Novell
Identity Manager Driver for SAP* HR

1.1

www.novell.com

IMPLEMENTATION GUIDE

December 13, 2005

Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not use, export, or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at http://www.novell.com/company/legal/patents/ and one or more additional patents or pending patent applications in the U.S. and in other countries.

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

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

Third-Party Materials

Contents

About This Guide

This guide explains how to install and configure the Identity Manager Driver for SAP* HR. It contains the following sections:

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with Novell Identity Manager. Please use the User Comments feature at the bottom of each page of the online documentation, or go to http://www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of this document, see the Identity Manager Driver for SAP HR Documentation Web site (http://www.novell.com/documentation/idmdrivers)

Additional Documentation

For documentation on using Identity Manager and the other drivers, see the Identity Manager Documentation Web site (http://www.novell.com/documentation/idm).

Documentation Conventions

In this documentation, a greater-than symbol (>) is used to separate actions within a step and items within a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell® trademark. An asterisk (*) denotes a third-party trademark.

Introducing the Identity Manager Driver for SAP HR

1

The Identity Manager Driver for SAP Human Resources (HR), subsequently referred to as the driver, creates an automated link between the SAP HR database and the Identity Vault. This technology enables data flow within a business enterprise based on its own unique requirements, and eliminates the labor-intensive and error-prone practice of re-entering the same data into multiple databases. As new records are added, modified, or deactivated (disabled) in SAP, network tasks associated with these events can be processed automatically.

Because the SAP HR system is the authoritative source of personnel information, the driver allows administrators to propagate this data to other non-SAP business applications and databases without the need for custom integration solutions. Administrators can decide what data will be shared and how data will be presented within their enterprises.

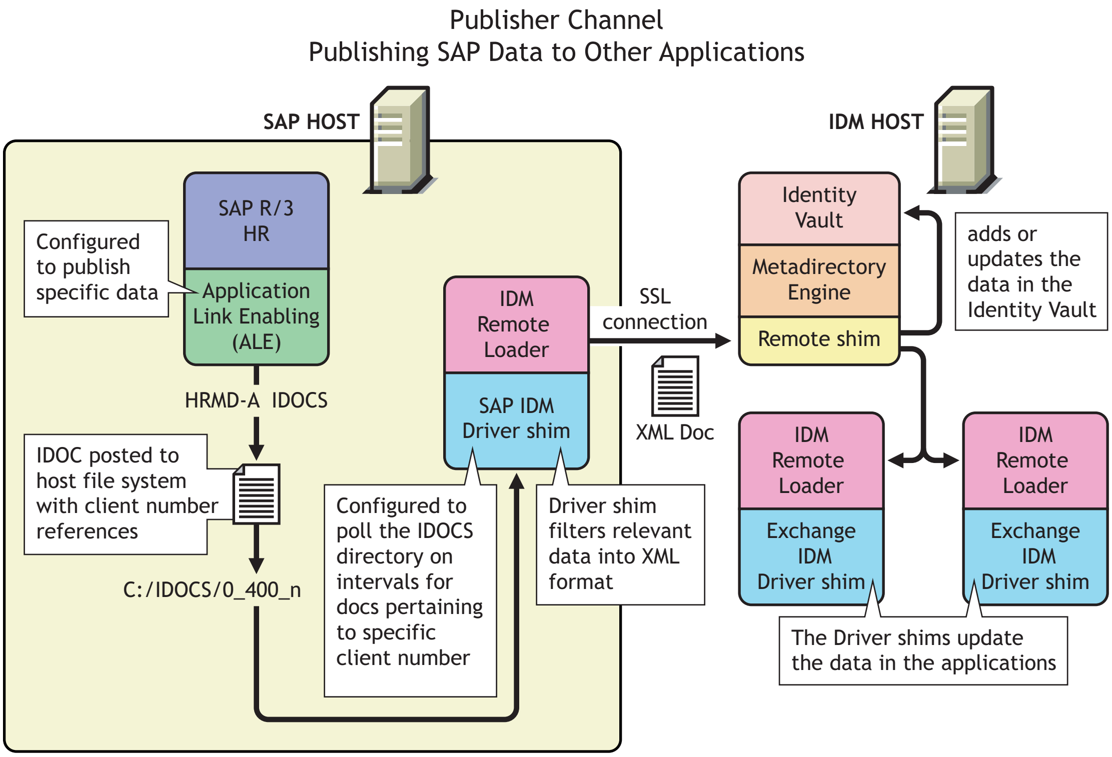
1.1 Understanding Driver Concepts

The driver is a bidirectional synchronization product between SAP R/3 HR systems and the Identity Vault. This framework uses XML to provide data and event transformation capabilities that convert Identity Vault data and events into SAP HR data and vice-versa.

The Identity Vault acts as a hub, with other applications and directories publishing their changes to it. The Identity Vault then sends changes to the applications and directories that have subscribed for them. This results in two main flows of data: the Publisher channel and the Subscriber channel.

1.1.1 Publisher Channel

Figure 1-1 *Publisher Channel Process*


Publisher Channel
Publishing SAP Data to Other Applications
SAP HOST
IDM HOST
SAP R/3 HR
Application Link Enabling (ALE)
Configured to publish specific data
HRMD-A IDOCS
IDOC posted to host file system with client number references
C:/IDOCS/0_400_n
IDM Remote Loader
SAP IDM Driver shim
Configured to poll the IDOCS directory on intervals for docs pertaining to specific client number
Driver shim filters relevant data into XML format
SSL connection
XML Doc
Identity Vault
Metadirectory Engine
Remote shim
adds or updates the data in the Identity Vault
IDM Remote Loader
Exchange IDM Driver shim
IDM Remote Loader
Exchange IDM Driver shim
The Driver shims update the data in the applications

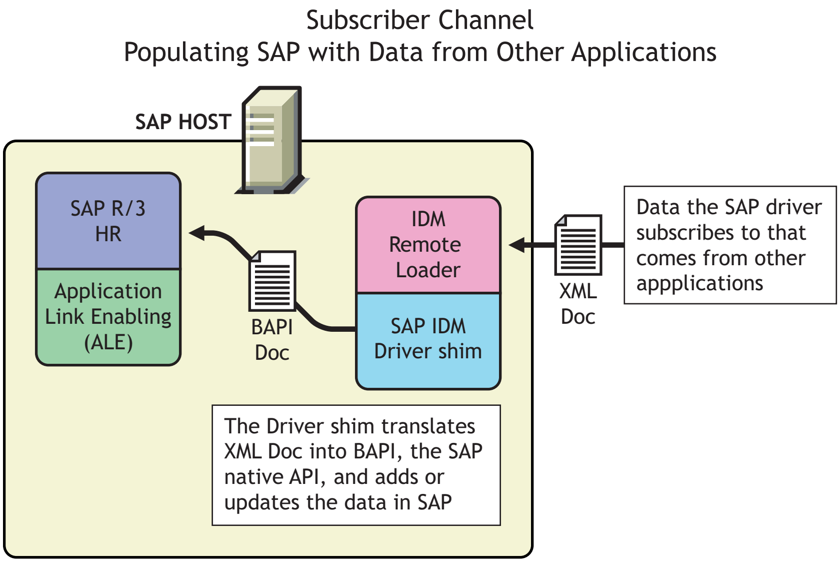
The SAP R/3 HR database publishes information in the form of HRMD_A IDocs using Application Link Enabling (ALE) technology. The driver is only interested in HRMD_A Message IDocs. Any object type in these IDocs can be mapped to an eDirectory object type and subsequently synchronized. The driver consumes the IDoc files and converts the data into XML format.

The Publisher channel polls the SAP HR database for changes, and then submits XML-formatted changes to the Metadirectory engine for publication into the Identity Vault. The engine processes the document by sequentially applying all configured policies based on standard driver process flow.

The driver can then manipulate the information using various policies and filters defined by the system administrator. The driver then submits the data to the Identity Vault. Using eDirectory and other Identity Manager drivers, the data can be shared with other business applications and directories. Based on business rules, these other applications can add additional data that can in turn be inserted back into the SAP HR database using Business Application Programming Interface (BAPI) technology.

1.1.2 Subscriber Channel

Figure 1-2 *Subscriber Channel Process*


Subscriber Channel
Populating SAP with Data from Other Applications
SAP HOST
SAP R/3 HR
Application Link Enabling (ALE)
BAPI Doc
IDM Remote Loader
SAP IDM Driver shim
XML Doc
Data the SAP driver subscribes to that comes from other appplications
The Driver shim translates XML Doc into BAPI, the SAP native API, and adds or updates the data in SAP

The Subscriber channel receives XML-formatted eDirectory events from the Metadirectory engine. The driver then converts these documents to an appropriate data format, and updates SAP via the BAPI interface.

The Identity Vault sends changes only to the applications that have subscribed to receive them.

1.2 Benefits

As the following examples illustrate, the driver enables you to automate and maintain business processes:

- Automatically create an eDirectory account when an individual is hired.
- Automatically delete or deactivate eDirectory accounts when an employee is terminated.
- Synchronize bidirectional data between SAP and eDirectory.
- Maintain accurate and consistent eDirectory IDs.
- Define password policies (for example, a birthdate, social security number, and first and last name combinations).
- Allow seamless integration between SAP and multiple applications (for example, eDirectory, Lotus Notes*, Netscape*, Exchange, and Active Directory*) using Identity Manager and eDirectory.
- Create other eDirectory objects associated with a SAP object (for example, account codes or department records).

You can configure SAP and the Driver objects to enhance your organization's business processes. Before installing and configuring the driver, you evaluate and define those processes. During installation, you configure the driver's policies to automate these processes wherever possible.

For more information about Identity Manager, refer to the *Novell Identity Manager Administration Guide* (http://www.novell.com/documentation/idm/index.html).

1.3 Driver Features

The following section contains information about the driver's features.

- Publisher Channel event status processing

  The Publisher channel treats each object in an IDoc as a unique event. The status of each event determines the appropriate IDoc filename extension. For example, all events with a Warning status are placed together in a file with the `.warn` extension.
- Publisher Channel Only configuration options

  The Publisher Channel Only option in the driver's parameters enables connectivity to a SAP host for read and query operations. The driver vetoes any subscription modifications sent to the SAP system if this option is selected.
- Publisher Connection option

  This option informs the driver whether or not Publisher channel connectivity to the SAP system is desired.
- Publish History Items

  This option specifies whether the driver returns data values that no longer have a current validity period.
- Future-dated IDoc processing

  Future-dated IDoc processing implements a stale event data check. When future-dated events are processed, the driver attempts to confirm the validity period of the event. If no matching validity period is found for the event data, the IDoc data is considered stale and is not applied. Validity checking can only be accomplished if SAP system connectivity is established through configuring the driver's authentication parameters. Publisher Channel Only drivers without connectivity processes all future-dated events at the indicated date.
- Character set encoding is used to parse data from IDocs.

  The driver allows you to specify which character set encoding is used to parse data from IDocs. If nothing is specified, the driver uses the platform default encoding. If you specify a character set incorrectly, the driver initialization fails. You specify this encoding option in the driver configuration parameters.
- Subscriber channel events are applied only to the current instance of SAP Infotype data. Future-dated instances are not affected.
- The Subscriber Channel offers several modes for synchronizing Communication and Internal Data infotypes. All other updates are made as changes to the current valid data.
- The JCOTEST utility validates connectivity.

  A JCOTEST utility validates that all JCO connectivity and authentication parameters are configured correctly.

1.4 Product Components

This section contains information about the following Identity Manager Driver for SAP HR components.

- "Driver Configurations" on page 13
- "Driver Shim" on page 13

- "Schema Map Generation Utility" on page 13
- "SAP Java Connector Test Utility" on page 13

1.4.1 Driver Configurations

Driver configurations provide you with preconfigured policies to get you started with your implementation. The driver configuration for this driver is SAPHR.xml and can be imported through Novell iManager or Designer.

1.4.2 Driver Shim

The driver shim handles communication between the SAP HR database and the Metadirectory engine.

1.4.3 Schema Map Generation Utility

The driver comes packaged with various schema maps of the HRMD_A IDoc file. These maps are generated using a Win32 executable schema map generation utility program called `metamap.exe`.

This program generates a schema file using the SAP RFCSDK and then parses the default schema file into a schema map. The schema map file is named after the IDoc type specified and contains a .meta filename extension (for example, `HRMD_A03.meta`). This program is available in Win32 form only. Default maps of `HRMD_A03.meta` (SAP R/3 version 4.5B) and `HRMD_A05.meta` (SAP R/3 version 4.6C) are provided with the product. Only SAP-defined IDocs can be mapped with the utility. Customized IDocs can be mapped manually if required.

1.4.4 SAP Java Connector Test Utility

Users implementing the driver must download the SAP JCO and install it. The SAP Java Connector (JCO) Test utility enables you to check for JCO installation and configuration issues prior to configuring the driver. You can use the JCO test utility to validate installation and connectivity to the SAP JCO client, as well as testing for accessibility to the HR BAPIs used by the driver. For more information, refer to Section 4.4, "Using the SAP Java Connector Test Utility," on page 38.

1.5 Publishing to the Identity Vault

The SAP HR system is the authoritative source of HR data, and can propagate all Add, Delete, and Modify object event data to the Identity Vault. The Publisher channel is the component used for propagation.

For data to flow from the SAP HR system, the driver utilizes the SAP ALE technology to publish HR Master data records and captures incremental changes using change pointers. The HRMD_A message IDocs are transported using a File port that stores the IDocs on the SAP host system. The driver handles the parsing and filtering of the IDoc file, and provides secure transport of the data to the eDirectory. Only data elements specifically selected by the system administrator are transported from the host system to the Identity Vault.

