



# BEA eLink Information Integrator

## Installation and Administration Guide

BEA eLink Information Integrator Version 1.0  
Document Edition 1.0  
April 2000

## Copyright

Copyright © 2000, 1996-1999 BEA Systems, Inc., or its suppliers, as applicable. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software-Restricted Rights Clause at FAR 52.227-19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, subparagraph (d) of the Commercial Computer Software--Licensing clause at NASA FAR supplement 16-52.227-86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, ObjectBroker, TOP END, and Tuxedo are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Connect, BEA Manager, BEA MessageQ, BEA Jolt, M3, eSolutions, eLink, WebLogic, and WebLogic Enterprise are trademarks of BEA Systems, Inc.

All other company names may be trademarks of the respective companies with which they are associated.

## **BEA eLink Information Integrator Installation and Administration Guide**

<b>Document Edition</b>	<b>Part Number</b>	<b>Date</b>	<b>Software Version</b>
1.0	870-001039-001	April 2000	BEA eLink Information Integrator 1.0

---

# Contents

## About This Document

What You Need to Know .....	vii
e-docs Web Site .....	viii
How to Print the Document .....	viii
Document Conventions .....	ix

## 1. Installing BEA eLink Information Integrator on Windows NT

Preparing for a New NT Installation .....	1-1
Required Disk Space .....	1-2
Setting Up the Environment .....	1-2
Oracle .....	1-2
SQL Server .....	1-2
Collecting Information .....	1-3
Oracle .....	1-3
MS SQL Server .....	1-3
Installing on Windows NT .....	1-4
Server Installation .....	1-4
Developer Installation .....	1-12
Preparing the Database for Schema Installation .....	1-18
Creating Tablespaces .....	1-18
Installing the Database Schema .....	1-19
Editing the Database Connection File .....	1-20
Uninstalling BEA eLink Information Integrator .....	1-21

## 2. Installing BEA eLink Information Integrator on a Unix Platform

Preparing for a Unix Installation .....	2-2
---	-----

---

Required Disk Space .....	2-2
Setting Up the Environment .....	2-2
Collecting Information .....	2-3
Installing on Unix Platforms.....	2-3
Distribution Libraries and Executables .....	2-7
Oracle Environment.....	2-8
SQL Server Environment .....	2-12
Preparing the Database Schema for Installation.....	2-15
Creating Tablespaces.....	2-15
Oracle .....	2-16
Installing the Database Schema.....	2-16

### **3. Configuring BEA eLink Information Integrator**

Defining the Server.....	3-1
Configuring BEA eLink Information Integrator .....	3-2
Defining the SERVER Section.....	3-3
Required Parameters .....	3-3
Optional Parameters .....	3-4
Defining the SERVICE Section .....	3-4
Required Parameters .....	3-4
Optional Parameters .....	3-5
Defining the IIRESET Service .....	3-5
Required Parameters .....	3-5
Optional Parameters .....	3-6
Sample eLink Information Integrator Configuration File .....	3-6
The Database Session Configuration File .....	3-7
Required Parameters.....	3-7
Optional Parameters .....	3-8
Sample Database Session Configuration File.....	3-8

### **4. Working with Formatter**

Formatter Overview.....	4-1
Fields and Input Controls .....	4-2
Output Controls .....	4-3
Formats .....	4-3

---

Importing and Exporting Formats .....	4-4
NNFie .....	4-4
NNFie Import Syntax.....	4-6
NNFie Export Syntax.....	4-7
Conflict Resolution .....	4-8
Testing Formats.....	4-11
APITEST .....	4-11
Using apitest.....	4-11
MSGTEST.....	4-12
Using msgtest.....	4-12
Configuration File.....	4-13

## 5. Working with Rules

Rules Overview .....	5-1
Application Groups .....	5-2
Message Types .....	5-2
Rule Names .....	5-2
Expressions, Arguments, Boolean, and Rules Operators.....	5-3
Subscriptions, Actions, and Options .....	5-3
Defining Users in Rules.....	5-3
Oracle .....	5-4
Creating Users .....	5-4
Granting Roles to Users .....	5-4
SQL Server .....	5-5
Creating Login Accounts .....	5-5
Assigning Users to a Database .....	5-6
Defining User Groups .....	5-6
Importing and Exporting Rules .....	5-7
NNRie .....	5-7
NNRie Import Syntax .....	5-10
NNRie Export Syntax .....	5-11
Command Line Functions.....	5-12
Conflict Resolution .....	5-14

---

## **A. Error Messages**

NNFie Error Messages .....	A-1
Information Integrator Server Error Messages .....	A-5

## **B. Calculating Sizing Data**

Formatter Space Requirements .....	B-1
Oracle .....	B-2
Microsoft SQL Server .....	B-2
Rules Space Requirements .....	B-3
Oracle .....	B-4
Microsoft SQL Server .....	B-4

## **C. Changing Database Sort Order**

Microsoft SQL Server .....	C-1
New Install of Microsoft SQL Server .....	C-1
Upgrade Microsoft SQL Server from Version 4.2 and Earlier .....	C-2
Established Microsoft SQL Server .....	C-2

---

# About This Document

This document provides instructions for installing BEA eLink Information Integrator on Windows NT and Unix platforms and configuring the Information Integrator environment.

This document covers the following topics:

- Installing BEA eLink Information Integrator on Windows NT
- Installing BEA eLink Information Integrator on a Unix Platform
- Configuring BEA eLink Information Integrator
- Working with Formatter
- Working with Rules
- Error Messages
- Calculating Sizing Data
- Changing Database Sort Order

## What You Need to Know

This document is intended for system administrators and application programmers who will configure the Information Integrator and use it to execute information transfers.

