项，请单击 《下一步》，然后指定远程装载程序的配置信息。有关更多信息，请参见 *《Novell Identity Manager 3.0 管理指南》*中的 《设置连接的系统》。
URL of the remote SPML Provisioning Service Point:（远程 SPML 供应服务点的 URL：） （视情况而定）订购者通道字段 注释：只有在 《配置数据流》字段中选择 《eDirectory 到 SPML》或 《双向》时，才显示这些字段。	指定远程 SPML 供应服务点 (PSP) 的 URL。 例如：http://137.66.10.13:18180/soap PSP 是监听、处理有效的 SPML 请求并返回其结果的软件部件。 提示：如果将驱动程序配置为使用 SSL，则 URL 必须以 https 而不是 http 开头。
鉴定 ID （视情况而定）订购者通道字段	如果远程 SPML PSP 要求鉴定 ID，则指定远程 SPML PSP 的鉴定 ID。否则，请将该字段保留为空。
鉴定口令 （视情况而定）订购者通道字段	如果在上面指定了鉴定 ID，则指定远程 SPML PSP 的鉴定口令以验证进入的请求。否则，请将该字段保留为空。
监听 IP 地址和端口 （视情况而定）发布者通道字段 注释：只有在 《配置数据流》字段中选择 《SPML 到 eDirectory》或 《双向》时，才显示这些字段。	指定安装驱动程序的服务器的 IP 地址和此驱动程序作为 PSP 监听的端口号。如果服务器中只安装了一个网卡，则可以指定 127.0.0.1。选择服务器上未使用的端口号。 示例：127.0.0.1:18180 驱动程序监听此地址的 SPML 请求，处理这些请求并返回结果。
鉴定 ID （视情况而定）发布者通道字段	指定鉴定 ID 以验证传入的 SPML 请求。
鉴定口令 （视情况而定）发布者通道字段	指定鉴定口令以验证传入的 SPML 请求。

字段	说明
远程主机名和端口 （视情况而定）远程装载程序字段 注释：只有在 《本地 / 远程驱动程序》字段中选择 《远程》时，才显示这些字段。	输入运行远程装载程序服务器的服务器主机名或 IP 地址以及端口。 示例：137.66.10.13:8090 端口 8090 是远程装载程序服务监听的默认端口。
驱动程序口令 （视情况而定）远程装载程序字段	驱动程序口令由远程装载程序用于将自身鉴定到 Identity Manager 服务器。它必须与在远程装载程序服务器的 《驱动程序对象口令》中指定的口令相同。
远程口令 （视情况而定）远程装载程序字段	远程口令用于控制对远程装载程序的访问。它必须与指定的远程装载程序服务器的远程装载程序口令相同。

3.2 了解 DSML 配置

样本 DSML 配置使用 DSML 2.0 并与 SOAP 1.1 相联结，使用 HTTP 或 HTTPS 1.1 作为传输协议。所有数据转换和处理都在策略和样式表中完成。

样本 DSML 导入文件执行以下操作：

- 显示用于与 Identity Vault DSML 实施相配对的简单配置。
- 在策略中提供 XDS 到 DSML 和 DSML 到 XDS 的转换。
- 处理用户、组和组织单元。

  通过自定义策略和样式表可以处理其它对象。
- 支持字符串、结构化和判别名 (DN) 特性类型。

  下面两个示例说明如何处理其它数据类型的特性。《邮递地址》特性显示如何处理结构化特性。《成员》特性显示如何处理 DN 特性。通过自定义策略和样式表可以处理其它数据类型的特性。
- 处理查询操作的子集。

  通过自定义策略和样式表可以处理特定的查询操作。
- 支持口令集操作。

  通过自定义策略和样式表可能实现口令同步。
- 订购者通道对关联键使用目标 DN。
- 发布者通道对关联键使用应用程序提供的 DN。

3.3 了解 SPML 配置

样本 SPML 配置使用 SPML 1.0 并与 SOAP 1.1 相联结，使用 HTTP 或 HTTPS 1.1 作为传输协议。所有数据转换和处理都在策略和样式表中完成。

样本 SPML 导入文件执行以下操作：

- 提供通用 SPML 功能。

  导入文件不与特定的 SPML 应用程序相配对。

- 在策略中提供 XDS 到 SPML 和 SPML 到 XDS 的转换。
- 处理用户、组和组织单元

  通过自定义策略和样式表可以处理其它对象。
- 对每个特性只处理一个值。

  通过自定义策略和样式表可以处理特性的多个值。
- 处理查询操作的子集。

  配置将所有查询都作为 SPML scope = "subtree" 进行处理，并使用项和从属范围概念。通过自定义策略和样式表可以处理特定的查询操作。
- 支持字符串、结构化和判别名 (DN) 特性类型。
- 支持口令集操作。