1.5.1 IDoc Consumption by the Driver

The driver consumes only Output IDoc files with the client number that is reserved for the driver, thus ensuring the privacy of other IDocs that might be generated by another driver configuration. Only the IDoc attributes that have been specified in the driver Publisher filter are published to the Identity Vault.

The format of a successfully published IDoc file is:

```
(O)utput>_<client number>_<consecutive IDoc number>
```

For example:

```
O_300_0000000000001001.
```

After the specified attributes have been published, the filename of the IDoc file is modified to reflect the status of the publication processes. The driver caches the status of every event and associates the status with the object information in the IDoc. If multiple objects are processed from the IDoc, there might be multiple output files with different extensions created.

The following table lists the IDoc status and corresponding suffix:

IDoc Status	Filename Suffix
Processing, but not published	.proc
Processing, but not published (future date IDoc)	.futp
Processed successfully and published	.done
Processed with an error or warning	.F.fail or W.warn
Processed with corrupt or illegitimate data	.bad
Process on date shown in timestamp	**<8 digit timestamp>**.futr

You should determine what action is required, if any, after IDoc publication is complete.

NOTE: Removing the filename extension makes the IDoc available for re-processing.

If a policy generates multiple events from one object, the worst-case status is cached for the IDoc object. For example, if an IDoc contains data for Person object 00001234 and that data triggers policy events for the eDirectory User, his Job, and his Position, three separate <status> elements are returned. If two of the events have a success status, and the third status is warning, the warning status is used.

After all of the objects in the IDoc have been processed, the driver creates output files based on the status of events. If the IDoc contains warning status events, an IDoc file is generated containing all of the objects whose status was warning. The name is a concatenation of the original IDoc name and a `W.warn` extension (for example, `O_001_0002` becomes `O_001_0002W.warn`.) In a similar fashion, if the original IDoc contains error or fatal status events, a file with an `F.fail` extension is generated with those events in it.

To reprocess the IDoc, remove the extension. The use of the *X* character before the extension helps ensure that subsequent reprocessing events do not overwrite the status files from the previous processing attempts.

1.5.2 IDoc Object Types Consumed by the Driver

Object Types vary from system to system and can include objects such as Person, Job, or Organizational Unit. The driver allows the administrator to configure which object types can be processed by the driver.

Only object types specified in the configuration and object types that are in the Publication Filter are processed. The driver parses the data for each object individually and transmits the data to the Metadirectory engine as a single transaction.

NOTE: If SAP connectivity is specified, the driver attempts to populate empty Publisher values by reading values from the SAP server. This only occurs if the Metadirectory engine requests more data (via a query request) when trying to complete an Add event operation.

1.5.3 Attribute Mapping from the SAP HR Database to eDirectory

Schema mapping is used by Identity Manager to translate data elements as they flow between the SAP HR database and eDirectory. The SAP HR schema is based on the SAP HRMD_A message type. The schema map contains all attributes of the various data infotypes in the HRMD_A message types.

Several of the HRMD_A infotypes could be instantiated multiple times on the HR personnel records. Infotypes such as P0006 (Private Address) and P0105 (Communication) might be used several times to indicate unique subtypes. The Private Address infotype might have, for example, Home, Work, or Temporary subtypes. The Communication infotype might contain Cell, Pager, EMail or other subtypes. The eDirectory system administrator can configure the driver to receive whatever subtypes of P0006 and P0105 infotypes are desired. The SAP HRMD_A messages that are generated by the SAP HR system are posted in the form of a text file. The schema map also contains the file position offset and length of each attribute in each segment of infotype data.

This information is presented in a schema map. The map elements have the following format:

```
<Segment Infotype>:<Infotype Attribute>:<Infotype Subtype> or none:
<Segment offset>:<Attribute length>
```

Below are a few examples of maps between SAP HRMD_A attributes and eDirectory attributes. The Infotype P0002 attributes have no possible subtypes. Infotypes P0006 and P0105 have a configurable set of subtypes.

eDirectory Attribute	SAP HR Attribute
Given Name	P0002:VORNA:none:134:25
Surname	P0002:NACHN:none:84:25
City	P0006:ORT01:US01:133:25
Home City	P0006:ORT01:1:133:25
Internet EMail Address	P0105:USRID:MAIL:78:30
Mobile	P0105:USRID:CELL:78:30
Pager	P0105:USRID:PAGR:78:30

eDirectory Attribute	SAP HR Attribute
Home Phone	P0006:TELNR:1:195:14

The driver only utilizes configuration for Private Address (0006) and Communication (0105) infotypes. Mapping of additional instance-specific infotype attributes might cause errors caused by a many-to-one object relationship.

1.6 Subscribing from the Identity Vault

The Subscriber channel of the driver is the component responsible for synchronizing data from the Identity Vault, including data that was obtained from other authoritative data sources, into the SAP HR database. Because the SAP HR system is always viewed as an authoritative source of personnel object creation and deletion, the Subscriber channel is configured to only allow data to be queried, or read, from the SAP HR system, and to allow modification of existing object records.

The Subscriber channel is capable of synchronizing fewer data elements to SAP than the Publisher channel can synchronize to eDirectory. For data to flow from the Identity Vault to the SAP HR system, the driver utilizes SAP-released BAPI functions to make changes to employee records. Because of BAPI restrictions, the driver completely supports only the following infotype data:

- Personal Data (Infotype 0002)
- Private Address (Infotype 0006)
- Communication (Infotype 0105)
- Internal Data (Infotype 0032)

The system administrator specifically selects which attributes from these infotypes can be modified.

Installing the Driver

2

As part of the driver installation and configuration, you should complete the following tasks:

- Section 2.1, “Understanding Driver Prerequisites,” on page 17
- Section 2.2, “Planning for Installation,” on page 17
- Section 2.3, “Overview: Basic Steps,” on page 18
- Section 2.4, “Installing the SAP HR Driver,” on page 19
- Section 2.5, “Installing Driver Configuration Import Files,” on page 21
- Section 2.6, “Importing the Driver Configuration,” on page 22
- Section 2.7, “Activating the Driver,” on page 26

These tasks are explained in detail in this section. After you finish installing the driver and importing the driver configuration file, proceed to Chapter 3, “Understanding ALE Technologies,” on page 27 to learn more about the SAP system configuration requirements.

2.1 Understanding Driver Prerequisites

The driver requires the following prerequisites. Ensure that you meet these criteria before you install the driver.

□ Novell® Identity Manager 3

The system where the driver shim is running must have the SAP Java* Connector (JCO) client technology installed for connectivity to the SAP HR system.

This client is freely available to SAP customers and developer partners through SAP, and is provided for most popular server operating systems. You can download the JCO from the SAP Connectors site (http://service.sap.com/connectors).

□ SAP HR revision level 4.5B or higher.

The driver shim runs on any SAP R/3 host system. As part of the installation, you can install the Remote Loader service on the SAP system. For more information about using SSL to secure the communication between the Remote Loader and the Metadirectory engine, refer to Section 2.4, “Installing the SAP HR Driver,” on page 19.

2.2 Planning for Installation

Before you install and use the driver, you should determine which kind of installation you want to use: local or remote.

When to Use a Local Installation

A local installation installs the driver on the same host computer where you have Identity Manager installed.

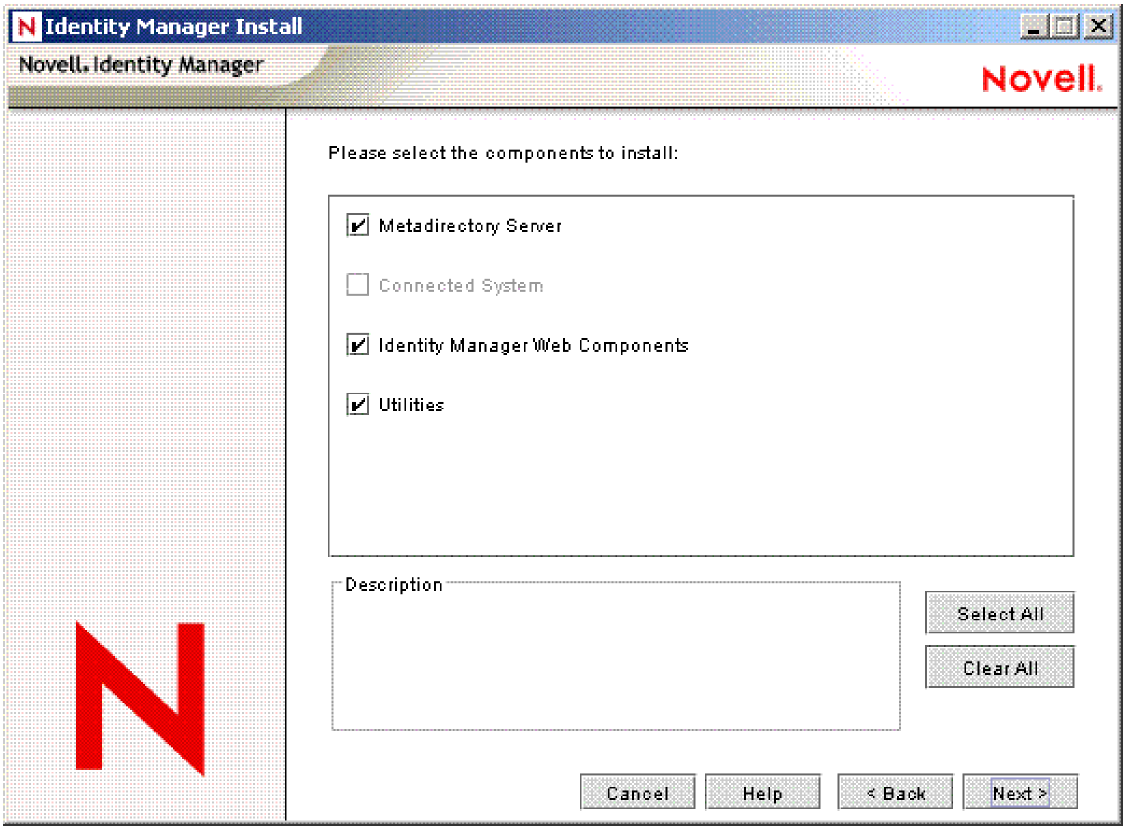
When to Use a Remote Installation

A remote installation installs the driver on a different computer than the one where Identity Manager is installed. Remote installations use SSL encryption to ensure data privacy. You should use this configuration when it is not possible or desirable to run Identity Manager on the SAP host system.

2.3 Overview: Basic Steps

The following figure illustrates options that you can select when installing Identity Manager.

Figure 2-1 *Identity Manager Installation Options*



Please select the components to install:

☑ Metadirectory Server

□ Connected System

☑ Identity Manager Web Components

☑ Utilities

Option	Description
Metadirectory Server	Installs Identity Manager and the Metadirectory engine
Connected System	Installs the Remote Loader
Identity Manager Web Components	Installs the driver configuration file

Installing the SAP HR driver shim requires two basic steps:

Step	What to Select during Installation
1. Install the SAP HR driver shim on the Metadirectory engine server or the Remote Loader server.	Select the Metadirectory Server or Identity Manager Connected System option. See Section 2.4, "Installing the SAP HR Driver," on page 19.
2. Import the driver configuration file for the SAP HR driver through iManager.	Select the Identity Manager Web Components option. See Section 2.5, "Installing Driver Configuration Import Files," on page 21.

Typically, you install the SAP HR driver components when you install Identity Manager (or Remote Loader) and Web components. However, you can install them later.

2.4 Installing the SAP HR Driver

- Section 2.4.1, "Installing the Shim on a Metadirectory Server," on page 19
- Section 2.4.2, "Installing the Shim on a Remote Loader," on page 20

2.4.1 Installing the Shim on a Metadirectory Server

1 On the server where the Identity Vault and the Metadirectory engine are running, launch the Identity Manager installation.

Run the installation program from the Identity Manager CD or the download image.

2 In the Welcome dialog box, click *Next*, then accept the license agreement.

3 In the first Identity Manager Overview dialog box, review information, then click *Next*.

The dialog box provides information on the following:

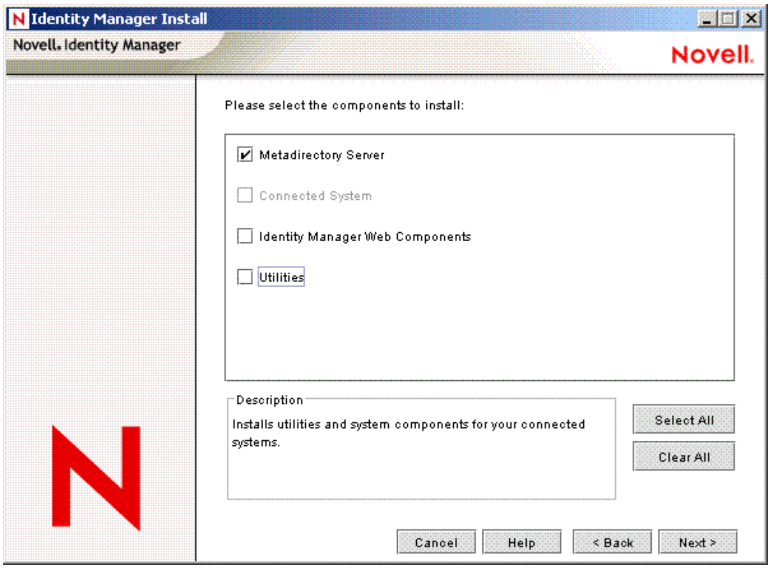
- A Metadirectory Server
- A Connected System Server

4 In the second Identity Manager Overview dialog box, review information, then click *Next*.

The dialog box provides information on the following:

- A Web-based Administration Server
- Utilities

5 In the Please Select the Components to Install dialog box, select *Metadirectory Server*, then click *Next*.



Please select the components to install:

☑ Metadirectory Server

□ Connected System

□ Identity Manager Web Components

□ Utilities

Description

Installs utilities and system components for your connected systems.