---

# e-docs Web Site

BEA product documentation is available on the BEA corporate Web site. From the BEA Home page, click on Product Documentation or go directly to the “e-docs” Product Documentation page at <http://e-docs.bea.com>.

## How to Print the Document

You can print a copy of this document from a Web browser, one file at a time, by using the File—>Print option on your Web browser.

A PDF version of this document is available on the BEA eLink Information Integrator documentation Home page on the e-docs Web site (and also on the documentation CD). You can open the PDF in Adobe Acrobat Reader and print the entire document (or a portion of it) in book format. To access the PDFs, open the BEA eLink Information Integrator documentation Home page, click the PDF files button and select the document you want to print.

If you do not have the Adobe Acrobat Reader, you can get it for free from the Adobe Web site at <http://www.adobe.com/>.

## Related Information

The following BEA publications are also available:

- ◆ *TUXEDO System 6 Reference Manual*
- ◆ *TUXEDO System 6 Programmer’s Guide, Volumes 1 and 2*

## Contact Us!

Your feedback on the BEA Information Integrator documentation is important to us. Send us e-mail at **docsupport@bea.com** if you have questions or comments. Your comments will be reviewed directly by the BEA professionals who create and update the BEA eLink Information Integrator documentation.

In your e-mail message, please indicate that you are using the documentation for the BEA BEA eLink Information Integrator 1.0 release.

If you have any questions about this version of the Information Integrator, or if you have problems installing and running the Information Integrator, contact BEA Customer Support through BEA WebSupport at **www.bea.com**. You can also contact Customer Support by using the contact information provided on the Customer Support Card, which is included in the product package.

When contacting Customer Support, be prepared to provide the following information:

- Your name, e-mail address, phone number, and fax number
- Your company name and company address
- Your machine type and authorization codes
- The name and version of the product you are using
- A description of the problem and the content of pertinent error messages

## Document Conventions

The following documentation conventions are used throughout this document:

---

Item	Examples
Variable names	<p>Variable names represent information you must supply or output information that can change; they are intended to be replaced by actual names. Variable names are displayed in italics and can include hyphens or underscores. The following are examples of variable names in text:</p> <p><i>error_file_name</i></p> <p>The <i>when-return</i> value...</p>
User input and screen output	<p>For screen displays and other examples of input and output, user input appears as in the first of the following lines; system output appears as in the second through fourth lines:</p> <pre data-bbox="602 646 1056 756"> <b>dir c:\accounting\data</b> Volume in drive C is WIN_NT_1 Volume Serial Number is 1234-5678 Directory of C:\BEADIR\DATA </pre>
Syntax	<p>Code samples can include the following elements:</p> <ul style="list-style-type: none"> <li>■ Variable names can include hyphens or underscores (e.g., <i>error_file_name</i>)</li> <li>■ Optional items are enclosed in square brackets: [ ]. If you include an optional item, do not code the square brackets.</li> <li>■ A required element for which alternatives exist is enclosed in braces { }. The alternatives are separated by the pipe (vertical bar) character:  . You must include only one of the alternatives for that element. Do not code the braces or pipe character.</li> <li>■ An ellipsis ( ... ) indicates that the preceding element can be repeated as necessary.</li> </ul>
Omitted code	<p>An ellipsis ( ... ) is used in examples to indicate that code that is not pertinent to the discussion is omitted. The ellipsis can be horizontal or vertical.</p>
Environment variables	<p>Environment variables are formatted in an uppercase font.</p> <pre data-bbox="602 1308 834 1333"> ENVFILE=\${APPDIR} </pre>
Key names	<p>Key names are presented in boldface type.</p> <p>Press <b>Enter</b> to continue.</p>

---

<b>Item</b>	<b>Examples</b>
Literals	Literals are formatted in a monospace font. <code>class extendSample</code>
Window items	Window items are presented in boldface type. Window items can be window titles, button labels, text edit box names or other parts of the window. Type your password in the <b>Logon window</b> . Select <b>Export</b> to make the service available to the client.



# 1 Installing BEA eLink Information Integrator on Windows NT

This chapter discusses the following topics:

- Preparing for a New NT Installation
- Installing on Windows NT
- Preparing the Database for Schema Installation
- Installing the Database Schema
- Editing the Database Connection File
- Uninstalling BEA eLink Information Integrator

## Preparing for a New NT Installation

This section describes the steps you should take before you run the Information Integrator installation procedure on Windows NT.

## Required Disk Space

The installation disk space requirements depend on which components you install and how much working space you need. Your working space depends on the number of queues, the number and size of the messages on the queues, whether the messages are persistent, and how many formats and rules you plan to build. Archiving capacity on disk, tape, or other media is also required.

## Setting Up the Environment

The following items must be verified prior to installing BEA eLink Information Integrator. Otherwise, the installation will not complete successfully.

### Oracle

Verify the following:

- You can connect to an appropriately resourced Oracle database that stores Information Integrator data, either directly or through an Oracle client.
- You know the Oracle SYS account information.
- There is sufficient disk space for your calculated needs.
- The Oracle utility program **plus33** (for Oracle 7) or **plus80** (for Oracle 8) is in the execution path for the user doing the install.
- Be sure the **PATH** environment variable includes the product bin directory and the database bin directory.

### SQL Server

Verify the following:

- You can connect to an appropriately resourced SQL Server database that stores Information Integrator data, either directly or through a SQL Server client.
- You are a database owner (or have the account information for the owner of the database).

- That the target database is the default database.
- There is sufficient disk space for your calculated needs.
- The Microsoft SQL Server utility program **isql** is in the execution path for the user doing the install.
- Be sure the **PATH** environment variable includes the product bin directory and the database bin directory.