  通过自定义策略和样式表可能实现口令同步。
- 处理 execution=synchronous 和 processing=sequential 的单个 （非批处理）操作。

  通过自定义策略和样式表可以支持批处理请求。
- 不处理 <addResponse><attributes> 或 <modifyResponse><modifications>。
- 订购者通道对关联键使用应用程序返回的标识符值。
- 发布者通道对关联键使用 DN，并以标识符值的形式返回关联键。

3.4 处理未关联对象的发布者通道上的修改事件

HTTP/SOAP 驱动程序的发布者通道有某些限制，只允许其监听更改事件。无法查询其它信息或巡回检测 HTTP/SOAP 源。因此，在未关联对象 （或并非相同驱动程序实例创建的对象）的发布者通道上收到的修改事件几乎总是失败 （返回一个错误）。发生这种情况的原因是：驱动程序和 Metadirectory 引擎在不能将查询发送到 HTTP/SOAP 源的情况下，无法成功地将未关联修改事件更改为添加命令。因为 SOAP 驱动程序没有向后查询源的机制，因此该驱动程序将返回一个表示未实施查询的错误。

没有对此限制的全面解决方案。因此，发生此情况时， DSML 和 SPML 的样本配置都将返回一个错误。在特定的驱动程序部署中，如果必须将关联应用于对象，并且允许该新关联的对象中有不一致的信息，则可以通过在修改事件中设置目标 DN 并创建与自己的设置关联的事件，在策略中解决这一问题。这样，即使先前没有关联，也允许对现有对象进行修改。

配置驱动程序

4

使用样本文件之一创建驱动程序对象后，需要配置 Identity Manager Driver for SOAP。本节包含以下有关配置驱动程序的信息：

4.1 配置驱动程序设置

1. 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。
2. 找到包含 SOAP 驱动程序的驱动程序集，然后单击驱动程序图标。
3. 在 《Identity Manager Driver 概述》中，单击 SOAP 驱动程序对象，将显示驱动程序配置。
4. 指定驱动程序模块信息：
   - **4a** 在 《驱动程序模块》部分中，选择 *Java*。
   - **4b** 在 《名称》字段中，指定以下 SOAP 驱动程序 Java 类名：
     com.novell.nds.dirxml.driver.soap.SOAPDriver
5. 指定驱动程序对象口令信息：
   - **5a** 滚动到 《驱动程序对象口令》部分，然后单击 《设置口令》。
   - **5b** 在字段中，输入两次驱动程序对象口令。
6. 指定鉴定信息：
   - **6a** 滚动到 《鉴定》部分。
   - **6b** 指定 《鉴定 ID》。
   - **6c** 指定 《鉴定环境》。
   - **6d** 指定 《远程装载程序连接参数》。
   - **6e** 指定 《驱动程序超速缓存限制》 (KB)。
   - **6f** 单击 《设置口令》以指定 《应用程序口令》。
   - **6g** 输入两次 《应用程序口令》。
7. 指定启动信息：
   - **7a** 滚动到 《启动》部分。
   - **7b** 请选择以下选项之一：
     - 《自动启动》：启动 eDirectory™ 时将自动启动驱动程序。
     - 《手工》：必须使用 iManager 手工启动驱动程序。
     - 《禁用》：驱动程序将不运行。
8. 指定以下驱动程序设置：

部分	字段	说明
驱动程序设置	*<nds>、 <input>、 <output> 元素处理*	如果希望驱动程序 Shim 去除和添加所需的 XML 元素 <nds>、 <input> 和 <output>，请指定 《去除 / 添加元素》。 将文档发送到 Metadirectory 引擎前，将这些必需的元素从发送到应用程序的 XML 文档中去除，并添加到从应用程序接收的 XML 文档中。否则，指定 《传递元素》以关闭此元素处理。
	自定义 Java 扩展	如果已开发出自定义 Java 类来扩展驱动程序 Shim 的功能，则指定 《显示》。否则，指定 《隐藏》。 有关更多信息，请参见附录 A “使用 Java 扩展”在第 35 页 。
订购者设置	*远程 DSML 服务器的 URL*	输入远程服务器的 URL 和服务器监听的端口号。 除非配置了 SSL 设置 （此时应以 https:// 开头），否则 URL 应以 http:// 开头，并使用 DNS 主机名而不是 IP 地址。
	（视情况而定）《鉴定 ID》	如果远程服务器要求鉴定 ID，请在字段中输入 ID。否则，请将该字段保留为空。
	（视情况而定）《鉴定口令》和《重新输入鉴定口令》	如果在上面输入了 《鉴定 ID》，请为远程服务器输入鉴定口令。否则，请将该字段保留为空。
	去除现有口令	单击该框以去除现有口令。然后在 《鉴定口令》和 《重新输入鉴定口令》字段中指定新口令。 只有选择该框才能更改口令。
	Truststore File （可信存储区文件）	指定包含可信证书 （当远程服务器配置为提供服务器鉴定时使用）的密钥存储区文件的名称和路径。例如：c:\security\truststore。如果不使用服务器鉴定，请将该字段保留为空。
	设置相互鉴定参数	指定 《显示》可设置相互鉴定信息。指定 《隐藏》，将不使用相互鉴定。
	Proxy Host and Port （代理主机和端口）	在使用代理主机和端口时，请指定主机地址和主机端口。例如：192.10.1.3:18180。 或者，如果不使用代理主机和端口，请将该字段保留为空。