If iManager is already installed on this machine, and if you prefer to install the iManager plug-ins and configuration files at this time, also select *Identity Manager Web Components*.

6 In the Select Drivers for Engine Install dialog box, select *Metadirectory Engine* and select *SAP HR*, then click *Next*.

7 In the Identity Manager Upgrade Warning dialog box, click *OK*.

8 In the Schema Extension dialog box, type a username and password, then click *Next*.

9 Review the selected options, then click *Finish*.

2.4.2 Installing the Shim on a Remote Loader

This option enables you to install the SAP HR driver shim to run on a server that is separate from the server running the Metadirectory engine.

1 On the server where the Remote Loader is running, launch the Identity Manager installation.

Run the installation program from the Identity Manager CD or the download image.

2 In the Welcome dialog box, click *Next*, then accept the license agreement.

3 In the first Identity Manager Overview dialog box, review information, then click *Next*.

The dialog box provides information on the following:

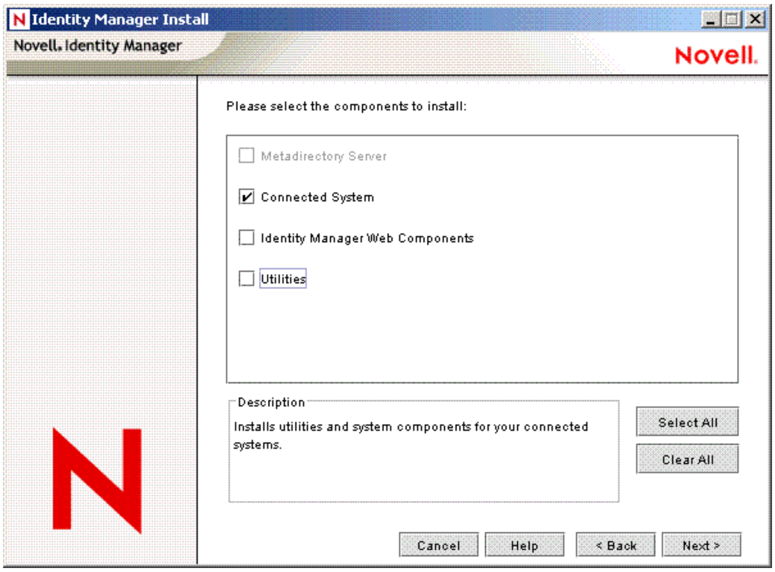
- A Metadirectory Server
- A Connected System Server

4 In the second Identity Manager Overview dialog box, review information, then click *Next*.

The dialog box provides information on the following:

- A Web-based Administration Server
- Utilities

5 In the Please Select the Components to Install dialog box, deselect *Metadirectory Server* and other options, select *Identity Manager Connected System*, then click *Next*.



Please select the components to install:

□ Metadirectory Server

☑ Connected System

□ Identity Manager Web Components

□ Utilities

Description

Installs utilities and system components for your connected systems.

6 Specify the installation path, then click *Next*.

7 In the Select Drivers for Engine Install dialog box, select *Remote Loader Service* and *SAP HR*, then click *Next*.

8 In the Identity Manager Upgrade Warning dialog box, click *OK*.

9 Review the selected options, then click *Finish*.

2.5 Installing Driver Configuration Import Files

This option installs the plug-ins to Identity Manager and the driver configurations. After installing the files, you use iManager to import the SAP HR configuration file into a driver set and configure the driver.

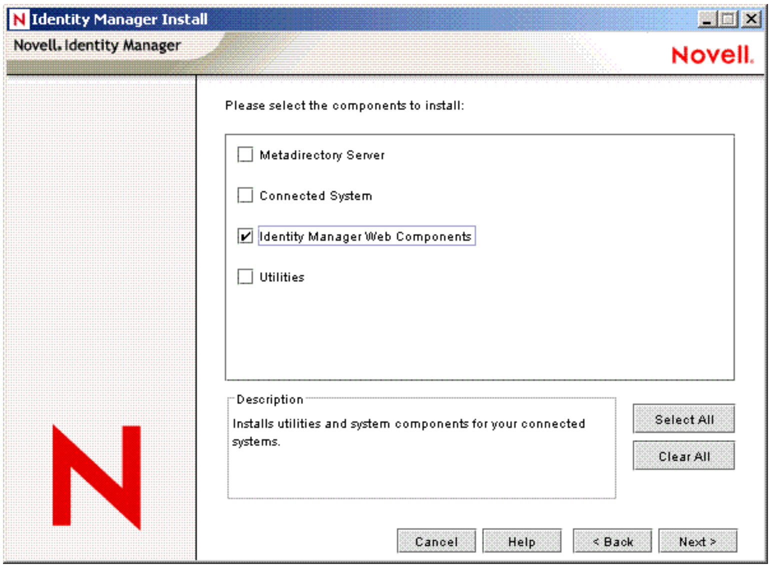
You might have already installed these files, when you installed the Metadirectory engine or Remote Loader. To install the files separately:

1 On the server where iManager is installed, launch the Identity Manager installation.

2 In the Welcome dialog box, click *Next*, then accept the license agreement.

3 In the two Identity Manager Overview dialog boxes, review information, then click *Next*.

4 In the Please Select the Components to Install dialog box, deselect all options except *Identity Manager Web Components*, then click *Next*.



Please select the components to install:

□ Metadirectory Server

□ Connected System

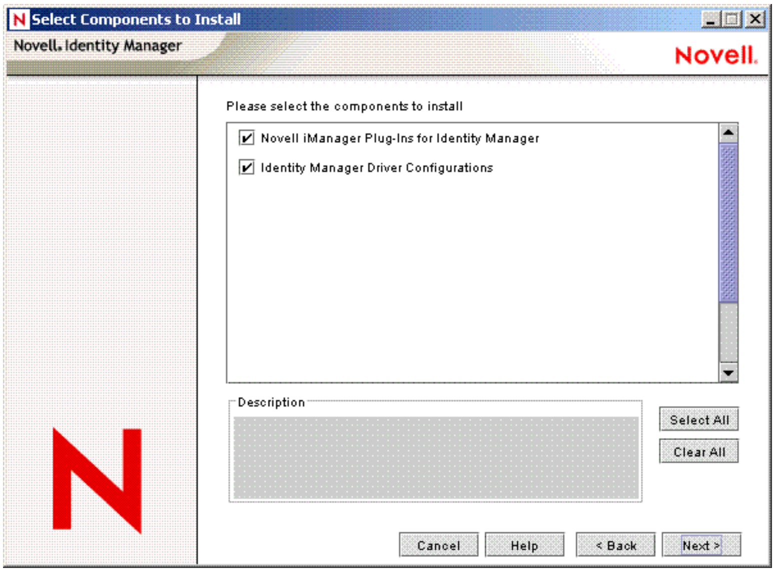
☑ Identity Manager Web Components

□ Utilities

Description

Installs utilities and system components for your connected systems.

5 Select *Identity Manager Driver Configurations*, then click *Next*.



Please select the components to install

☑ Novell iManager Plug-Ins for Identity Manager

☑ Identity Manager Driver Configurations

Description

You can install the driver configuration files when you install the Novell iManager plug-ins, or you can install the files separately.

6 Review the selected options, then click *Finish*.

2.6 Importing the Driver Configuration

The Create Driver Wizard helps you import the basic driver configuration file for SAP HR. This file creates and configures the objects and policies needed to make the driver work properly.

The following instructions explain how to create the driver and import the driver's configuration.

1 In Novell iManager, click *Identity Manager Utilities* > *Create Driver*.

2 Select a driver set.

If you place this driver in a new driver set, you must specify a driver set name, context, and associated server.

3 Select *Import a Driver Configuration from the Server*, then select *SAPHR.xml*.

The driver configuration files are installed on the Web server when you install Identity Manager. During the import, you are prompted for the driver's parameters and other information. Refer to table below for more configuration information.

Parameter Name	Parameter Description
Driver name	The actual name you want to use for the driver.
Organizational Object Container	The name of the Organization Unit object under which published SAP Organizational (O) objects are placed. You can modify this via the driver's Global Configuration Values (GCVs.)
Position Object Container	The name of the Organizational Unit object under which published SAP Position (S) objects are placed. You can modify this via the driver's Global Configuration Values (GCVs.)
Job Object Container	The name of the Organizational Unit object under which published SAP Job (C) objects are placed. You can modify this via the driver's Global Configuration Values (GCVs.)
Active Users Container	The name of the Organizational Unit object where Active users are placed. You can modify this via the driver's Global Configuration Values (GCVs.)
Inactive Users Container	The name of the Organizational Unit object where Inactive users are placed. You can modify this via the driver's Global Configuration Values (GCVs.)
Active Employees Group	The name of the Group object to which Active Employee users are added. To learn more about determining Employee status, refer to "Using the Relationship Query" on page 48. You can modify this via the driver's Global Configuration Values (GCVs.)
Active Managers Group	The name of the Group object to which Active Manager users are added. To learn more about determining Employee status, refer to "Using the Relationship Query" on page 48. You can modify this via the driver's Global Configuration Values (GCVs.)
SAP Client Number	The client number to be used on the SAP application server. This is referred to as the Client in the SAP R/3 logon screen.
SAP Language Code	The language this driver uses for the SAP session. This is referred to as the Language in the SAP R/3 logon screen.
Metadata File Directory	The file system location in which the SAP Metadata definition file resides. By default, this is in the `SAPUtils` subdirectory of the driver's installation directory. **IMPORTANT:** This must be on the same system where the driver shim runs.

Parameter Name	Parameter Description
IDoc File Directory	The file system location in which the SAP HR IDoc files are placed by the SAP ALE system. **IMPORTANT:** This must be accessible to the driver shim process.
Password Failure Notification User	Password synchronization policies are configured to send e-mail notifications to the associated user when password updates fail. You have the option of sending a copy of the notification e-mail to another user, such as a security administrator. If you want to send a copy, you can specify or browse for the DN of that user. Otherwise, leave this field black.
Publisher Channel Only	Select whether you want the driver to use the Publisher channel only or if you want it to use both the Publisher and Subscriber channels.
Enable or Disable Publisher Connection to the SAP Application Server	Select Enable if you want the Publisher channel to read data from the SAP server in addition to IDoc data. Select Disable to use IDoc data only.
SAP Application Server	The host name or IP address for connecting to the appropriate SAP application server. This is referred to as the Application Server in the SAP logon properties.
SAP System Number	The SAP system number on the SAP application server. This is referred to as the System Number in the SAP logon properties.
SAP User ID	The ID of the user this driver uses for the SAP system logon. This is referred to as the User in the SAP R/3 logon screen.
SAP User Password	The User password this driver uses for the SAP system logon. This is referred to as the Password in the SAP R/3 logon screen.
Install Driver as Remote/ Local	Configure the driver for use with the Remote Loader service by selecting the Remote option, or select Local to configure the driver for local use. If Local is selected, you can skip the remaining parameters.
Remote Host Name and Port	Specify the host Name or IP address and port number for where the Remote Loader service has been installed and is running for this driver. The default port is 8090.
Driver Password	The driver object password is used by the Remote Loader to authenticate itself to the Identity Manager server. It must be the same password that is specified as the driver object password on the Identity Manager Remote Loader.
Remote Password	The Remote Loader password is used to control access to the Remote Loader instance. It must be the same password that is specified as the Remote Loader password on the Identity Manager Remote Loader.

The additional driver parameters are set to default values during the import process, but they can be modified in iManager (by clicking the Driver Configuration tab on the driver object.)

Parameter Name	Parameter Description
Character Set Encoding	The character set encoding used to parse data from IDocs. If not specified, the driver uses the platform default encoding. If you incorrectly specify a character set, the driver initialization fails (default: blank)
Master HR IDoc	The name of the IDoc type that is generated by the SAP ALE system to publish SAP HR database Master data modification. If not specified, the driver determines the revision of the SAP HR system and default to the standard IDoc type for that revision of SAP (default: HRMD_A05) This field is optional, unless you select the Publisher channel Only option.
Object Type Code	A list parameter that allows an administrator to specify which HR object types are synchronized (default: P, S, O, and C.)
(Optional) Address Subtype Code	A list of configuration parameters that allows an administrator to specify which subtype of data the SAP Private Address infotype the driver synchronizes (default: 1 and US01)
(Optional) Communication Subtype Code	A list configuration parameter that allows an administrator to specify which subtype data of the SAP Communication infotype the driver synchronizes (default: CELL, MAIL, PAGR.)
Poll Interval (seconds)	Specifies how often the driver polls for unprocessed IDocs (default: 5 seconds.)
Future-dated Event Handling Option	The processing of this option is determined by the Begin and End validity dates of the desired IDoc infotypes. There are four possible values for this parameter. The driver default is to Publish on Future Date. Publish Immediately - Indicates that all attributes will be processed by the driver when the IDoc is available. A time stamp is set for each attribute that represents the validity period. Publish on a Future Date - Indicates that only attributes that have a current or past time stamp will be processed by the driver when the IDoc is available. Future-dated infotype attributes are cached in a `.futr` file to be processed at a future date. Publish Immediately and on Future Date - Indicates that the driver will blend options 1 and 2. All attributes will be processed, with a time stamp, at the time the IDoc is available. All future-dated infotype attributes are also cached in a`.futr` file to be processed at a future date. Publish Immediately and Daily through Future Date - Indicates that the driver will process all events at the time the IDoc is made available. All future-dated infotype attributes are cached in `a .futr` file to be processed again on the next calendar day. This continues until the attributes are sent for a final time on the future date.
Publish History Items	Specifies if data values that are no longer valid are published by the driver (default: *Do Not Publish History Data*.)