## Collecting Information

Before beginning the installation, know the drive letter of the CD-ROM drive from which you will run the installation and the information for the appropriate operating system in the following sections:

### Oracle

- SYS userid
- password for SYS userid
- TNS service name to be used in connecting to the database

### MS SQL Server

- database name
- server name
- username
- password

## Installing on Windows NT

There are two options for installing BEA eLink Information Integrator on a Windows NT machine:

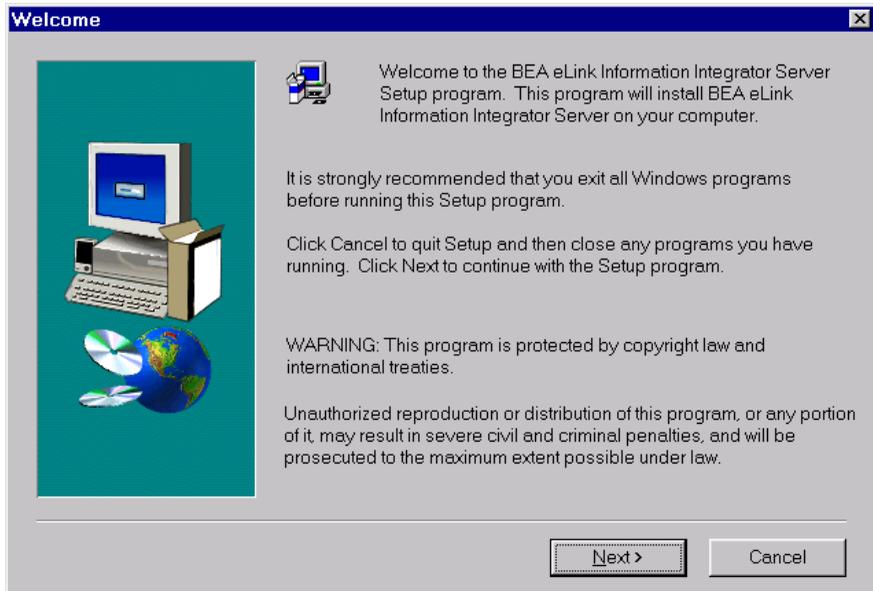
- **Developer Installation** — Installs the graphical user interface for the utilities available with eLink Information Integrator. Use this installation to enter information into and extract information from the eLink Information Integrator database.
- **Server Installation** — Installs the runtime utilities available with eLink Information Integrator. Use this installation to perform operations against the database information built with the Developer Installation.

## Server Installation

Perform the following steps to install the BEA eLink Information Integrator Server software on a Windows NT system:

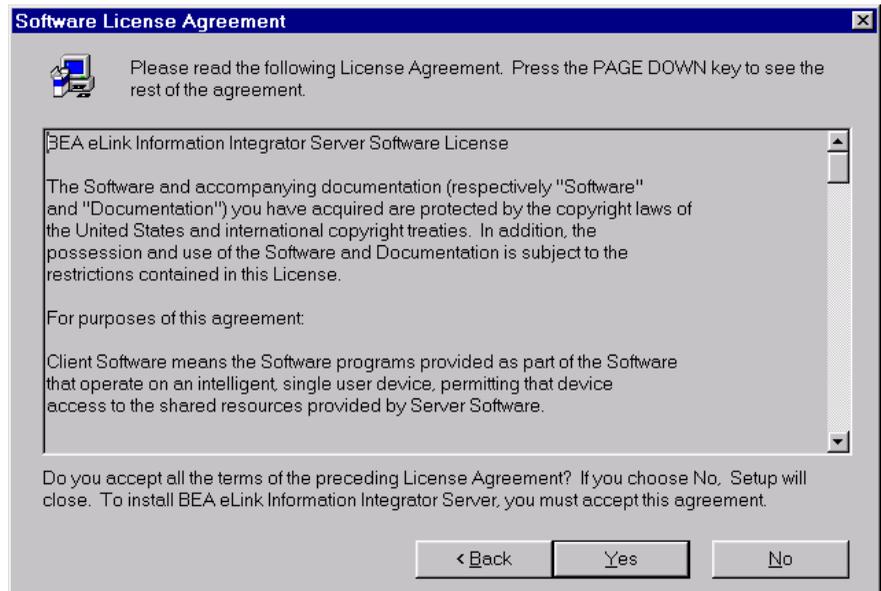
1. Insert the BEA eLink Information Integrator Server CD-ROM. Click **Run** from the **Start menu**. The **Run** window displays. Click on the **Browse** button to select the CD-ROM drive. Select the `winnt` directory and select the `Setup.exe` program. Click **OK** to run the executable and begin the installation. The following **Welcome** screen displays. Click **Next** to continue with the installation.

Figure 1-1 Welcome



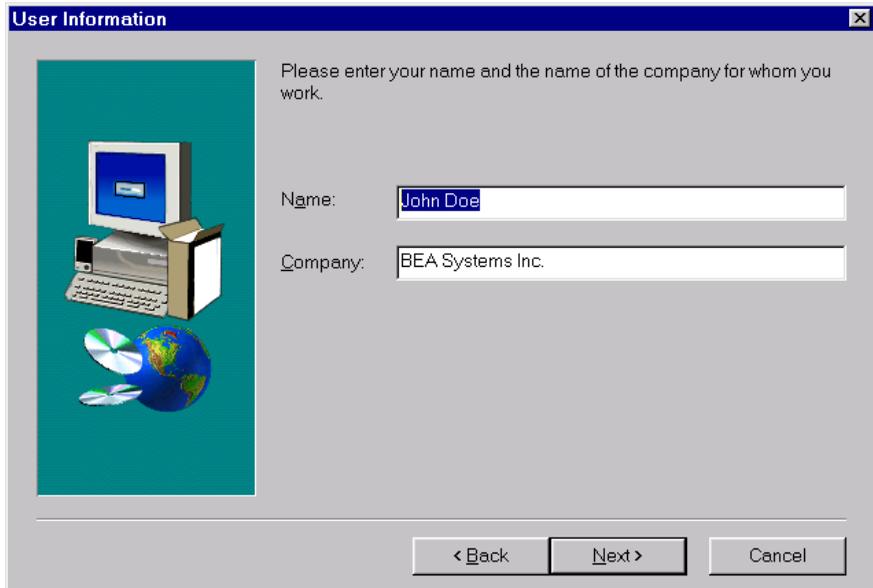
2. The **License Agreement** screen displays after the **Welcome** screen. Click **Yes** to accept the license agreement and continue with the installation.