部分	字段	说明
	处理 HTTP 会话 Cookie	某些 HTTP 应用程序设置 Cookie，并期望在将来请求时它们依然存在。如果希望驱动程序跟踪会话 Cookie，请选择 *Handle Cookies*（处理 Cookie）。 在驱动程序停止之前， Cookie 将一直保留。 如果 HTTP 应用程序不需要 Cookie，请选择 *Ignore Cookies*（忽略 Cookie）。
	自定义 HTTP 请求标题字段	选择 《显示》以启用自定义的标题字段或选择 《隐藏》以禁用该功能。下列每个字段取决于选择 《用户》还是 《忽略》。 ◆ 《授权》：如果选择 《使用》，请在相应字段中指定密钥和值。如果在 《订购者设置》中输入鉴定 ID 和口令，将自动使用此标题。 ◆ *Context Type*（环境类型）：如果选择 《使用》，请在相应字段中指定密钥和值。 ◆ *SOAPAction*：如果选择 《使用》，请在相应字段中指定密钥和值。 ◆ *Optional Request Header*（可选请求标题）：如果选择 《使用》，请在相应字段中指定密钥和值。可指定最多三个可选请求标题。
发布者设置	*监听 IP 地址和端口*	指定安装 SOAP 驱动程序的服务器的 IP 地址和此驱动程序监听的端口号。 如果导入一个样本配置文件，该字段将包含在向导中指定的 IP 地址和端口。
	（视情况而定）《鉴定 ID》	指定远程服务器的 《鉴定 ID》以验证传入的请求。如果远程服务器不发送 《鉴定 ID》，可将该字段保留为空。 如果导入一个样本配置文件，该字段将包含在向导中指定的 IP 地址和端口。
	（视情况而定）《鉴定口令》和《重新输入鉴定口令》	如果在上面输入了 《鉴定 ID》，请指定远程服务器的鉴定口令以验证传入的请求。否则，请将这些字段保留为空。
	去除现有口令	单击该框以去除现有口令，然后在 《鉴定口令》和 《重新输入鉴定口令》字段中指定新口令。 只有选择该框才能更改口令。

部分	字段	说明
	KMO name （KMO 名称）	指定要在 eDirectory 中使用的 《KMO 名称》。 当此服务器配置为接受 HTTPS 连接时，此名称将成为 eDirectory 中的 KMO 名称。《KMO 名称》是 RDN 中 《-》（短线）前的名称。 在使用密钥存储区文件 （请参见下面的密钥存储区文件）或未使用 HTTPS 连接时，可将该字段保留为空。
	密钥存储区文件	指定密钥存储区文件的密钥存储区名称和路径。如果服务器配置为接受 HTTPS 连接，则可使用此文件。 在使用 KMO 名称 （请参见下面的 KMO 名称）或未使用 HTTPS 连接时，可将该字段保留为空。
	密钥存储区口令	如果此服务器配置为接受 HTTPS 连接，则指定上面指定的密钥存储区文件使用的密钥存储区文件口令。 在使用 KMO 名称或未使用 HTTPS 连接时，可将该字段保留为空。
	Server key alias （服务器密钥别名）	如果此服务器配置为接受 HTTPS 连接，则可指定服务器密钥别名。 在使用 KMO 名称或未使用 HTTPS 连接时，可将该字段保留为空。
	Server key password （服务器密钥口令）	如果此服务器配置为接受 HTTPS 连接，则为密钥别名口令 （而不是密钥存储区口令）。在使用 《KMO 名称》（参见上文）或未使用 HTTPS 连接时，可将该字段保留为空。
	Heartbeat Interval in Seconds （心跳间隔 （秒））	指定心跳间隔 （秒）。 将该字段保留为空可关闭心跳。有关心跳的更多信息，请参见 *《Novell Identity Manager 3.0 管理指南》* 中的 《添加驱动程序心跳》。