Parameter Name	Parameter Description
Communication Change Mode	This Subscriber channel parameter specifies how the driver handles requests to change, remove, or add Communication (Infotype 0105) record instances on employees. There are three modes of operation available. For more information on the functionality of the various modes of operation, see Appendix C, "Subscriber Change Modes and Validity Date Modes," on page 65. Options include: Delimit mode Delete mode Change mode (default driver mode)
Communication Validity Date Mode	This Subscriber channel parameter specifies how Beginning and Ending validity dates are set on newly created Communication record instances on employees. There are two modes of operation available. For more information on the functionality of the various modes of operation, see Appendix C, "Subscriber Change Modes and Validity Date Modes," on page 65.
Internal Data Change Mode	This Subscriber channel parameter specifies how the driver handles requests to change, remove, or add Internal Control Data (Infotype 0032) record instances on employees. There are three modes of operation available. For more information on the functionality of the various modes of operation, see Appendix C, "Subscriber Change Modes and Validity Date Modes," on page 65. Options include: Delimit mode Delete mode Change mode (default driver mode)
Internal Data Validity Date Mode	This Subscriber channel parameter specifies how Beginning and Ending validity dates are set on newly created Internal Control Data record instances on employees. There are two modes of operation available. For more information on the functionality of the various modes of operation, see Appendix C, "Subscriber Change Modes and Validity Date Modes," on page 65. Options include: Default mode Current Date Mode (default driver mode)

4 Specify the driver's parameters, then click *OK* to import the driver.

When the import is finished, you can define security equivalences and exclude administrative roles from replication.

The driver object must be granted sufficient eDirectory rights to any object it reads or writes. You can do this by granting Security Equivalence to the driver object. The driver must have Read/Write access to users, post offices, resources, and distribution lists, and Create, Read, and Write rights to the post office container. Normally, the driver should be given security equal to Admin.

5 Review the driver objects in the Summary screen, and then click *Finish*.

2.7 Activating the Driver

Activation must be completed within 90 days of installation or the driver will not run. For activation information, refer to Activating Novell Identity Manager Products (http://www.novell.com/documentation/idm/install/data/afbx4oc.html) in the **Novell Identity Manager Installation Guide**.

Understanding ALE Technologies 3

3.1 Application Link Enabling Technology

Application Link Enabling (ALE) technology enables communication between SAP and external systems such as Novell® eDirectory™. ALE is comprised of various components. When configuring the SAP system to enable the driver, you should consider the following ALE components and their relationship to the driver:

- Clients and Logical Systems
- Message Types
- IDoc Type
- Distribution Model
- Partner Profiles
- Port Definition
- File Port
- Change Document/IDoc Outbound Processing

Refer to Section 4.1, "Configuring the SAP System," on page 31 for instructions on how to configure these SAP system parameters.

3.2 Clients and Logical Systems

In the SAP configuration for the driver, a logical system is a representation of either a SAP system or an external system. The logical system is used to distribute data to and from SAP. Every R/3 or SAP system needs to have a base logical system associated with a client. There is a one-to-one relationship between the client and the logical system.

The driver uses an outbound ALE interface. In an outbound ALE interface, the base logical system becomes the **sender** for outbound messages and the **receiver** of inbound messages. A SAP user is likely logged into the base logical system/client when making changes to the database (for example, hiring an employee, updating position data, terminating an employee, etc.) A logical system must also be defined for the receiving process. This logical system acts as the receiver of outbound messages.

3.3 Message Type

A message type represents the type of data that is exchanged between the two systems. For the driver, the HRMD_A message type is used. A message type characterizes data being sent across the systems and relates to the structure of the data, also known as an IDoc type (for example, HRMD_A05).

3.4 IDoc Type

Intermediate Document (IDoc) Type represents the structure of the data associated with a message type. ALE technology uses IDocs to exchange data between logical systems. An IDoc is an object with the data of a specific message type in it. IDocs consist of three record types:

1. The control record
2. The data record
3. The status record

The control record contains information about the IDoc, such as what IDoc type it is, the message type, the sending and receiving systems, direction, etc.

The data record contains the application data. Data records consist of several fields that describe the content of the specific object.

The status record contains data on the state of the processing of the IDoc.

3.5 Distribution Model

The distribution model is a tool that stores information about the flow of message types between systems. A distribution model must be configured when setting up the driver. After the two logical systems have been defined and you have a general understanding of message types and IDocs, you can configure your distribution model.

The distribution model determines what message types can be sent from a client to another client, as well as the sending and receiving systems. Filters for IDoc segments can also be applied to distribution models.

3.6 Partner Profiles

Partner profiles specify the components used in an outbound process. Some of these components include the IDoc type, message type, IDoc size, mode, and the person to be notified in case of errors.

3.7 Port

A port is the communication link between the two logical systems.

3.8 Port Definition

A port definition is used in an outbound process to define how documents are transferred to the destination system.

3.9 File Port

A file port is used when IDocs are transferred to a file.

3.10 Change Pointers

Change pointers capture a master data change in SAP for a specific message type. These changes are saved into a change document. For example, when a new employee is hired, a change is made and captured in a change document.

3.11 Change Document/IDoc Outbound Processing

A SAP variant is defined for the HRMD_A0# message type. After the variant is defined, a job is scheduled for that variant, which captures the change documents and converts them into IDocs. The outbound process is then triggered.

NOTE: Multiple change documents can be captured within a single IDoc. The number of IDocs is determined by how frequently jobs are scheduled, not by the number of change documents created. For example, several records may be added, modified, or deleted within the specified job process period. All of these changes are included in a single IDoc.

Configuring the SAP System

4

You must configure the SAP system parameters to enable Application Link Enabling (ALE) processing of HRMD_A IDocs. This allows for data distribution between two application systems, also referred to as messaging. Novell® follows SAP's general guidelines for configuring BAPI (Business Application and Programming Interface) and ALE technologies.

This section contains the following sections:

- Section 4.1, "Configuring the SAP System," on page 31
- Section 4.2, "Using the Schema Metadata File," on page 36
- Section 4.3, "Using the Schema Map Generation Utility," on page 37
- Section 4.4, "Using the SAP Java Connector Test Utility," on page 38

4.1 Configuring the SAP System

As part of configuring the SAP system, you should complete the following steps in this order:

1. "Defining Sending and Receiving Systems" on page 31
2. "Creating a Distribution Model" on page 32
3. "Creating a Port Definition" on page 33
4. "Generating Partner Profiles" on page 33
5. "Generating an IDoc" on page 34
6. "Activating Change Pointers" on page 34
7. "Scheduling a Job for Change Pointer Processing" on page 35
8. "Scheduling a Job" on page 35
9. "Testing the Change Pointer Configuration" on page 35
10. "Creating a CPIC User" on page 36

NOTE: The following instructions are for SAP version 4.6C. If you are using a previous version of SAP, the configuration process is the same; however, the SAP interface is different.

4.1.1 Defining Sending and Receiving Systems

The sending and receiving systems must be defined for messaging. In order to distribute data between systems, you must first define both the sending and receiving systems as unique logical systems.

You must assign a client to the sending logical system. Since the receiving logical system is an external system, there is no need to assign it to a client. You should never assign the same client to more than one logical system.

For this particular solution, we recommend defining two logical systems. One logical system acts as the receiver and the other logical system acts as the sender. Although only one of these logical systems is used as a data source process (that is, the client/logical system where employee data is stored and “actions” occur), the second logical system is needed to represent the receiving process (in this case, the driver.)

NOTE: Depending on your current SAP environment, you might not need to create a logical system. You might only need to modify an existing distribution model by adding the HRMD_A message type to a previously configured model view. For more information, see “Creating a Distribution Model” on page 32.

It is important, however, that you follow SAP’s recommendations for logical systems and configuring your ALE network. The following instructions assume that you are creating new logical systems and a new model view.

Creating a Logical System

1. In SAP, type transaction code `BD54`.
2. Click *New Entries*.
3. Type an easily identifiable name to represent the SAP **sender** system. SAP recommends the following format for logical systems representing R/3 clients: *systemID*CLNT*client number* (such as ADMCLNT100).
4. Type a description for the logical system (such as Central System for SAP HR Distribution).
5. Add a second logical system name to represent the Identity Manager external **receiver** system (such as DRVCLNT100).
6. Type a description for the logical system (such as IDM HR Integration).
7. Save your entry.

Assigning a Client to the Logical System

1. In SAP, type transaction code `SCC4`.
2. Click *Table View* > *Display* > *Change* to switch from display to change mode.
3. Select the client from which you want User information distributed (such as 100).
4. Click *Goto* > *Details* > *Client Details*.
5. In the Logical System field, browse to the **sender** logical system you want to assign to this client (such as ADMCLNT100).
6. Save your entry.

4.1.2 Creating a Distribution Model

The distribution model contains essential information about message flow. The model view defines the systems that will communicate with each other and the messages that will flow between them. The distribution model forms the basis of distribution and controls it directly.

To create a distribution model:

1. Verify that you are logged on to the sending system/client.
2. In SAP, type transaction code `BD64`. Ensure that you are in Change mode (click *Table View* > *Display* > *Change*.)
3. Click *Edit* > *Model View* > *Create*.
4. Type the short text to describe the distribution model (such as Client 100 Distribution to IDM).

5 Type the technical name for the model (such as SAP2IDM).

6 Accept the default *Start* and *End* dates or specify valid values. Click the check mark icon to save your entry.

7 Select the view you created, then click *Add Message Type.*

8 Define the sender/logical system name.

9 Define the receiver/server name.

10 Define the Message Type you want to use (*HRMD_A*), then click *Continue*.

11 Click *Save*.

4.1.3 Creating a Port Definition

The port is the communication channel to which IDocs are sent. The port describes the technical link between the sending and receiving systems. You should configure a file port for this solution. The file port is used to determine the directory and the file location to which IDocs are sent.

To create a file port definition:

1 Type transaction code `WE21`.

2 Select *File*, then click the *Create* icon. Specify information for the following fields:

- Name port
- Port description
- Version: Select SAP release 4.X

3 On newer SAP servers, it is possible that the database is Unicode. If this is true, select the *Unicode Format* checkbox on the S*ystem Setting* tab.

4 Define the outbound file:

4a Select the physical directory. This is the directory where you want IDocs placed. You might need to create this directory.

Enter the directory where the outbound files are written, for example: `\\SAPDEV\NOV\SYS\GLOBAL\SAPNDSCONNECTOR`.

4b Enter the function module. This names the IDoc file in a specific format. Always use the following format: `EDI_PATH_CREATE_CLIENT_DOCNUM`.

5 Save your changes.

NOTE: You do not need to configure the other three tabs for the port properties (outbound:trigger, inbound file, and status file).

4.1.4 Generating Partner Profiles

The system automatically generates a partner profile or you can manually maintain the profile.

NOTE: If you are using an existing distribution model and partner profile, you do not need to automatically generate a partner profile. Instead, you can modify it to include the HRMD_A message type.

To automatically generate a partner profile:

1. Type transaction code `BD82`.
2. Select the model view. This should be the model view previously created in "Creating a Distribution Model" on page 32.
3. Ensure the *Transfer IDoc immediately* and *Trigger Immediately* option buttons are selected.
4. Select a reasonable packet size value to ensure that IDoc files are not too large to process. We recommend a value of 100.
5. Click *Execute*.

Modifying Port Definition

When you generated a partner profile, the port definition might have been entered incorrectly. For your system to work properly, you need to modify the port definition.

1. Type transaction code `WE20`.
2. Select *Partner Type LS*.
3. Select your receiving partner profile.
4. Select *Outbound Parameters*, then click *Display*.
5. Select message type *HRMD_A*.
6. Click *Outbound Options*, then modify the receiver port so it is the file port name you created in "Creating a Port Definition" on page 33.
7. From the Output Mode, select *Transfer IDoc Immediately* to send IDocs immediately after they are created.
8. From the IDoc Type section, select the latest version available for your system.
9. Click *Continue/Save*.

4.1.5 Generating an IDoc

1. Type transaction code `PFAL`.
2. Insert the *Object Type P* for person objects.
3. Enter an Employee's ID for the *Object ID* or select a range of employees.

   Under the *Parallel Processing* tab, set *Number of Objects per Process* to *100* if you select a range of employees.
4. Click *Execute*.

   Ensure that the status is set to P*assed to Port Okay.*

   The IDoc has been created. Go to the directory where IDocs are stored (it was defined in the file port setup) and verify that the IDoc text file was created.