**Figure 1-2 License Agreement**



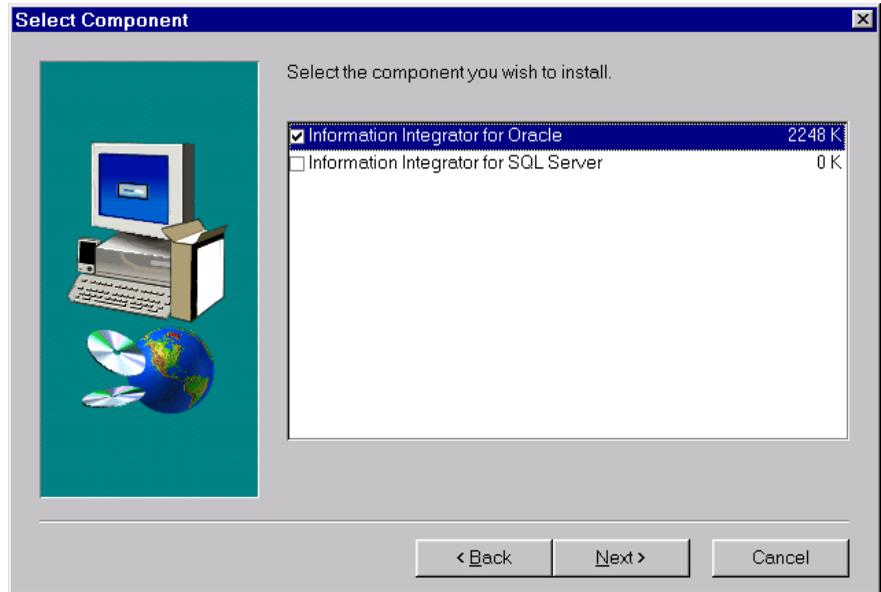
3. The **User Information** screen displays after the **License Agreement** screen. Enter your name in the **Name** field. Enter the name of your company in the **Company** field. Click **Next** to continue with the installation.

**Figure 1-3 User Information**



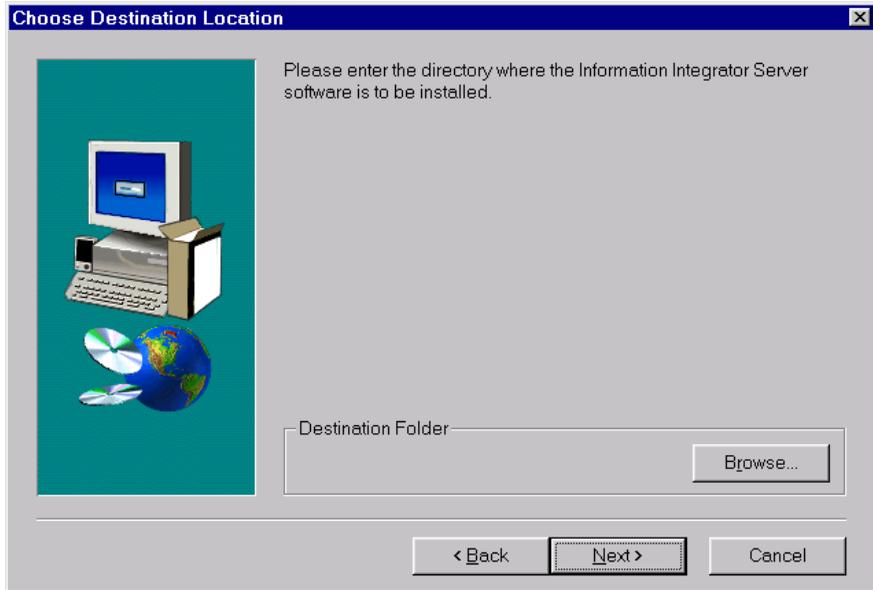
4. The **Select Components** screen displays next. Choose either **Information Integrator for Oracle** or **Information Integrator for SQL Server** and click **Next** to continue.

**Figure 1-4 Select Components**



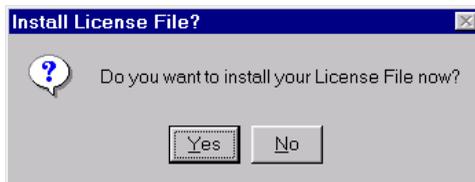
5. The **Choose Destination Location** screen displays next. Enter the directory path where you want to install eLink Information Integrator, or click **Browse** to select the desired drive and directory.

**Figure 1-5 Choose Destination Location**



6. The **Install License File?** option screen displays next. You may select **Yes** to install your BEA Software License File, or you may select **No** to bypass this step and continue installing the eLink Information Integrator software. If you select **Yes**, continue with Step 7. If you select **No**, continue with Step 9.