9 单击 《应用》，然后单击 《确定》。

4.2 配置订购者设置

1 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。

2 找到包含 SOAP 驱动程序的驱动程序集，然后单击驱动程序的图标以显示 《Identity Manager 驱动程序概述》 页。

3 在 《Identity Manager 驱动程序概述》 页上，再次单击驱动程序的图标，然后滚动到 《订购者设置》。

4 按步骤 8 在第 19 页 中所述指定订购者设置。

5 单击 《应用》，然后单击 《确定》。

4.2.1 配置订购者以建立与远程万维网服务的 HTTPS 连接

如果访问的远程万维网服务允许 HTTPS 连接，则可以配置订购者以利用此功能。您需要包含由证书授权者 （签署了服务器证书）发布的证书的可信存储区。请参见 “配置发布者以接收 HTTPS 连接”在第 24 页 中的示例。

使用 Java 的密钥工具将此证书导入可信存储区。有关密钥工具的更多信息，请参见密钥工具 – 密钥和证书管理工具 (http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html)。

1 通过在命令提示符处输入下面的命令，将证书导入可信存储区或创建新的可信存储区：

```
keytool -import -file name_of_cert_file -trustcacerts -noprompt -
keystore filename -storepass password
```

例如：

```
keytool -import -file tree_ca_root.b64 -trustcacerts -noprompt -
keystore dirxml.keystore -storepass novell
```

2 配置订购者以使用在步骤 1 中创建的可信存储区。

2a 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。

2b 找到包含 SOAP 驱动程序的驱动程序集，然后单击驱动程序的图标以显示 《Identity Manager 驱动程序概述》页。

2c 在 《Identity Manager 驱动程序概述》页上，再次单击驱动程序的图标，然后滚动到 《订购者设置》。

2d 在 《可信存储区文件》设置中，指定在步骤 1 中创建的可信存储区的路径。

3 单击 《应用》，然后单击 《确定》。

4.2.2 配置订购者以使用代理

可以配置订购者以使用 HTTP 或 HTTPS 代理服务器。

1 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。

2 找到包含 SOAP 驱动程序的驱动程序集，然后单击驱动程序的图标以显示 《Identity Manager 驱动程序概述》页。

3 在 《Identity Manager 驱动程序概述》页上，再次单击驱动程序的图标，然后滚动到 《订购者设置》。

4 在 《代理主机和端口》设置中，使用下面的格式指定代理的主机和端口：

host:port

5 单击 《应用》，然后单击 《确定》。

4.3 配置发布者设置

1 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。

2 找到包含 SOAP 驱动程序的驱动程序集，然后单击驱动程序的图标以显示 《Identity Manager 驱动程序概述》 页。

3 在 《Identity Manager 驱动程序概述》 页上，再次单击驱动程序的图标，然后滚动到 《发布者设置》。

4 按步骤 8 在第 19 页 中所述指定订购者设置。

5 单击 《应用》，然后单击 《确定》。

4.3.1 配置发布者以接收 HTTPS 连接

1 在 iManager 中创建服务器证书。

1a 单击 《Novell 证书服务器》 > *Create Server Certificate* （创建服务器证书）。

1b 浏览并选择安装了 SOAP 驱动程序的服务器对象。

1c 指定证书昵称。

1d 选择 《标准》创建方法，然后单击 《下一步》。

1e 单击 《完成》，然后单击 《关闭》。

2 在 eDirectory 中导出来自证书授权者的自签名证书。

2a 单击 《eDirectory 管理》 > 《修改对象》。

2b 选择树的证书授权者对象，然后单击 《确定》。

它通常可以在安全树枝中找到，其名称类似于 *TREENAME CA.Security*。

2c 单击 《证书》 > *Self Signed Certificate* （自签名证书）。

2d 单击 《导出》。

2e 当系统询问您是否要将私用密钥和证书一起导出时，请单击 《否》，然后单击 《下一步》。

2f 根据要访问万维网服务的客户端，为证书选择 *File in binary DER format* （二进制 DER 格式的文件）或 *File in Base64 format* （Base64 格式的文件），然后单击 《下一步》。

如果客户端使用基于 Java 的密钥存储区或可信存储区，那么可以选择任一格式。

2g 单击 *Save the exported certificate to a file* （将导出的证书保存到文件）。

2h 单击 《保存》并浏览到计算机上的某个已知位置。

2i 单击 《保存》，然后单击 《关闭》。

3 将自签名证书导入客户端的可信存储区。

导入证书的具体步骤取决于与发布者通道的 HTTPS 监听器连接的客户端。如果客户端使用标准 Java 密钥存储区，那么可以执行下列步骤来创建密钥存储区：

3a 使用任何 Java JDK* 附带的可执行密钥工具。

有关密钥工具的更多信息，请参见密钥工具 – 密钥和证书管理工具 (http://java.sun.com/j2se/1.4.2/docs/tooldocs/windows/keytool.html)。

3b 在命令提示符处键入下面的命令：

```
keytool -import -file name_of_cert_file -trustcacerts -noprompt
-keystore filename -storepass password
```