4.1.6 Activating Change Pointers

To activate change pointers globally:

1. Type transaction code `BD61`.
2. Enable the *Change Pointers Active* tab.

To activate change pointers for a message type:

1 Type transaction code `BD50`.

2 Scroll to the *HRMD_A message type*.

3 Select the *HRMD_A* check box, then click *Save*.

4.1.7 Scheduling a Job for Change Pointer Processing

1 Type transaction code `SE38` to begin defining the variant.

2 Select the *RBDMIDOC program,* select *Variant*, then click the *Create* icon.

3 Name the variant and give it a description.

NOTE: Make note of the variant name so you can use it when scheduling the job.

4 Select the *HRMD_A* message type, then click *Save*.

You will be prompted to select variant attributes. Select the background processing attribute.

5 Click *Save*.

4.1.8 Scheduling a Job

1 Type transaction code `SM36`.

2 Name the job.

3 Assign Job Class.

Job Class is the priority in which jobs are processed. Class *A* is the highest priority and will be processed first. For a production environment, we recommend assigning the class to *B* or *C*.

4 Schedule a start time. Click the *Start Condition* tab, then click *Date and Time*. Specify a scheduled start time, which must be a future event.

4a Mark the job as a periodic job > click the *Periodic Values* tab, schedule how frequently you want the job to run, then press *Enter*. For testing purposes, we recommend setting this period to 5 minutes.

4b Click *Save*.

5 Define the job steps.

5a Type the ABAP program name: `RBDMIDOC`.

5b Select the variant you created in the previous step.

6 Click *Save*.

IMPORTANT: Click *Save* once; otherwise, the job will be scheduled to run multiple times.

4.1.9 Testing the Change Pointer Configuration

1 From the SAP client, hire an employee.

2 Ensure that an IDoc was created.

You can verify IDoc creation in two locations:

- Type transaction code `WE02`
- Go to the IDoc file locations

4.1.10 Creating a CPIC User

Users are client-dependent. For each client that will be using the driver, a system user with CPIC access must be created.

1. From *User Maintenance in SAP,* specify a username in the user dialog box, then click the *Create* icon.
2. Click the *Address* tab, then specify data in the *Last Name* and *Format* fields.
3. Click the *Logon Data* tab, then define the initial password and set the user type to *CPIC*.
4. Click the *Profiles* tab, then add the *SAP_ALL*, *SAP_NEW* and *S_A.CPIC* profiles.
5. Click *Save*.

Initially, you can create a dialog user to test your SAP system configuration. If there are processing problems, you can analyze the dialog user in the debugger. You should also log into the SAP system once to set this user's password. After the system is tested and works properly, you should switch to a CPIC user for security measures.

IMPORTANT: If restricted rights are assigned to the CPIC User, the Identity Manager and SAP administrators are responsible to ensure that sufficient rights are assigned to enable the configured level of integration. Appendix B, "Driver BAPIs," on page 63 contains a table describing which BAPIs the driver uses.

4.2 Using the Schema Metadata File

The driver includes to default Metadata files: `HRMD_A03.meta` and `HRMD_A05.meta`. These files contain the SAP metaschema definitions of the HRMD_A03 IDoc type, which is the standard HR Master Data IDoc for version 4.5B of SAP R/3; and the HRMD_A05 IDoc type, which is the standard HR Master Data IDoc for version 4.6C.

These files are provided to two distinct purposes:

1. The driver uses a metadata file to generate an Application Schema Map when requested via the *Refresh Application Schema* option in iManager.
2. If a *Character Set Encoding* value is specified in the configuration, the driver opens the metadata file to determine if the encoding value specified is valid.

A schema map must exist for the IDoc type that the driver consumes, whether that type is specified in the *Master HR IDoc* configuration parameter or if the driver selects a default type based on the version of the SAP Application server. Because only two maps are provided with the driver, you might need to create a new map for the IDoc type needed by the driver. There are two options for doing this:

- You can simply copy the `HRMD_A05.meta` file to a new file, such as `HRMD_A06.meta`. This is acceptable as long as you do not need to publish newer infotypes not found in the `HRMD_A05` version. It is unlikely that newer infotypes will be needed.
- You can execute the `metamap.exe` Schema Map Generation Utility. See Section 4.3, "Using the Schema Map Generation Utility," on page 37 for more information.

4.2.1 Schema Metadata File Reduction

The size of the metaschema definitions can create problems for your driver configuration. The schema refresh can take a long time to process, especially because a copy of the map is generated for each object type you choose to synchronize. Additionally, the size of the schema in the driver configuration can be extremely large and cumbersome to navigate. For these reasons, it is acceptable to reduce the number of infotypes in the metadata files.

You can edit the appropriate metadata file and remove all infotypes that are not used for your implementation. Simply search for the infotypes to remove (for examples, Infotype 0008 values can be found by searching for P0008) and deleting the SEGMENT: line and subsequent infotype field lines from the file. You should modify a copy of the original file. For most integrations, only 20-30 percent of the infotypes are actually used.

IMPORTANT: You must be careful that you do not remove infotypes that are useful for policies or other object types being synchronized. Two infotypes of this nature are Infotype 1000 (for Descriptions of non-person objects) and Infotype 1001 (Relationships between objects.) These are both used in the default driver configuration.

You should also not remove fields from infotypes that are used in your integration. Field removal is extremely hard to detect if a mistake is made or if you want to return to an earlier version.

4.2.2 Schema Metadata File Extension

There are many situations where an IDoc is extended with custom infotypes or infotype fields. Because the schema map is based on standard SAP IDoc types, you must manually create these types of metadata extensions. There are several areas of concern:

- If the infotype is an extension to the IDoc (for example, Infotype Z0001), you must ensure that the infotype header fields are present in a standard format. These standard fields start with the field PERNR and extend through field RESE2 in data infotypes. If these fields are not present or contain no data, many of the driver features such as future-dating and history-dating do not work.
- The format of new infotypes is similar to the standard infotypes. The first field should be <5 character infotype>:PERNR:0:8. When parsing an actual IDoc, the physical offset for the PERNR field is 63 (when starting from position 0.)

You might also create schema extensions directly to the Mapping Rule without the need to update the metadata file. If you choose this option, which is often easier, remember the physical offset mentioned above when determining where your data fields of interest begin. The format for a direct mapping is described in Section 1.5.3, "Attribute Mapping from the SAP HR Database to eDirectory," on page 15. Selecting field names is up to you, because the driver does not use them for processing, but they should be limited to 5 characters for consistency.

4.3 Using the Schema Map Generation Utility

The driver comes packaged with various schema maps of the HRMD_A IDoc file. These maps are generated using a Win32 executable schema map generation utility program called `metamap.exe`.

The schema map generation utility is installed in the SAPUTILS folder in the remote loader or the Novell eDirectory™ directory. It contains the following files:

- `Metamap.exe`
- `SAPRFC.INI`
- `HRMD_A03.meta` and `HRMD_A05.meta`
- `Logon.txt`
- `Readme.txt`

This program generates a schema file using the SAP RFCSDK and then parses the default schema file into a schema map. The schema map file is named after the IDoc type specified and contains a `.meta` filename extension (for example, `HRMD_A03.meta`). This program is available in Win32 form only. Only IDocs defined by SAP can be mapped with this utility. Custom IDocs can only be mapped manually using the base `.meta` file.

4.3.1 Editing SAPRFC.INI and LOGON.TXT

Follow the directions in the `readme.txt` file to configure these files for use on your SAP system.

4.4 Using the SAP Java Connector Test Utility

The driver uses the SAP Java Connector (JCO) and Business Application Programming Interface (BAPI) technologies to connect to and integrate data with eDirectory. The SAP JCO is a SAP client that creates service connections to a SAP R/3 system. After the driver is connected to the R/3 system, it calls methods on business objects within the R/3 system via BAPI.

This utility enables you to check for JCO installation and configuration issues prior to configuring the driver. Use the JCO test utility to validate installation and connectivity to the SAP JCO client, as well as testing for accessibility to the HR BAPIs used by the driver.

In order to configure the driver, you must first download the SAP JCO and install it. For installation instructions, refer to the documentation accompanying the SAP JCO.

There might be minor modifications to JCO components as the connector is updated by SAP. Always refer to the SAP installation documentation for proper configuration instructions.

4.4.1 What Does the Utility Do?

The SAP JCO Test utility completes the following checks:

- Ensures that the `jco.jar` file, which contains the exported JCO interface, is present.
- Ensures that the JCO native support libraries are properly installed.
- Ensures that connection parameters to the SAP R/3 target system are correct.
- Ensures that the authentication parameters to the SAP R/3 target system are correct.
- Ensures that the selected language code is valid.
- Ensures that the BAPIs used by the driver are present as expected for the version of the SAP R/3 target system.

4.4.2 Utility Prerequisites

Before you run the JCO Test utility, you must install the SAP JCO client for the desired platform. The JCO can only be obtained from the SAP Service Marketplace Web site (http://www.sap-ag.de/

services). The download is free to any SAP software customer or development partner, but you are required to log in.

Follow the installation instructions for your platform. Each installation requires you to set one or two environment variables, such as CLASSPATH for the jco.jar file location. For the UNIX* platforms, set either the LD_LIBRARY_PATH or LIBPATH variables for the location of native support libraries. Ensure that these variables are set in the shell environment to run this test and for the subsequent use of the Identity Manager Driver for SAP HR.

You must also make sure that you have your PATH environment variable set to include the path to your Java executable file. For Win32 platforms, the environment variables are set via the System configuration in the Control Panel. On UNIX systems, edit the appropriate .profile or .bash_profile to include and export these path variables.

4.4.3 Components

The JCO Test utility includes a `JCOTest.class` file. You need to create a batch or script file to run the test. The format of the batch or script file varies, depending on the platform on which the JCO client has been installed.

The basic content of the file includes a path to the Java executable (or just java if your PATH is appropriately configured), and the name of the JCOTest.class file. A sample UNIX script file and Win32 batch file are listed below, where `jco.jar` is in the executable directory of the JCOTest.class file and the batch file:

```
Win32 jcotest.bat file
java -classpath %CLASSPATH%;. JCOTest

Unix jcotest file
java JCOTest
```

You must use proper slash notation when specifying pathnames and use the proper classpath delimiter for the platform. You must also remember that the name of the `jco.jar` or `sapjco.jar` file is case-sensitive on UNIX platforms and that the name of the test class, `JCOTest`, must be specified with proper case for any platform.

4.4.4 Running and Evaluating the Test

Running the Test

To run the JCO Test utility on a Win32 platform:

1 From Windows Explorer, double-click your `.bat` file.

or

From a command prompt, run your `.bat` script.

To run the JCO Test utility on a UNIX platform:

1 From your preferred shell, run your `jcotest` script file.

NOTE: When you run the test program, an error message might appear before any test output is displayed. This indicates an improper installation of the JCO client components. The error messages are documented for each platform in "Understanding Test Error Messages" on page 41.

Evaluating the Test

If the JCO client is installed properly, the following output is displayed:

```
**The SAP JCO client installation has been verified to be correct.

Version of the JCO-library: version information

Input SAP Server Connection Information
----------------------------------------
```

You then receive a series of prompts for connection and authentication information. All data must be provided unless a default value, identified by [] delimiters, is provided. Failure to fill in a response value to each prompt ends the test. Enter information for the following fields when prompted:

- Application server name or IP address
- System number[00]
- Client number
- User
- User Password
- Language code [EN]

The values you provide are the same values that could be used to authenticate via the SAPGUI client. Based on the validity of the input, the test either displays error messages with solution suggestions or runs to completion. At the end of the test, a status message displays. If the test indicates full functionality as required by the driver, the following status message appears (it describes valid values that can be used as the configuration parameters for the driver:

```
**All expected platform support is verified correct.

JCO Test Summary
----------------
Full JCO/BAPI Functionality has been verified.
The following parameters may be used for SAP HR Driver Configuration

Authentication ID: Username
Authentication Context: SAP Host Name/IP Address
Application Password: User password
Publisher Channel Only? 1
SAP System Number: System Number
SAP User Client Number: Client Number
SAP User Language: Language Code
Master HR IDoc: Default IDoc type for SAP R/3 version
```

If the test indicates that the functionality required by the driver is not available, the following status message is displayed:

```
**There are <number> required BAPI functions NOT supported on this
platform.

JCO Test Summary
----------------
JCO/BAPI functionality issues have been detected that will prevent
proper SAP HR Driver functionality.
```

Post-Test Procedures

After the JCO Test Utility has passed all tests successfully, the driver can be configured to run. Make sure that the `jco.jar` file is copied to the location where the `sapshim.jar` file has been installed.

On UNIX systems, ensure that the environment variables used for the successful completion of the JCO Test are also in the environment of the driver. If these conditions are met, there should be no driver errors that are related to the JCO.

4.4.5 Understanding Test Error Messages

Use the information in this section to analyze error messages that might display during the JCO Test. Some errors are applicable to all platforms, and other errors are platform-specific.

The test has been run on the platforms listed below. Other UNIX platforms supported by JCO are configured in a similar manner and errors generated by improper JCO installation and configuration should be similar to the errors described for IBM*-AIX* and Solaris*.

- "General Errors" on page 41
- "Errors on Win32 Systems" on page 42
- "Errors on IBM-AIX Systems" on page 42
- "Errors on Solaris Systems" on page 43
- "Errors on Linux Systems" on page 43

General Errors

Error Message	Problem
Error connecting to SAP host: com.sap.mw.jco.JCO$Exception: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed Check values of Application Server Name/IP Address and System Number	This indicates that one or both of the values entered for Application Server Name or IP address and System Number are incorrect. Verify that these values are consistent with the information found in the Properties page of the SAP Logon dialog box used to connect to the SAP R/3 system.
Error authenticating to SAP host: com.sap.mw.jco.JCO$Exception: (103) RFC_ERROR_LOGON_FAILURE: You are not authorized to logon to the target system (error code 1).	The authentication credentials are not valid. Verify that the values for Client Number, User, and User Password are correct.