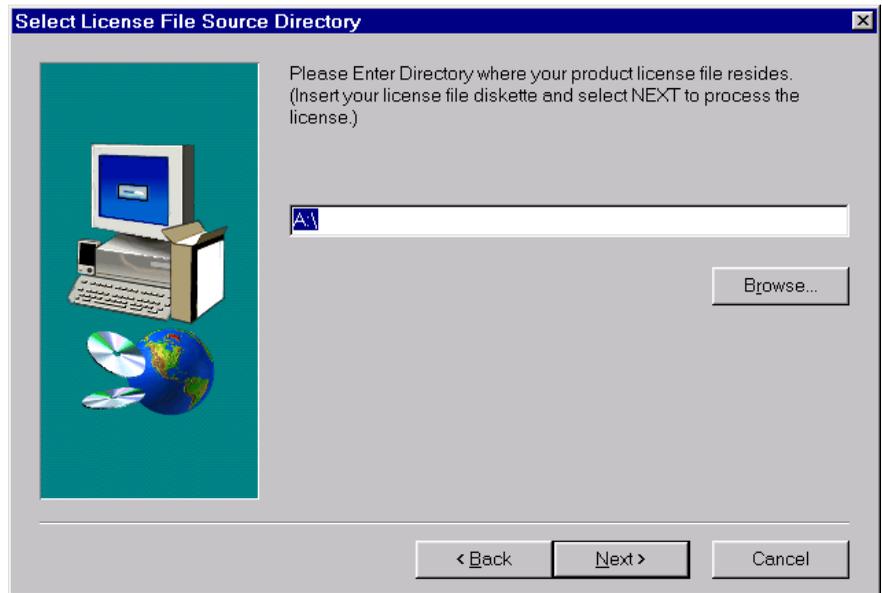
**Figure 1-6 Install License File?**



7. The **Select License File Source Directory** screen displays. Enter the directory path where your license file resides in the field. You can browse and click directories by clicking the **Browse** button. Typically, the license file is installed in the `tuxedo/udataobj` directory.

If you entered a valid directory path, click **Next** to continue with the installation. Go to Step 9. If you entered an invalid directory path, go to Step 8.

**Figure 1-7 Select License File Source Directory**



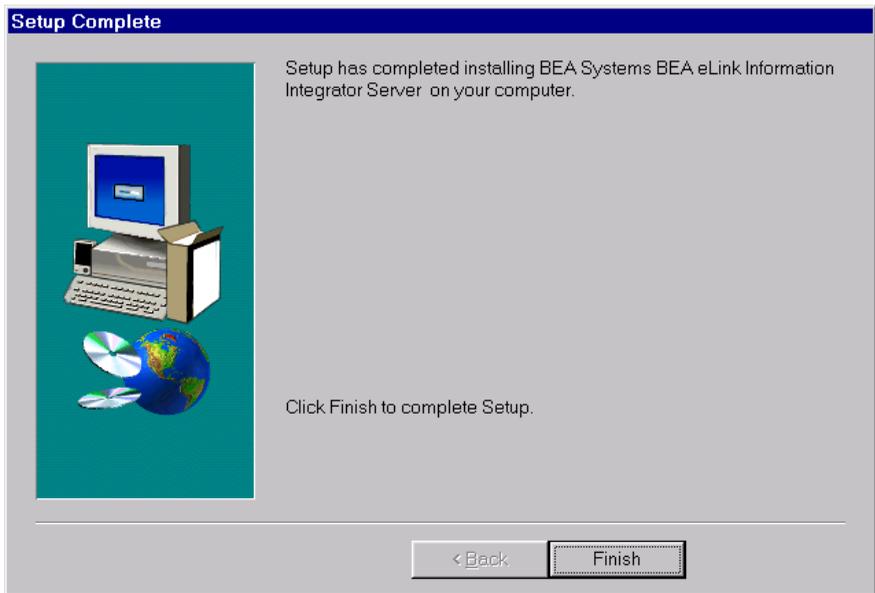
8. If you do not enter a valid directory path for your license file, the installation software generates an error message dialog box. You can select **Yes** to enter a valid directory path, or you can select **No** to continue with the installation. If you select **No**, the installation software automatically searches for the TUXEDO software. If it finds TUXEDO installed, the installation software completes the process. If TUXEDO is not found, the installation software aborts the process.

**Note:** If you select **No**, the installation continues but an error is generated in the `ulog.mm/dd/yy` file indicating that the product is unlicensed. Please refer to the “Using the License Key” section of the *BEA eLink Information Integrator Release Notes* for instructions on using the license file.

Once you have entered a valid directory path, click **Next** to continue with the installation. Go to Step 9.

9. A progress bar displays showing the status of the installation.
10. The **Setup Complete** screen displays notifying you that the BEA eLink Information Integrator Server software is installed on your system. Click **Finish** to complete the Setup process.

**Figure 1-8 Setup Complete**

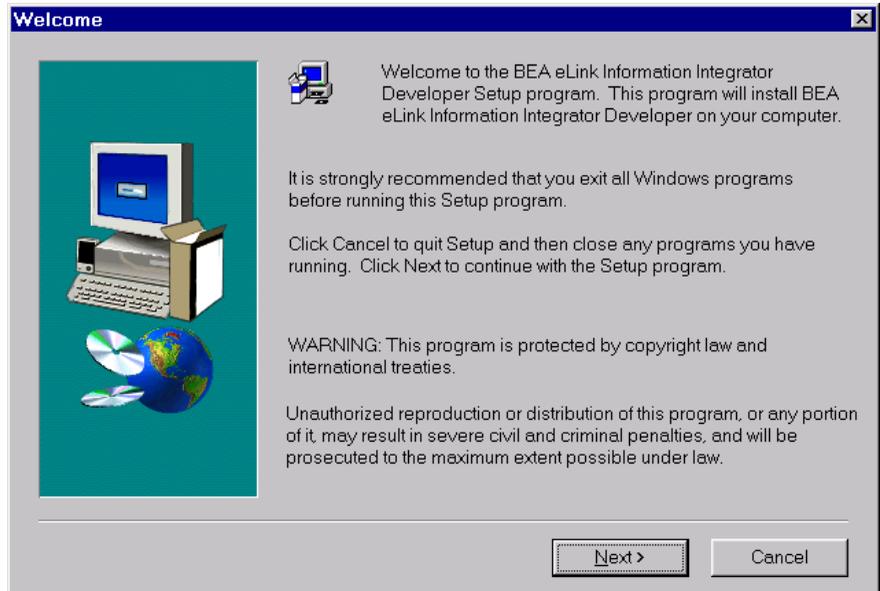


## Developer Installation

Perform the following steps to install the BEA eLink Information Integrator Developer on a Windows NT system:

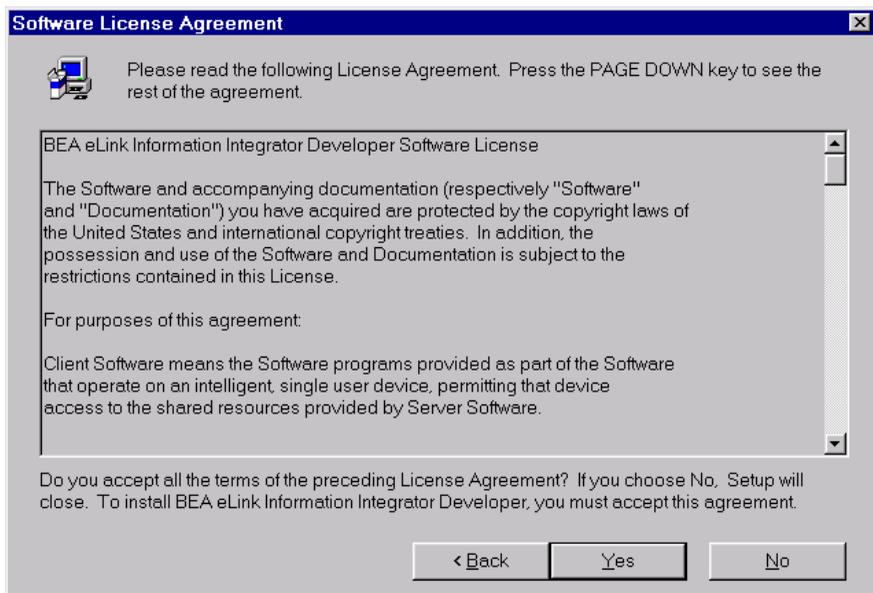
1. Insert the BEA eLink Information Integrator Developer CD-ROM and click on the **Run** option from the **Start menu**. The **Run** window displays. Click on the **Browse** button to select the CD-ROM drive. Select the `winnt` directory and select the `Setup.exe` program. Click **OK** to run the executable and begin the installation. The following **Welcome** screen displays. Click **Next** to continue with the installation.

**Figure 1-9 Welcome**



2. The **License Agreement** screen displays after the **Welcome** screen. Click **Yes** to accept the license agreement and continue with the installation.