例如：

keytool -import -file tree_ca_root.b64 -trustcacerts -noprompt -keystore dirxml.keystore -storepass novell

4 配置发布者以使用在步骤 1 中创建的服务器证书。

4a 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。

4b 找到包含 SOAP 驱动程序的驱动程序集，然后单击驱动程序的图标以显示《Identity Manager 驱动程序概述》页。

4c 在 《Identity Manager 驱动程序概述》页上，再次单击驱动程序的图标，然后滚动到 《发布者设置》。

4d 在 《KMO 名称》设置中，指定步骤 1 中使用的证书昵称。

5 单击 《应用》，然后单击 《确定》。

4.4 创建 XSLT 样式表

要使 SOAP 驱动程序能够用于除 DSML 或 SPML 默认配置以外的任何设置，需要创建 XSLT 样式表。特定于应用程序的协议处理在输入转换和输出转换样式表中完成。

有关编写样式表以处理其它文档类型的详细信息，请参考该驱动程序附带的样本样式表。有关样式表的更多信息，请参见 《策略构建器和驱动程序自定义指南》中的 《使用 XSLT 样式表定义策略》。

4.5 操作数据

驱动程序 Shim 对基于 <operation-data> 元素的订购者命令执行特殊处理。在订购者通道，<operation-data> 元素可以添加到命令中以用于两个目的。

1. 指定期望在命令结果中包含的 XML 数据。这样，便可以将命令与其生成的响应相匹配，这对于创建关联非常有用。
2. 逐个命令覆盖默认订购者选项。

如第 1 章 “概述”在第 5 页 中所讨论的， <operation-data> 元素可以添加到订购者通道的某个策略的命令中。驱动程序 Shim 将在操作数据发送到应用程序之前将其从命令中去除，并将 <operation-data> 元素 （和所有子元素）恢复到结果响应中。如果需要，规则和样式表可以访问结果中的操作数据元素。

4.5.1 使用操作数据指定在结果中返回 XML

SOAP 驱动程序的样本配置使用 <operation-data> 元素跟踪命令的标识信息，从而可以识别结果并正确地指派关联。检查这些样本以获得有关如何使用 <operation-data> 元素的细节。

当响应之后恢复 <operation-data> 元素时，该元素将被追加为根节点的子元素。通过向 <operation-data> 元素提供一个或多个 parent-node-*n* 特性可以改变这种情况，其中 *n* 是从 1 开始并按每个要提供的父限定词递增的数字。驱动程序 Shim 查找 parent-node-*n* 特性。找到后，将检查该特性以确定是否存在命名节点。如果找到节点，那么将其用作响应中 <operation-data> 元素的父节点。

4.5.2 使用操作数据覆盖默认订购者选项

可以采用两种方式覆盖默认的订购者命令选项。

1. 在配置中创建多个订购者选项集 （称为连接），并使用 <operation-data> 元素指定用于当前命令的连接集。
2. 指定用于覆盖当前命令的特定选项，如 URL、方法或 SOAP 操作。

创建和使用多个订购者选项集 （连接）

使用 <operation-data> 元素覆盖默认订购者连接参数：

1 编辑驱动程序配置的订购者设置部分。

2 使用 iManager 的 XML 编辑功能，查找以短线和数字 1 结尾 （如 subURL-1）的每个订购者设置，然后复制它，并使数字递增。

例如：subURL-2

3 将新设置的值编辑为希望用于第二个连接的值。

可以采用这种方式配置任意数量的连接，只要使用的数字不间断依次递增即可。

4 将特性添加到称为 connection 的 <operation-data> 元素，然后为其提供要使用的连接编号的值。

例如：

```
<operation-data connection="2"> ...(other operation-data elements)
</operation-data>
```

覆盖单个订购者选项

无需使用连接概念来覆盖多个订购者选项，只需直接使用 <operation-data> 元素的特性便可仅覆盖 URL、HTTP 方法或 SOAP 操作值。下表列出了可以使用的特性及其要覆盖的订购者选项。

表 ***4-1*** 用于覆盖订购者选项的特性

<operation-data> 特性	被覆盖的订购者选项	说明
url	subURL-1	这是万维网服务或 HTTP 应用程序的 URL （或 URI）。如果应用程序具有一个用于添加用户的万维网服务和另一个用于删除用户的万维网服务，覆盖 URL 将会非常有用。
方法	subHttpMethod-1	默认情况下为 POST，但也可以根据需要设置为 RFC 2616 第 9 部分中定义的其它方法。
SOAP 操作	带有密钥 《SOAPAction》 的 HTTP 请求标题字段	对 DSML 和 SPML 样本而言，该值始终是 #batchRequest。但是，根据所使用的命令，某些万维网服务需要更改该值。