Error Message	Problem
Error connecting to SAP host: com.sap.mw.jco.JCO$Exception: (101) RFC_ERROR_PROGRAM: Language ’<value>’ not availableCheck value of Language Code	The language code selected is not valid or is not installed on the SAP R/3 system.

Errors on Win32 Systems

Error Message	Problem
“jcotest' is not recognized as an internal or external command, operable program, or batch file.	The `jcotest.bat` batch file is not present.
Exception in thread “main” java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$AbapExceptionor Exception in thread “main” java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$Exception	The `sapjco.jar file` is not in the location specified in the jcotest.bat batch file.
Exception while initializing JCO client.java.lang.UnsatisfiedLinkError: no jRFC12 in java.library.path Verify proper installation of JCO Native support libraries packaged with JCO client.	The `jRFC12.dll` file that shipped with the JCO client is not installed or is installed in an improper location. The default location for j`RFC12.dll` and `libRfc32.dll` is `/ WINNT/system32`.
Exception while initializing JCO client.java.lang.UnsatisfiedLinkError: C:\WINNT\system32\jrfc12.dll: Can’t find dependent libraries. Verify proper installation of JCO Native support libraries packaged with JCO client.	The `librfc32.dll` file shipped with the JCO client is not installed or installed in an improper location. The default location for `jRFC12.dll` and `libRfc32.dll` is `/WINNT/system32.`

Errors on IBM-AIX Systems

Error Message	Problem
ksh: jcotest: not found.	The `jcotest` script file is not present in the directory.
Exception in thread “main” java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$AbapExceptionorException in thread “main” java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$Exception	The `sapjco.jar` file is not in the location specified in the `jcotest` script file or the case specified for `sapjco.jar` does not match the actual file name.
Exception while initializing JCO client.java.lang.UnsatisfiedLinkError: no jRFC12 (libjRFC12.a or .so) in java.library.path. Verify proper installation of JCO Native support libraries packaged with JCO client.	The `libjRFC12.so` file that shipped with the JCO client is not installed or is installed in an improper location. You must configure a LIBPATH environment variable to specify the location in which the file resides.

Error Message	Problem
Exception while initializing JCO client.java.lang.UnsatisfiedLinkError: <path>/ libjRFC12.so: A file or directory in the path name does not exist. Verify proper installation of JCO Native support libraries packaged with JCO client.	The `librfccm.so` file shipped with the JCO client is not installed or is installed in an improper location. You must copy the file to the same location as `libjRFC12.so` or configure the LIBPATH environment variable to specify the location in which the file resides.

Errors on Solaris Systems

Error Message	Problem
ksh: jcotest: not found.orbash: jcotest: command not found	The `jcotest` script file is not present in the directory.
Exception in thread "main" java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$AbapExceptionorException in thread "main" java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$Exception	The `jsapjco.jar` file is not in the location specified in the `jcotest` script file or the case specified for `sapjco.jar` does not match the actual file name.
Exception while initializing JCO client.java.lang.UnsatisfiedLinkError: no jRFC12 in java.library.path Verify proper installation of JCO Native support libraries packaged with JCO client.	The `libjRFC12.so` shipped with the JCO client is not installed or is installed in an improper location. You must configure a LD_LIBRARY_PATH environment variable to specify the location in which the file resides.
Exception while initializing JCO client.java.lang.UnsatisfiedLinkError: <path>/ libjRFC12.so: ld.so.1: <search-path>: fatal: librfccm.so: open failed: No such file or directory Verify proper installation of JCO Native support libraries packaged with JCO client.	The `librfccm.so` file shipped with the JCO client is not installed or installed in an improper location. You must copy the file to the same location as `libjRFC12.so` or configure the LD_LIBRARY_PATH environment variable to specify the location in which the file resides.

Errors on Linux Systems

Error Message	Problem
ksh: jcotest: not found.orbash: jcotest: command not found	The `jcotest` script file is not present in the directory.
Exception in thread "main" java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$AbapExceptionorException in thread "main" java.lang.NoClassDefFoundError: com/sap/mw/jco/ JCO$Exception	The `sapjco.jar` file is not in the location specified in the `jcotest` script file or the case specified for `sapjco.jar` does not match the actual file name.

Error Message	Problem
Exception while initializing JCO client.java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer ’com.sap.mw.jco.rfc.MiddlewareRFCno jRFC12 in java.library.path. Verify proper installation of JCO Native support libraries packaged with JCO client.	The `libjRFC12.so` file shipped with the JCO client is not installed or is installed in an improper location. You must configure a LD_LIBRARY_PATH environment variable to specify the location in which the file resides
Exception while initializing JCO client.java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer ’com.sap.mw.jco.rfc.MiddlewareRFC<path>/ libjRFC12.so: librfccm.so: cannot open shared object file: No such file or directory. Verify proper installation of JCO Native support libraries packaged with JCO client.	The `librfccm.so` file shipped with the JCO client is not installed or is installed in an improper location. You must copy the file to the same location as `libjRFC12.so` or configure the LD_LIBRARY_PATH environment variable to specify the location in which the file resides.

Understanding the Default Driver Configuration

5

This section explains how the default driver configuration uses policies and filters. You can use this overview as a basis to create your own policies and filters for specific business implementations.

5.1 Using Policies

Policies are highly configurable for use within any business environment. Although each business is different, the default driver configuration is built with a scenario that involves synchronizing SAP Person (P), Organization (O), Position (S), and Job (C) objects into the Identity Vault.

5.1.1 Modifying Policies and the Filter

You must modify policies and filters to work with your specific business environment. We recommend that you make modifications in this order:

- Modify the driver filter to include desired attributes to be synchronized.
- Modify the Mapping policy to include all attributes specified in the driver filter.
- Modify the InputTransformation policy
- Modify the OutputTransformation policy
- Modify the Publisher Placement policy
- Modify the Publisher Matching policy
- Modify the Publisher Creation policy
- Modify the Publisher Command Transformation policy
- Modify the Subscriber Matching policy

The Driver Filter

The driver filter contains the set of classes and attributes whose updates publish from the SAP system to the Identity Vault, and from the Identity Vault to SAP.

NOTE: To use the default driver configuration, you shouldn’t filter out any of the CommExec, Organizational Role, or Organizational Unit attributes. Also, do not remove the Given Name, Surname, and workforceID attributes from the User class object.

The following table includes some examples of classes and attributes found in the driver filter:

Classes	Attributes
CommExec	Description

Classes	Attributes
Organizational Role	Description
	directReports
	manager
	Role Occupant
Organizational Unit	Description
User	employeeStatus
	Full Name
	Given Name
	homePhone
	Initials
	isManager
	Login Disabled
	manager
	managerWorkforceID
	mobile
	OU
	pager
	Physical Delivery Office Name
	Postal Code
	S
	SA
	Surname
	Telephone Number
	Title
	workforceID

The Schema Mapping Policy

The Schema Mapping policy is referenced by the driver object and applies to both the Subscriber and Publisher channel. The purpose of the Schema Mapping policy is to map schema names (particularly attribute names and class names) between the eDirectory™ and the SAP HR database. Any modification or removal of existing entries in the Schema Mapping policy could destroy the default configuration and policies processing behavior. Adding new attribute mappings is optional. The following attribute mappings are included with the default driver configuration:

eDirectory Class	SAP Class	SAP Description
CommExec	C	Job
Organizational Role	S	Position
Organizational Unit	O	Organization
User	P	Person

The User class is configured to synchronize bidirectionally between SAP and eDirectory. A change made in one system will transfer to the other system. However, changes made to the CommExec, Organizational Role, and Organizational Unit attributes are synchronized from SAP to eDirectory only.

All attributes in the Publisher and Subscriber filters should be mapped unless they are only used for policies processing (for example, Login Disabled.)

The following table includes common attribute mappings for the User class and their descriptions:

eDirectory Attribute	SAP Attribute Description	SAP Attribute
Given name	First Name	P0002:VORNA:none:134:25
Initials	Initials	P0002:INITS:none:74:10
Internet EMail Address	Communication ID/Number (with a mail subtype)	P0105:USRID:MAIL:78:30
NSCP:employeeNumber	Personnel Number	P0001:PERNR:none:0:8
OU	Organizational Unit	P0001:ORGEH:none:125:8
Postal Code	Postal Code (work address subtype)	P0006:PSTLZ:US01:183:10
S	Region (State, Province, or County for the work address subtype)	P0006:STATE:US01:248:3
Surname	Last Name	P0002:NACHN:none:84:25
employeeStatus	Country ISO Code (work subtype)	P0000:STAT2:none:79:1
homeCity	City (permanent address subtype)	P0006:ORTO1:1:133:25
homeFax	Communication Type (permanent address subtype)	P0006:COM01:1:274:20
homePhone	Telephone Number (permanent address subtype)	P0006:TELNR:1:195:14
Title	Position	P0001:PLANS:none:133:8
mobile	Communication ID/Number (cell phone subtype)	P0105:USRID:CELL:78:30
pager	Communication ID/Number (pager subtype)	P0105:USRID:PAGR:78:30
jobCode	Job	P0001:STELL:none:141:8

eDirectory Attribute	SAP Attribute Description	SAP Attribute
personalTitle	Other title	P0002:NAMZU:none:189:15
preferredName	Known As	P0002:RUFNM:none:234:25
workforceID	Personnel Number	P0002:PERNR:none:0:8

The Input Transformation Policy

You modify the Input Transformation policy to implement your specific business rules. The Input Transformation policy is applied to transform the data received from the driver shim.

The policy is applied as the first step of processing an XML document received from the driver shim. The Input Transformation policy converts the syntax of the SAP attributes into the syntax for eDirectory. The Input Transformation policy is implemented as an XSLT style sheet.

The default driver configuration includes templates that complete the following actions:

- Modifies the association for non-Person objects to include Class code.
- Manipulates the OU attribute to contain a name-number syntax.
- Manipulates the Title to contain text data.
- Manipulates the Job Code to contain text data.
- Transforms Postal Address from string syntax to structure syntax.
- Translates telephone numbers from a numerical string into a formatted telephone number.
- Translates employee status from numerical format into either an A (Active) or I (Inactive) status code.
- Adds an employee status code if not present in query replies.

The Output Transformation Policy

You modify the Output Transformation policy to implement your specific business rules. The Output Transformation policy is referenced by the driver object and applies to both the Subscriber channel and to the Publisher channel. The purpose of the Output Transformation policy is to perform any final transformation necessary on XML documents sent to the driver by Identity Manager and returned to the driver by Identity Manager. The Output Transformation policy is implemented as an XSLT style sheet.

The Output Transformation policy reverses the logic of the Input Transformation policy. The default driver configuration includes templates that complete the following actions:

- Transforms Postal Address from structure syntax to string syntax.
- Returns telephone numbers to string format.
- Removes Class code from non-Person object associations.

The Publisher Placement Policy

The Publisher Placement policy is applied to an Add Object event document to determine the placement of the new object in the hierarchical structure of eDirectory. Only the Publisher channel utilizes the Placement policy.

The Placement policy uses the employeeStatus attribute value and the values of driver object placement Global Configuration Values (GCVs) to place objects in specified eDirectory containers.

The Publisher Matching Policy

The Publisher Matching policy is applied to a modify object event document. Matching policies establish links between an existing entry in eDirectory and an existing entry in the SAP system. The Matching policy attempts to find an existing object that matches the object generating the event by the criteria specified in the policy.

The default driver checks for matches based primarily on the workforceID attribute. A secondary rule is provided to attempt matching by Surname and Given Name values.

The Publisher Creation Policy

The Publisher Creation policy is applied when a new object is to be added to eDirectory. The Creation policy is implemented using both Policy Builder and XSLT style sheets.

The default driver configuration has Creation policies for the following:

- Organizational Unit (if a Description attribute is present).
    - Creates a name for the object based on its Description.
    - Creates the OU attribute.
- Organizational Role Object (if a Description attribute is present).
    - Creates a name for the object based on its Description.
    - Creates the CN attribute.
- CommExec Object (if Description attribute is present).
    - Creates a name for the object based on its Description.
    - Creates the CN attribute.
- User Object (the Surname and Given Name are transferred).
    - Generates an object name based on Given Name and Surname.
    - Sets initial password to the user's Surname.

The Publisher Command Transformation Policy

The Publisher Command Transformation policy is used to apply any remaining business logic to event documents received from the driver. The default driver performs the following transformations:

- Creates and maintains User object Manager and Direct Reports organizational relationships.
- Sets Login Disabled attribute based on employee status.
- Maintains proper Group Membership to an Employee or Manager group based on a User's position, employee status, and GCV group name values.
- Handles placement of User objects in Active or Inactive containers based on employee status and GCV user placement values.

5.1.2 Using the Relationship Query

The SAP HR system is a relational database. Individual HR objects, such as the Person object, do not contain all of the information that is typically needed to describe the function of the Person within an organization. Organizational and Position information is contained in different objects that are related to the Person object for a specified period of time. The name of a Position a Person holds, the name of the Organization he belongs to, and the Organizational hierarchy to which a person belongs can only be determined by traversing the various relationships between objects.