**Figure 1-10 License Agreement**

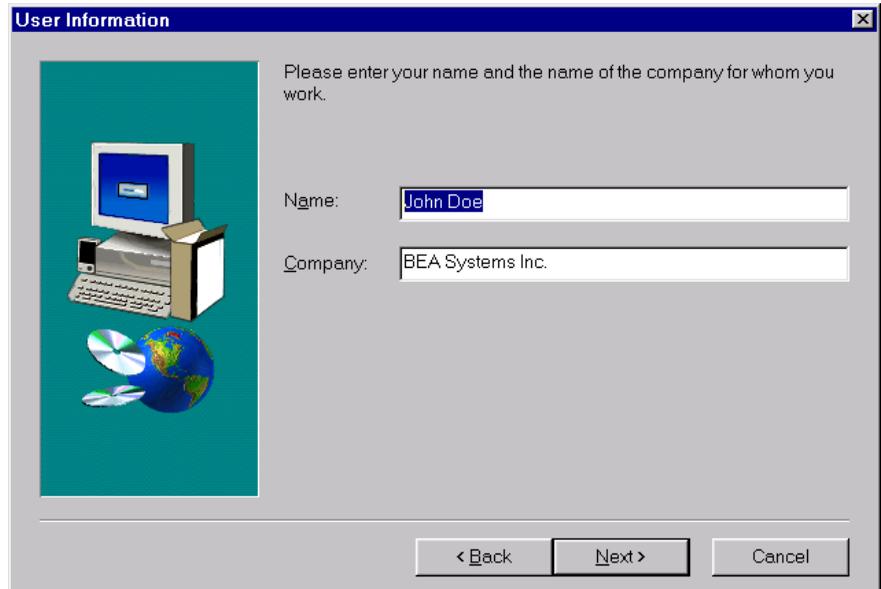


# 1 *Installing BEA eLink Information Integrator on Windows NT*

---

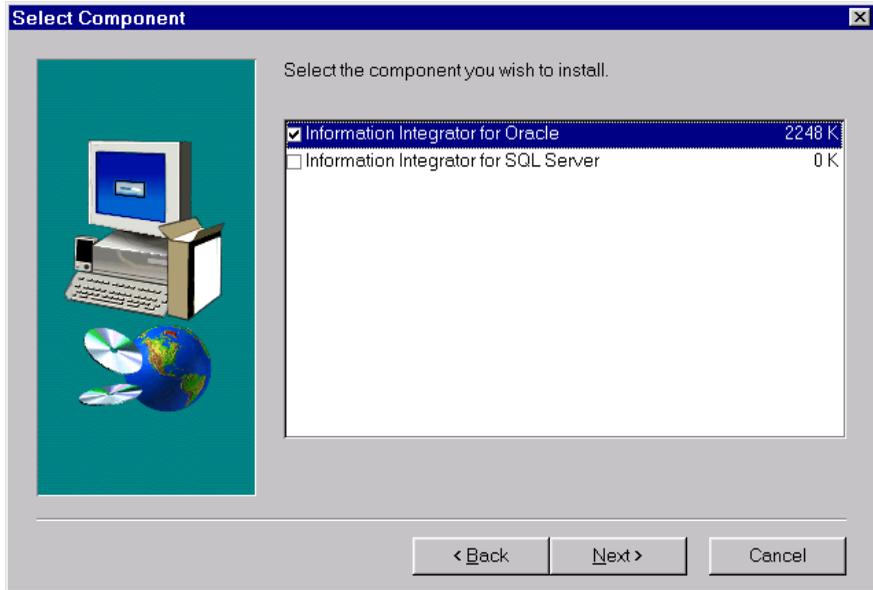
3. The **User Information** screen displays after the **License Agreement** screen. Enter your name in the **Name** field. Enter the name of your company in the **Company** field. Click **Next** to continue with the installation.

**Figure 1-11 User Information**



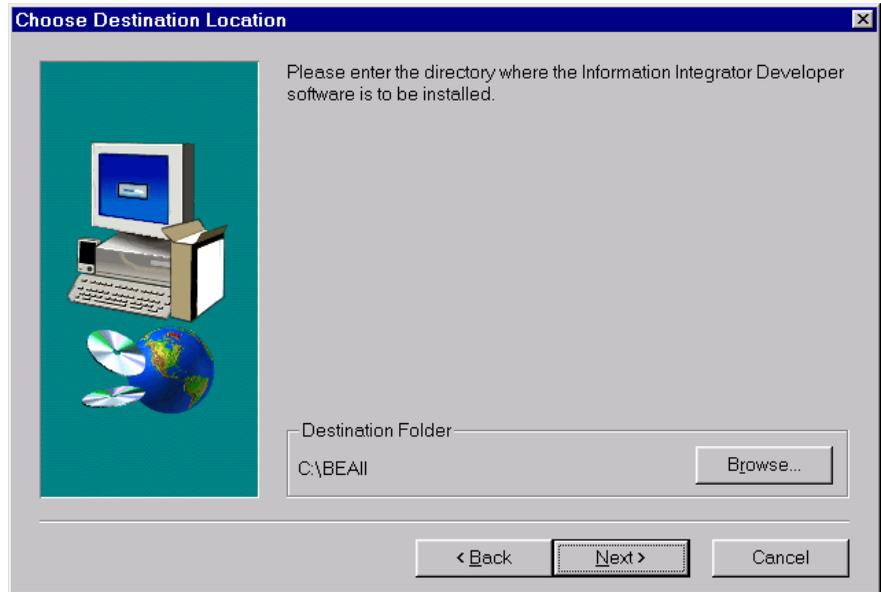
4. The **Select Components** screen displays next. Choose either **Information Integrator for Oracle** or **Information Integrator for SQL Server** and click **Next** to continue.

**Figure 1-12 Select Components**



5. The **Choose Destination Location** screen displays next. Enter the directory path where you want to install eLink Information Integrator, or click **Browse** to select the desired drive and directory.

**Figure 1-13 Choose Destination Location**



6. A progress bar displays showing the status of the installation.

7. The **Setup Complete** screen displays notifying you that the BEA eLink Information Integrator Developer software is installed on your system. Click **Finish** to complete the Setup process.

**Figure 1-14 Setup Complete**



# Preparing the Database for Schema Installation

This section explains the steps necessary to prepare the eLink Information Integrator database for schema installation.

## Creating Tablespaces

When working with BEA eLink Information Integrator in an Oracle environment, you must have eLink Information Integrator tablespaces created before you can install the database schema.

**Note:** You do not have to create tablespaces with Microsoft SQL Server.

The size of your tablespaces depends on the numbers of Rules and Formats used at your site. See “Calculating Sizing Data” for specific sizing information.

You may want to place the tablespaces on different physical disks to balance I/O to avoid disk-access bottlenecks. You should separate data tablespaces and index segments by placing them on different disks and/or controllers. This optimizes index and data access parallelism.

To create Oracle tablespaces:

1. Create a dedicated Oracle instance where the Information Integrator database resides.
2. Create the following tablespaces in the Oracle database:

<b>Table</b>	<b>Minimum Size</b>
TOOLS	1 MB
TEMP	10 MB
FORMATTER_DATA	20 MB

Table	Minimum Size
FORMATTER_INDEX	20 MB
RULES_DATA	20 MB
RULES_INDEX	20 MB

**Note:** For information on creating an Oracle database, refer to the Oracle installation documentation.

## Installing the Database Schema

The `inst_db` script creates the necessary tables and stored procedures in the Information Integrator database. The script sends the commands from the files in the `install.sql` directory.

To install the database schema, make sure you are logged in with the SYS user id and password; then, perform the following steps:

**Note:** Make sure your Oracle environment is set up properly before performing these steps. See “Setting Up the Environment” for more information.

1. At the command line prompt, change to the **install.sql** directory.

Example:

```
cd \InfoInt\install.sql
```

2. To build the Information Integrator schema, type one of the following:

- Oracle

```
set SQLPLUS=<plus name>
inst_db.cmd SYS <SYS password> <Service Name>
```

For Oracle 7, <plus name> is **plus33**. For Oracle 8, <plus name> is **plus80**.

- SQL Server

```
inst_db.cmd <username> <password> <servername> <dbname>
```

If you have not defined a password for your database username, use two single quotes to specify the password. For example:

```
inst_db.cmd sa '' mycomputer
```

3. As the script runs, answer the prompts and look for errors.
4. When the script completes the instantiation, a verification message appears.
5. For installation details, look at the **inst\_db.log** file located in the **c:\temp** directory.