示例：

```
<operation-data url="http://137.66.10.13:18180/soap"> ...(other
operation-data elements if required) </operation-data>

<operation-data method="GET"> ...(other operation-data elements if
required) </operation-data>

<operation-data soap-action="addUser"> ...(other operation-data
elements if required) </operation-data>
```

使用驱动程序

5

完成驱动程序安装并导入样本配置文件后，必须完成下列任务：

5.1 启动驱动程序

1. 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。
2. 浏览并选择驱动程序所在的驱动程序集，然后单击 《搜索》。
3. 单击 SOAP 驱动程序图标的右上角，然后单击 《启动驱动程序》。

要进一步配置启动选项，请参见 “配置驱动程序设置”在第 19 页 。

5.2 迁移和重新同步数据

当数据更改时， Identity Manager 会同步数据。如果要立即同步所有数据，可以选择以下选项：

- 从 **Identity Vault** 中迁移数据： 使您能够选择希望从 Identity Vault 迁移至应用程序的树枝或对象。在迁移对象时， Identity Manager 引擎将所有匹配、布局和创建策略以及订购者过滤器应用到对象。
- 将数据迁移到 **Identity Vault** 中： 假定可以查询远程应用程序 （通常为万维网服务）以查找符合发布者过滤器准则的项目。但是，由于 SOAP 驱动程序的一般特性，查询万维网服务的方法 （如果有）不为驱动程序 Shim 所知。因此，该功能通常不适用于 SOAP 驱动程序。
- 同步： Identity Manager 引擎在订购者类过滤器中查找并处理这些类的所有对象。合并关联对象，将不关联的对象作为添加事件进行处理。

使用上述选项之一：

1. 在 iManager 中，单击 *Identity Manager* > 《Identity Manager 概述》。
2. 浏览并选择驱动程序所在的驱动程序集，然后单击 《搜索》。
3. 单击驱动程序图标。
4. 单击相应的迁移按钮。

5.3 激活驱动程序

必须在驱动程序安装完成后的 90 天内激活它，否则驱动程序将被停用。

有关激活的信息，请参考 《*Novell Identity Manager 3.0* 管理指南》中的 《激活 Novell Identity Manager 产品》。

排查驱动程序错误

6

本节包含有关错误讯息的以下信息：

6.1 驱动程序 Shim 错误

下表列出了可能在内核驱动程序 Shim 中出现的错误。包含数字代码的错误讯息有多种，具体取决于应用程序或万维网服务。

错误讯息	级别	说明
307 Temporary Redirect （307 临时重定向）	重试	订购者通道试图将数据发送到应用程序或万维网服务，但收到一个 《307 临时重定向》响应。 订购者等待一段时间 （通常为 30 秒）后重试。
408 Request Timeout （408 请求超时）	重试	订购者通道试图将数据发送到应用程序或万维网服务，但收到一个 《408 请求超时》响应。 订购者等待一段时间 （通常为 30 秒）后重试。
503 Service Unavailable （503 服务不可用）	重试	订购者通道试图将数据发送到应用程序或万维网服务，但收到一个 《503 服务不可用》响应。 订购者等待一段时间 （通常为 30 秒）后重试。
504 Gateway Timeout （504 网关超时）	重试	订购者通道试图将数据发送到应用程序或万维网服务，但收到一个 《504 网关超时》响应。 订购者等待一段时间 （通常为 30 秒）后重试。
Various numeric error codes not listed above. （以上未列出的各种数字错误代码。）	错误	HTTP 服务器 （如订购者通道可能与其进行通讯的服务器）将返回数字值和一条简短的说明讯息以表示请求的状态。 介于 200-299 之间的数字表明操作成功，因此不会生成错误讯息。 上面列出的数字 （307、 408、 503 和 504）表示临时情况，因此需要重试请求。 如果出现其它数字错误代码，将出现由该代码和 HTTP 服务器提供的讯息构成的错误讯息。大部分情况下，驱动程序将继续运行，不会重试导致该错误的命令。
Problem communicating with HTTP server.Make sure server is running and accepting requests. （与 HTTP 服务器通讯时出现问题。确保服务器正在运行并接受请求。）	重试	订购者通道与 HTTP 服务器通讯或试图与其通讯时收到 IOException。 您收到此错误的原因可能是：服务器没有运行、超载、由于防火墙或其它限制无法访问，或者是订购者配置中提供的 URL 不正确。 将在稍后重试引起此错误的命令。