The SAP driver has a special capability that allows a query to be made for the object relationships between an SAP object being processed in the Publisher channel and other SAP objects. This information is contained in Infotype 1001 (Object relationships) in the HRMD_A IDoc. (The documentation for the meaning of the various fields of this Infotype can be found on the SAP system using transaction WE60.) Because this relationship information cannot be easily mapped to eDirectory attributes, and because namespace attributes are stripped out of XML documents during various phases of processing, the capability to query for the pseudo-class RELATIONSHIPS was built into the driver.

The Relationship Query uses two different forms described below.

Query 1

This query uses the class identifier of the last object sent by the driver to the engine. In the context of the driver’s default configuration, this query provides accurate results for obtaining relationship data from Position objects as they are processed.

```
<nds dtdversion=”1.0” ndsversion="8.5">
  <input>
     <query class-name="RELATIONSHIPS" event-id="0"
      scope="entry">
        <association>50000354</association>
     </query>
  </input>
</nds>
```

Query 2

This query utilizes the <search-class> element to specify the class of the object from which relationship data is desired. The driver combines the value of the element with the association to identify the proper relationship vector to return. This allows the policies to obtain relationship data from any object in the current IDoc being processed. The new default driver configuration contains queries of this type to provide working examples.

```
<nds dtdversion="1.0" ndsversion="8.5">
  <input>
     <query class-name="RELATIONSHIPS" event-id="0"
      scope="entry">
         <association>50000354</association>
         <search-class class-name="S"/>
     </query>
   </input>
</nds>
```

The driver has been modified to allow the return of all relationship information in a structured <value> format. This has been done to allow the style sheets to utilize any relationship data that is desired for implementing business rules. It is the responsibility of the configuration expert to determine which data is utilized, including time stamp information. The driver returns all requested fields in the 1001 (Relationships) infotype that contain a value. If a field is not populated or present, it is not returned. A sample of a reply to the RELATIONSHIPS Query 2 is presented below:

```
<nds dtdversion="1.0" ndsversion="8.5">
   <source>
<product build="INVALID_BUILD_ID" instance="SAP-HR"
version="1.0.2">Identity Manager Driver for SAP/HR</product>
   <contact>Novell, Inc.</contact>
    </source>
     <output>      <instance class-name="RELATIONSHIPS"
timestamp="20030529" xmlns:sapshim="http://www.novell.com/dirxml/
drivers/SAPShim">
            <association>50000354</association>
            <sapshim:policyAttr attr-name="RELATIONSHIPS">
             <value type="structured">
                <component name="ITXNR">00000000</component>
                <component name="BEGDA">20020225</component>
                <component name="INFTY">1001</component>
                <component name="SEQNR">000</component>
                <component name="ISTAT">1</component>
                <component name="OTYPE">S</component>
                <component name="RELAT">003</component>
                <component name="ENDDA">99991231</component>
                <component name="SCLAS">O</component>
                <component name="PLVAR">01</component>
                <component name="MANDT">001</component>
                <component name="UNAME">NOVADM</component>
                <component name="RSIGN">A</component>
                <component name="SOBID">50000127</component>
                <component name="OBJID">50000354</component>
                <component name="VARYF">O 50000127</component>
                <component name="AEDTM">20020225</components>
             </value>
             <value type="structured">
                <component name="ITXNR">00000000</component>
                <component name="BEGDA">20020225</component>
                <component name="INFTY">1001</component>
                <component name="SEQNR">000</component>
                <component name="ISTAT">1</component>
                <component name="OTYPE">S</component>
                <component name="RELAT">005</component>
                <component name="ENDDA">99991231</component>
                <component name="SCLAS">S</component>
                <component name="PLVAR">01</component>
                <component name="MANDT">001</component>
                <component name="UNAME">NOVADM</component>
                <component name="RSIGN">A</component>
                <component name="SOBID">50000485</component>
                <component name="OBJID">50000354</component>
                <component name="VARYF">S 50000485</component>
             <component name="AEDTM">20020301</component>
           </value>
           <value type="structured">
             <component name="ITXNR">00000000</component>
             <component name="BEGDA">20020225</component>
             <component name="INFTY">1001</component>
             <component name="SEQNR">000</component>
             <component name="ISTAT">1</component>
             <component name="OTYPE">S</component>
             <component name="RELAT">007</component>
             <component name="ENDDA">99991231</component>
             <component name="SCLAS">C</component>
             <component name="PLVAR">01</component>
             <component name="MANDT">001</component>
             <component name="UNAME">NOVADM</component>
             <component name="RSIGN">B</component>
             <component name="SOBID">50000144</component>
             <component name="OBJID">50000354</component>
             <component name="VARYF">C 50000144</component>
             <component name="AEDTM">20020225</component>
             </value>
      </sapshim:policyAttr>
    </instance>
  </output>
</nds>
```

The <read-attr> implementation of the driver RELATIONSHIPS query has been modified as follows:

- The lack of a <read-attr> element implies a request to return all components of each matching relationship value.
- An empty <read-attr/> element specifies that no values will be returned. This is a useless operation that is not recommended.
- <read-attr> elements with attr-name attribute values indicate which specific component values are desired for each matching relationship value.

The <search-attr> functionality of the XDS DTD has been added to the driver RELATIONSHIP query. This enables queries for relationships matching more exacting criteria to reduce the quantity and type of reply data. Multiple <search-attr> values are interpreted as a logical AND of the individual search components. The default Publisher Command Transformation policy has been modified to use the new capabilities of the driver.

The following example is from the set-roles-manager-attr template, used to retrieve the SOBID value from any relationship with an RSIGN value of A and an SCLAS value of S:

Query 3

```
<nds dtdversion="1.0" ndsversion="8.5">
 <input>
    <query class-name="RELATIONSHIPS" event-id="0" scope="entry">
      <association>
         <xsl:value-of select="$newRole-ID"/>
      </association>
      <search-class class-name="S"/>
      <search-attr attr-name="RSIGN">
          <value>A</value>
      </search-attr>
   <search-attr attr-name="SCLAS">
       <value>S</value>
    </search-attr>
    <read-attr attr-name="SOBID"/>
  <query>
 </input>
</nds>
```

Populating the Identity Vault with Organizational Data

In order to populate the Identity Vault with the organizational data, the existing data must be exported from SAP. To export your organization's hierarchical data, perform the following steps before starting the driver:

1. From the SAP client, enter transaction code `PFAL`.
2. Insert the Object Type O for Organization objects.
3. Enter the organizations you want to export to the Identity Vault. You can choose to export one organization, a range of organizations, or all organizations.

   If exporting a range of objects, under the *Parallel Processing* tab on the *HR: ALE Distribution of HR Master Data* screen, select a value of *100* or less at the *Number of Object per Process* prompt. This ensures that driver processing does not consume too much Java heap space.
4. Click *Execute*. Ensure that the status is set to *Passed to Port Okay*.
5. Repeat the above process for Object Type C for Job objects.
6. Repeat the above process for Object Type S for Position objects.

   IMPORTANT: It is important that you export the objects in the order specified above. This ensures that the driver creates the correct relationships when users are imported into the Identity Vault.

Troubleshooting the Driver

6

This section contains potential problems and error codes you might encounter while configuring or using the driver.

6.1 Using the DSTrace Utility

You can troubleshoot the driver using the DSTrace utility. You should configure the utility's options by selecting Edit > Properties > Identity Manager Drivers.

For each event or operation received, the driver returns an XML document containing a status report. If the operation or event is not successful, the status report also contains a reason and a text message describing the error condition. If the result is fatal, the driver shuts down.

After you have configured the DSTrace Utility, you can monitor your system for errors.

6.1.1 Driver Load Errors

If the driver does not load, check DSTrace for the following error messages:

java.lang.ClassNotFoundException:com.novell.nds.dirxml.driver.SAPShim. SAPDriver Shim

This is a fatal error that occurs when `SAPShim.jar` is not installed properly. Ensure that the file is in the proper location for either a local or Remote Loader configuration.

java.lang.ClassNotFoundException:com.novell.nds.dirxml.drivers.SAPShim. SAPDriver Shim

This is a fatal error that occurs when the class name for the `SAPShim.jar` is incorrect. Ensure that the Java class name is set on the Driver Module tab in a local installation and that the -class parameter is set in a Remote Loader configuration.

The proper class name is com.novell.nds.dirxml.driver.SAPShim.SAPDriverShim

6.1.2 Driver Initialization Errors

You might see the following driver initialization errors in the DSTrace utility. An explanation of the error is given along with recommended solutions.

com/sap/mw/jco/JCO

This error occurs when the SAP Java Connector `sapjco.jar` file or the JCO native support libraries are not present or are improperly located.

Make sure the proper platform version of `sapjco.jar` is located in the same directory as `SAPShim.jar`.

Also check the JCO native support libraries to make sure they are present and properly configured. Use the JCO installation instructions for the appropriate platform.

no jRFC12 in java.library.path

This error occurs when the SAP Java Connector (JCO) native RFC12 support library is not present or is located improperly. Make sure the JCO native support libraries are present and configured properly. Use the JCO installation instructions for the appropriate platform.

/usr/jdk1.3.1/lib/sparc/libjRFC12.so:<classpath info>:fatal librfccm.so:open failed: No such file or directory

This error occurs when the SAP Java Connector (JCO) native RFC support library `librfccm.so` is not present or is improperly located. This sample error is from a Solaris system.

Make sure the JCO native support libraries are present and properly configured. Follow the JCO installation instructions for the appropriate platform.

com.novell.nds.dirxml.engine.VRDException

This error occurs when the SAP Java Connector (JCO) components cannot be located. This error generally occurs if the driver or Remote Loader has not been restarted after the JCO has been configured. Restart Novell® eDirectory™ if you are using a local configuration or restart the Remote Loader for a remote configuration.

Error connecting to SAP host

This error occurs when the SAP authentication or connection information is not configured properly. Ensure that the values for Authentication and Driver Parameters are correct for authentication to the SAP host system.

nsap-pub-directory parameter is not a directory

This error occurs when the Publisher IDoc Directory parameter in the Publisher Settings of the Driver Parameters does not specify a valid file system location. Ensure that this parameter specifies the directory on the SAP system configured in the SAP ALE subsystem for IDoc file output.

No connection to remote loader

This error occurs when the Remote Loader connection parameter information is incorrect. Configure the proper connection information for the remote connection to the system where the Remote Loader is running.

Authentication handshake failed, Remote Loader message: “Invalid loader password.”

This error occurs when the Remote Loader password configured on the remote system does not match the Remote Loader password on the Driver object.

Set matching passwords for both remote loaders. In iManager, ensure that both the application password and Remote Loader passwords are set at the same time.

Authentication handshake failed: Received invalid driver object password

This error occurs when the driver password configured on the remote system does not match the Driver object password on the Driver object. To correct this, you should set both Driver object passwords identically.

6.1.3 Attribute Mapping Error

If the Mapping policy Add Dialog contains no data for the APP (application properties of class mappings), the driver can not find the HRMD_A schema metafile.

You should ensure that the metafile directory and Master HR IDoc driver parameters are set to a valid file system location and contain the proper IDoc name. Validate that the metadata file for the configured IDoc type is in the file system location. For example, if Master HR IDoc is set to the default HRMD_A03, ensure that `HRMD_A03.meta` exists in the metafile directory.

6.1.4 Changes in SAP Do Not Generate an IDoc/Change Document

Ensure that the ALE and change pointer processes are configured properly, and that you have properly entered data.

The proper way of inserting or changing data is through using the *Edit* > *Create* or *Edit* > *Change* menus. If an error or a change is entered by overwriting an existing record and saving it, the change document is not created.

6.1.5 The Driver Does Not Recognize IDocs in the Directory

Verify that the driver parameters contain the correct client number and proper IDoc directory.

6.1.6 IDocs Are Not Written to the Directory

You should first test the ALE and IDoc interface. Refer to your SAP documentation for more information.

If the IDoc interface fails:

- Using transaction WE21, ensure that the file port is configured properly. Validate the path to the directory and make sure the Transfer IDoc Immediately option button is selected.
- Using transaction WE20, ensure that the appropriate file port is selected in the Partner Profile. Also, verify that it is on the outbound parameters of the receiving system.

If the IDoc interface succeeds:

- Ensure the change pointers have been configured.
- Ensure that the scheduled processes are not scheduled too closely together. For example, if one job is in process and another job begins, the second job might be cancelled because the first job is still running.

6.1.7 The Driver Does Not Authenticate to SAP

First ensure that you have configured all of the driver parameters and that the proper passwords have been entered.

If you are using the Publisher Channel Only configuration of the driver, make sure you have entered the correct parameters. If you have previously used a Publish and Subscribe driver, make sure that all files have been replaced by the Publish-only files.

If you are running the driver remotely, make sure that the Remote Loader has been started before you start the driver.

6.1.8 JCO Installation and Configuration Errors

For detailed instructions on using the JCO Test utility and analyzing error messages, refer to Section 4.4, "Using the SAP Java Connector Test Utility," on page 38.

6.1.9 Error When Mapping Drives to the IDoc Directory

You might see the following error in DS Trace if the IDoc directory parameter specifies an invalid local file system container or if it specifies a mapped drive on a remote system.

```
*** NDS Trace Utility - BEGIN Logging *** Fri Sep 13 15:45:59 2005

Identity Manager Log Event -------------------
    Driver = \FLIBBLE_TREE\n\Driver Set\SAP-HR
    Channel = publisher
    Status = fatal
    Message = <description>SAP Document Poller initialization failed:
com.novell.nds.dirxml.driver.SAPShim.SAPDocumentPollerInitFailure:
Specified Publisher IDoc Directory is invalid.</description>

*** NDS Trace Utility - END   Logging *** Fri Sep 13 15:46:31 2005
```

This error occurs because the Windows operating system service controls the rights of the local system, not the rights of a user. Thus, the local Windows system does not have rights to access any file resources outside of its own system, including the IDoc directory.