错误讯息	级别	说明
The HTTP/SOAP driver doesn 諚 return any application schema by default.（默认情况下， HTTP/SOAP 驱动程序不返回任何应用程序纲要。） If there is an application-specific schema you want the shim to report, you can write your own Java class that implements the SchemaReporter interface and then configure the driver to load your class as a Java extension.（如果希望 Shim 报告一个特定于应用程序的纲要，可以编写自己的 Java 类来实施 SchemaReporter 接口，然后配置驱动程序，以 Java 扩展的形式装载类。）	警告	Metadirectory 引擎调用驱动程序的 DriverShim.getSchema() 方法，而驱动程序未通过 SchemaReporter 自定义得到扩展。 驱动程序继续运行。
Subscriber.execute() was called but the Subscriber was not configured correctly.The command was ignored.（调用 Subscriber.execute()，但订购者配置不正确。命令被忽略。） You should either configure the Subscriber or clear the Subscriber 謂 filter so it doesn 諚 receive commands.（应该配置订购者或清除订购者的过滤器，这样就不会接收命令。）	警告	驱动程序的订购者通道初始化不正确。最可能的原因是驱动程序配置的格式不正确。 驱动程序继续运行，但是每次订购者通道收到事件时，它都会显示此讯息。
pubHostPort must be in the form host:port （pubHostPort 必须采用 host:port 格式）	致命错误	发布者通道配置发生错误。 查看发布者通道参数以校验是否提供了有效的主机和有效的端口号。
MalformedURLException	致命错误	订购者通道参数中提供的 URL 采用了无效的 URL 格式。
Multiple Exceptions （多个异常）	致命错误	HTTP 监听器不能正确初始化时，将在跟踪中显示此讯息。出现此讯息的原因很多。检查发布者设置以确保指定了未使用的端口而且其它发布者设置正确。
HTTPS Hostname Wrong:Should Be . . .（HTTPS 主机名错误：应该是 . . .）	重试	当在订购者通道上 SSL 握手失败时，将显示此讯息。这表明服务器证书显示的对象与 HTTPS URL 中提供的 IP 地址或主机名不匹配。 在 URL 中使用 DNS 主机名而非 IP 地址。

6.2 Java 自定义错误

下表列出了自定义 Java 扩展中可能出现的错误。

讯息	级别	说明
SchemaReporter init problem:*xtension-specific message*（SchemaReporter 初始化问题：*xtension-specific message*）	致命错误	SchemaReporter Java 自定义存在初始化问题。 驱动程序关闭。
Extension (custom code) init problem:*extension specific message*（扩展 （自定义代码）初始化问题：*extension specific message*）	致命错误	以下某个 Java 扩展初始化失败： ◆ SubscriberTransport ◆ PublisherTransport ◆ DocumentModifiers ◆ ByteArrayModifiers 驱动程序关闭。
各种其它错误	不确定	为 Java 扩展提供的接口在跟踪屏幕上返回错误讯息，有时会将讯息返回至 Identity Manager 引擎。 有时很难将此种类型的错误与内核驱动程序 Shim 中发生的其它错误加以区分。如果遇到此表中未列出的错误，且正在使用 Java 扩展，请向提供扩展的人员核实是否有该特定扩展的错误代码列表。

使用 Java 扩展

A

使用 Java 可以扩展 Identity Manager Driver for SOAP 的功能。使用 Java 接口定义的 API，可以创建自己的自定义 Java 类，这些类可以访问通过订购者通道和发布者通道的数据。这些类可以读取和解释数据，也可修改数据。有些 Java 接口还定义为用您自己的自定义订购者或发布者替换默认的订购者或发布者 （使用 HTTP）。

本节包含有关使用 Java 扩展的以下信息：

A.1 概述

如果用于 Identity Manager Driver for SOAP 的应用程序使用非 XML 数据，那么可以创建 Java 扩展将非 XML 数据转换为 XML 数据。或者，也可以更改各种协议，包括 XML 和 HTTP。例如，可以替换默认的 HTTP。这些 Java 扩展可以用于操作数据，而且必须用于将非 XML 数据转换为 XML 数据。如下面的图表所示，共有十一点可以进行功能扩展：

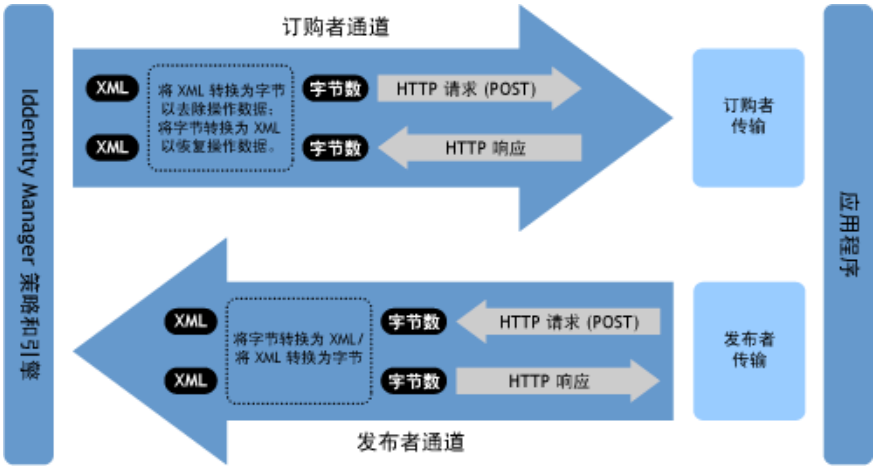
- 订购者通道中有四点
- 发布者通道中有四点
- 两点用于指定传输
- 一点用于报告应用程序纲要