6.1.10 Driver Configured as “Publisher-only” Still Tries to Connect to the SAP System

The driver is designed to use a connection to SAP even when it is configured as a Publisher-only driver. Part of the interface for the Publisher channel is the ability to respond to <query> requests from the Metadirectory engine. These queries can be generated by the engine itself (converting a <modify> event to an <add> event) or can be generated by a policy. If SAP connection parameters are present, the driver attempts to read attributes from the SAP system to respond to those queries. The driver always uses the data in a published IDoc as the primary source for responding to those queries, but if attributes in the Publisher filter are not present in the IDoc, the data obtained in read operations is used to fill in the missing data.

This connection also verifies the validity time stamps of desired infotypes during processing of future-dated event IDocs. This is an extremely critical function that should always be enabled if future-dated processing options are chosen in the driver configuration. Disabling this capability could result in the propagation of old or stale events that have been subsequently overridden.

If you don’t want a connection to the SAP server, you should configure the driver to Disable Publisher Channel Read access. In this situation, the IDoc data being processed is used as a completely authoritative source of reliable data.

Example XML Document Received from the Driver

A

The following example is a typical XML document that has been parsed from HRMD_A number O_200_0000000000008134.

```
<nds dtdversion="1.0" ndsversion="8.5">
     <source>
          <product build="20050916_0956" instance="SAP-HR" version
"1.1">Identity Manager
        Driver for SAP/HR</product>
        <contact>Novell, Inc.</contact>
     </source>
      <input xmlns:sapshim="http://www.novell.com/dirxml/drivers/
SAPShim">
       <modify class-name="P" event-id="O_200_0000000000008134" src-
            dn="00000049" timestamp="20011204-99991231">
              <association>00000049</association>
                  <modify-attr attr-name="P0001:STELL:none:141:8">
                  <remove-all-values/>
                         <add-value>
                        <value timestamp="20011018-
99991231">50000055</value>
                        </add-value>
                  </modify-attr>
                  <modify-attr attr-name="P0000:STAT2:none:79:1">
                   <remove-all-values/>
                        <add-value>
                         <value timestamp="20011018-99991231">3</
value>
                        </add-value>
                  </modify-attr>
              <modify-attr attr-name="P0002:NACHN:none:84:25">
                        <remove-all-values/>
                        <add-value>
                              <value timestamp="19960421-
99991231">Jones</value>
                        </add-value>
                  </modify-attr>
                  <modify-attr attr-name="P0002:VORNA:none:134:25">
                        <remove-all-values/>
                        <add-value>
                          <value timestamp="19960421-99991231">Paul</
value>
                        </add-value>
                  </modify-attr>
                  <modify-attr attr-name="P0006:STRAS:1:103:30">
                         <remove-all-values/>
                         <add-value>
                              <value timestamp="20010101-99991231">123
```

```
Main Street</value>
                         </add-value>
                  </modify-attr>
            </modify>
      </input>
</nds>
```

Some characteristics to note:

- All XML documents received from the SAP HR system are translated into <modify> documents. This translation occurs because it is not possible to determine whether the object described by the document has been modified or is new. Additional modification or translation of the document is accomplished through policies and the Metadirectory engine.
- The <modify> element contains the class-name of the object described (that is, P= Person). The event-id attribute contains the IDoc number from which the data is derived. The src-dn attribute contains the SAP Object ID value. The timestamp attribute contains the date that the IDoc was processed by the driver.
- The <association> element data always contains the SAP Object ID.
- The <modify-attr> element contains the attr-name described in SAP format (Segment:Attribute Name:SubType:Value Offset:Value Length).
- Because multivalue attributes cannot be consistently mapped across systems, the <remove-all-values> element is used prior to all <add-value> tags. This instructs the Metadirectory engine to remove all existing values for the attribute prior to assigning the new value. If this functionality is not desired, one of the XSLT policies may be used to modify the document.
- The <value> element contains a timestamp attribute with the BEGIN VALIDITY-END VALIDITY time stamp of the attribute's data segment (that is, Segment P001 data has a time stamp of 20011018-99991231). This means the data became valid on October 18, 2001 and remains valid to the SAP maximum date. All data segments might have different or future-dated validity time stamps.
- All values are in a string format.

Driver BAPIs

B

The following table contains a list of BAPIs used by the driver.

Table B-1 Driver BAPIs

BAPI Name	Description
BAPI_EMPLOYEE_CHECKEXISTENCE	Used to check for the existence of an employee with a specified Personnel Number (PERNR.) Only used for queries with no <read-attr> elements.
BAPI_EMPLOYEE_ENQUEUE	Used to lock employee records prior to Subscriber modifications.
BAPI_EMPLOYEE_DEQUEUE	Used to unlock employee records after Subscriber modifications.
BAPI_EMPLOYEE_GETDATA	Used to read an employee's Organizational Assignment (Infotype P0001) records. Used during processing of future-dated IDocs to verify that a key with the validity dates of Organizational Assignment and Action (Infotype P0000) instances in the IDoc still exists in the SAP server (stale data checking.)
BAPI_PERSDATA_GETLIST	Used to obtain a list of keys for an employee's Personal Data (Infotype P0002) records. Used during processing of future-dated IDocs to verify that a key with validity dates of Personal Data instances in the IDoc still exists in the SAP server (stale data checking.)
BAPI_PERSDATA_GETDETAIL	Used to read the current data field values of a specified instance of an employee Personal Data record.
BAPI_PERSDATA_CHANGE	Used to modify the current data field values of a specified instance of an employee Personal Data record.
BAPI_ADDRESSEMP_GETLIST	Used to obtain a list of keys for an employee's Address (Infotype P0006) records. Used during processing of future-dated IDocs to verify that a key with the validity dates of Address instances in the IDoc still exists in the SAP server (stale data checking.)
BAPI_ADDRESSMP_GETDETAIL	Used to read the current data field values of a specified instance of an employee Address record.
BAPI_ADDRESSMP_CHANGE	Used to modify the current data field values of a specified instance of an employee Address record.

BAPI Name	Description
BAPI_EMPLCOMM_GETLIST	Used to obtain a list of keys for an employee's Communication (Infotype P0105) records. Used in SAP R/3 versions 4.6 and later. Used during processing of future-dated IDocs to verify that a key with the validity dates of Communication instances in the IDoc still exists in the SAP server (stale data checking.)
BAPI_EMPLCOMM_GETDETAIL	Used to read the current data field value of a specified instance of an employee Communication record. Used in SAP R/3 versions 4.6 and later.
BAPI_EMPLCOMM_CHANGE	Used to modify the current data field value of a specified instance of an employee Communication record. Used in SAP R/3 version 4.6 and later.
BAPI_EMPLCOMM_CREATE	Used to create a new instance of an employee Communication record. Used in SAP R/3 version 4.6 and later.
BAPI_EMPLCOMM_DELIMIT	Used to set the Ending validity period date of a current instance of an employee Communication record. Delimit always sets to the day prior to the current date. If the Starting validity date and the Ending date are the same, the record instance is deleted. Used in SAP R/3 versions 4.6 and later.
BAPI_EMPLCOMM_DELETE	Used to delete the current instance of an employee Communication record. Used in SAP R/3 versions 4.6 and later.
BAPI_HRMASTER_SAVE_REPL_MULT	Used to create or replace the current instance of an employee Communication record. Used in SAP R/3 version 4.5.
BAPI_INTCONTROL_GETLIST	Used to obtain a list of keys for an employee's Internal Control Data (Infotype P0032) records. Used during processing of future-dated IDocs to verify that a key with the validity dates of Internal Control Data in the IDoc still exists (stale data checking.)
BAPI_INTCONTROL_GETDETAIL	Used to read the current data field value of a specified instance of an employee Internal Control Data record.
BAPI_INTCONTROL_CREATE	Used to create a new instance of an employee Internal Control Data record.
BAPI_INTCONTROL_CHANGE	Used to modify the current data field of a specified instance of an employee Internal Control Data record.
BAPI_INTCONTROL_DELIMIT	Used to set the Ending validity period date of a current instance of an employee Internal Control Data record. Delimit always sets to the day prior to the current data. If the Starting validity period date and the Ending date are the same, the record instance is deleted. Used in SAP R/3 versions 4.6 and later.
BAPI_INTCONTROL_DELETE	Used to delete the current instance of an employee Internal Control Data record.

Subscriber Change Modes and Validity Date Modes

C

C.1 Change Mode Notes

- The field name BEGDA indicates the Stating validity date of a value
- The field name ENDDA indicates the Ending validity date of a value.
- The term "active value" indicates a value that has a BEGDA less than or equal to the current date and an ENDDA greater than or equal to the current date.
- Although the driver can handle multiple value synchronization of any particular Communication Subtype on either the Publisher or Subscriber channel, there are issues related to the IDocs generated by SAP value deletion/delimit events that make multiple value synchronization *unadvised* and *unsupported* by the Subscriber channel. It is recommended that only *one* value for each Communication subtype is maintained.
- Because multiple fields are available in the Internal Control Data infotype, a remove-value operation does not result in the deletion of the record instance. The result is the removal of the specified field value form the record instance.
- For Communication values (Infotype P0105), this new functionality is only available in SAP R/3 version 4.6A or later and on all Web Application Server versions. On 4.5 systems (no support prior to 4.5B) the driver uses the BAPI_HRMASTER_SAVE_REPL_MULT function for all operations. <remove-value> and <remove-all-value> operations remove all values of the specified Communication Subtype. <add-value> operations remove all values of the Communication Subtype and create a new value with a BEGDA of *current date* and an ENDDA of 99991231.
- For Internal Control Data values (Infotype P0032), the DELIMIT mode is not available prior to SAP R/3 version 4.6A.

The following sections describe the driver's behavior for each event type and change mode.

C.1.1 <remove-all-values> command

The following operations occur when a <remove-all-values/> element exists in a <modify-attr> command. This is a non-standard XDS Subscriber operation that is generate by a policy.

Delimit Mode

The driver obtains a list of all active values of the specified Infotype record. The driver delimits the validity of each instance (set ENDDA) to *current date -1*. This is the standard SAP delimitation method. If BEGDA is equal to the current date, the value is deleted. This is also standard functionality.

Delete Mode

The driver obtains a list of all active values of the specified Infotype record and deletes each instance.

Change Mode

The driver obtains a list of all active values of the specified Infotype record and deletes each instance.

C.1.2 <remove-value> command without accompanying <add-value>

The following operations occur when a <remove-value> element *without* an accompanying <add-value> element exists in a <modify-attr> command. This is the format of a standard Subscriber value remove XDS event.

Delimit Mode

The driver obtains a list of all active values of the specified Infotype record. The driver tries to match the existing values to the removed value. If a match is found, the driver delimits the validity of the matching value to (current date -1.)

Delete Mode

The driver obtains a list of all active values of the specified Infotype record. The driver tries to match the existing values to the removed value. If a match is found, the driver deletes the matching value.

Change Mode

The driver obtains a list of all active values of the specified Infotype record. The driver tries to match the existing values to the removed value. If a match is found, the driver deletes the matching value.

C.1.3 <remove-value> command with accompanying <add-value>

The following operations occur when a <remove-value> element *with* an accompanying <add-value> element exists in a <modify-attr> command. This is the format of a standard Subscriber value change XDS format.

Delimit Mode

The driver obtains a list of all active values of the specified Infotype record. The driver tries to match the existing values to the removed value. If a match is found, the driver delimits the validity of the matching value to (current date -1.) If the added value is not already an active value, the added value is created.

Delete Mode

The driver obtains a list of all active values of the specified Infotype record. The driver tries to match the existing values to the removed value. If a match is found, the driver deletes the matching value. If the added value is not already an active value, the added value is created.

Change Mode

The driver obtains a list of all active values of the specified Infotype record. The driver tries to match the existing values to the removed value. If a match is found, the driver changes the matching value to the added value. If a match is not found, the driver deletes the removed value. If the added value is not already an active value, the added value is created.

C.1.4 <add-value> command without prior <remove-value>

If the added value is not already an active value, the driver creates the added Infotype for all modes.

NOTE: This new functionality is only available on SAP R/3 version 4.6A or later and on all Web Application Server versions. On 4.5 systems (no support prior to 4.5B) the driver uses the BAPI_HRMASTER_SAVE_REPL_MULT function for all operations. <remove-value> and <remove-all-value> operations remove all values of the specified Communication Subtype. <add-value> operations remove all values of the Communication Subtype and create a new value with a BEGDA of (current date) and an ENDDA of 99991231.

C.2 Validity Date Modes

The driver contains configuration parameters that allow an administrator to specify how validity begin dates (BEGDA) and validity end dates (ENDDA) are set when new Communication or Internal Control Data values are created for an Employee object. The new parameters are *Communication Validity Date Mode* and *Internal Data Validity Date Mode*. They allow two modes of operation:

Current Date Mode

This mode configures the driver to set validity dates in the same manner employed by all other previous versions of the driver. The driver sets the current date for the validity begin field (BEGDA) and sets the maximum SAP date for the validity end field (ENDDA).

Default Mode

This mode configures the driver to not set any BEGDA and ENDDA field values. When these values are not set, the default validity dating scheme of the SAP server is used to set these two field values. Standard SAP configuration sets the BEGDA value to the date that the Employee record was created and sets the ENDDA value to the maximum SAP date value.