SOAP 驱动程序的设计注重灵活性和可扩展性。它为想要扩展或修改驱动程序功能的 Java 程序员提供了实现这一目标的编程接口。在需要执行无法在策略或样式表中完成的转换时，才能使用这些接口。

Javadoc (http://www.novell.com/documentation/beta/dirxmldrivers/javadoc/api/index.html) 介绍了这些接口。

共有五个 Java 接口可用于扩展或自定义驱动程序行为。它们是 DocumentModifiers、ByteArrayModifiers、 PublisherTransport、 SubscriberTransport 和 SchemaReporter。

图 ***A-1*** 使用 *Java* 扩展功能的方法



DocumentModifiers 和 ByteArrayModifiers 具备类似的功能，所以可能只需要使用其中一个即可。它们都可以用于访问 （必要时可修改）经过驱动程序 Shim 的命令和事件。DocumentModifiers 可以访问 XML DOM 文档格式的数据。ByteArrayModifiers 可以访问相同的数据，但串行化为字节数组。

PublisherTransport 接口允许用其它监听器替换驱动程序在发布者通道上使用的默认 HTTP 监听器。 PublisherTransport 实施可以由事件引起，也可以按指定间隔巡回检测。

如果想要用其它连接替换驱动程序在订购者通道上使用的 HTTP 或 HTTPS 连接，那么需要实施一个 SubscriberTransport 接口。

如果能够以编程的方式来确定远程万维网服务使用的类和特性，那么可以使用最后一个接口 SchemaReporter。使用该接口的优点在于：如果可以动态确定纲要，则创建纲要映射规则就会更容易一些。

A.2 创建和配置 Java 扩展

以 Novell 的开发者下载万维网站点 (http://developer.novell.com/ndk/downloadaz.htm) 中提供的样本代码和 SOAP Driver Javadoc 为指南，编写符合您的类的 Java 代码。在 A-Z 列表中，搜索 SOAP 驱动程序。应该根据您的环境和组织的便利，用任何 Java 包和类名称来命名类。

例如，如果您自己编写的类实施了 DocumentModifiers 接口，并在名为 com.novell.idm 的包中将其命名为 *MyDocumentModifiers*，则需要执行以下步骤以对类进行编译、压缩为 jar 格式并进行部署：

1. 准备环境。

   确保您的计算机上安装了最新的 Java 开发工具 (JDK)。如果需要下载该开发工具，请访问 Java 万维网站点 (http://java.sun.com/)。

2. 在按照包命名定义的正确目录结构中收集源代码。

   在上面的例子中，您具有一个包含 novell 目录 （包含 idm 目录）的 com 目录。在 idm 目录中，您具有一个名为 MyDocumentModifiers.java 的源文件。

3 确保具有编译类所需的 jar 文件。

至少需要 SOAPUtil.jar。如果在类中使用 XML 文档，还需要 nxsl.jar。

4 将所需 jar 文件的拷贝放在方便的位置 （如 com 目录外编译目录的根中），然后访问系统命令提示符或壳层提示符，以该位置作为当前目录。

5 通过输入下列命令之一编译类：

- 对于 Windows 系统：javac -classpath SOAPUtil.jar;nxsl.jar com\novell\idm*.java
- 对于 Linux 或 UNIX 系统：javac -classpath SOAPUtil.jar:nxsl.jar com/novell/idm/*.java

6 通过输入下列命令之一创建包含您的类的 Java 档案文件：

- 对于 Windows 系统：jar cvf mydriverextensions.jar com\novell\idm*.class
- 对于 Linux 或 UNIX 系统：jar cvf mydriverextensions.jar com/novell/idm/*.class

7 将在步骤 6 中创建的 jar 文件放在包含 SOAPShim.jar 的同一目录中。

在 Windows 系统中，该目录通常为 C:\Novell\NDS\lib。

8 在 iManager 中，编辑驱动程序设置。

8a 在 Custom Java Extensions （自定义 Java 扩展）旁边，选择 《显示》。

8b 在 Document Handling （文档处理）旁边，选择 《已实施》。

8c 指定 *com.novell.idm.MyDocumentModifiers* 作为类的值，指定任何字符串作为初始化参数的值。

由于初始化参数是传递给类的初始化方法的字符串，因此可在此处输入要在类初始化过程中使用的任何信息。

9 重启动驱动程序。

现在可以使用自定义的类。