In the `inst_db.log` file, the error “table or view does not exist” does not indicate a problem with database instantiation. The database successfully instantiated if this is the only error you receive.

## Editing the Database Connection File

Some executables connect to the database using the database connection file `sqlsvses.cfg`. This file contains entries for DBMS sessions that detail the server name, user ID, password, and database name that a particular session uses. Executables search the `sqlsvses.cfg` file for a given session name and attempt to connect to the database (for example, `msgttest` searches for `new_format_demo`).

A sample `sqlsvses.cfg` file that is commented out is provided in the **bin** directory. Uncomment the section that applies to your DBMS type. You must edit the sample file with your site-specific information. This file enables certain Information Integrator executables to connect to the database. For more information, refer to “The Database Session Configuration File”.

To edit the database connection file:

1. Change to the **bin** directory.
2. In the `bin` directory, locate the text file **sqlsvses.cfg**.

3. In the `sqlsvses.cfg` file, edit the following:

#### Oracle

```
<sessionname>:<servicename>:<username>:<password>:
```

Example:

```
new_format_demo:dodge:beauser:beapwd:  
rules:dodge:beauser:beapwd:  
nmfie:dodge:beauser:beapwd:  
nnrmie:dodge:beauser:beapwd:
```

#### SQL Server

```
<sessionname>:<servername>:<username>:<password>:<dbname>
```

Example:

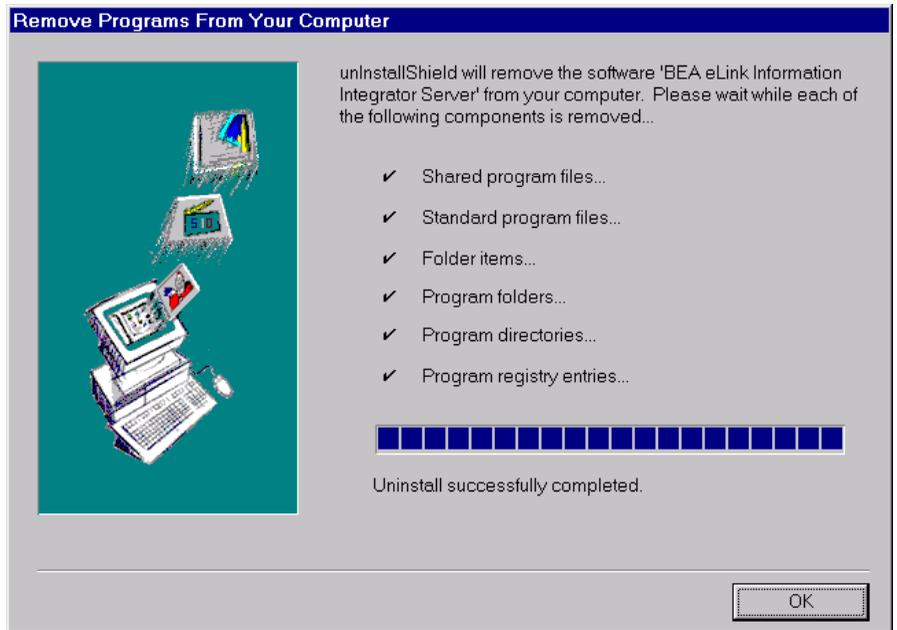
```
new_format_demo:dodgebeauser:beapwd:TESTDATABASE  
rules:dodge:beauser:beapwd:TESTDATABASE
```

# Uninstalling BEA eLink Information Integrator

Perform the following steps to uninstall the eLink Information Integrator software on a Windows NT system:

1. Access the **Control Panel** window from the **Start>Settings>Control Panel** menu option.
2. Double-click on the **Add/Remove Programs** option from the Control panel listings to access the **Add/Remove Programs Properties** window.
3. In the **Add/Remove Programs Properties** window, select **BEA eLink Information Integrator Server** or **BEA eLink Information Integrator Developer** from the program list and click the **Add/Remove** button.
4. The uninstall process for eLink Information Integrator begins. The **Remove Programs From Your System** screen displays. Click **OK** to complete the uninstall process.

**Figure 1-15 Remove Programs From Your Computer**



# 2 Installing BEA eLink Information Integrator on a Unix Platform

This chapter describes the steps required to install BEA eLink Information Integrator on Unix machines. The following topics are discussed:

- Preparing for a Unix Installation
- Installing on Unix Platforms
- Distribution Libraries and Executables
- Preparing the Database Schema for Installation
- Installing the Database Schema

# Preparing for a Unix Installation

This section describes the steps necessary before installing Information Integrator on a Unix machine.

## Required Disk Space

The installation disk space requirements depend on which components you install and how much working space you need. Your working space depends on the number of queues, the number and size of the messages on the queues, whether the messages are persistent, and how many formats and rules you plan to build. Archiving capacity on disk, tape, or other media is also required.

## Setting Up the Environment

On the Unix machine, verify the following:

- The Oracle utility program `sqlplus` (for AIX and Solaris installations), `plus33` (for HP-UX installations using Oracle 7), or `plus80` (for HP-UX installations using Oracle 8) is in the execution path for the user doing the install.
- You can connect to the database that stores Information Integrator data.
- An Oracle database exists with the appropriate resources and table spaces.
- The `ORACLE_HOME` environment variable is set to the location of the database home.
- The environment variable that defines where the library for the product and database reside (`LIBPATH` for AIX, `SHLIB_PATH` for HP-UX, and `LD_LIBRARY_PATH` for Solaris) points to the product lib directory and the database lib directory.
- Be sure the `PATH` environment variable includes the product bin directory and the database bin directory.
- You know the Oracle `SYS` account information.

## Collecting Information

Before beginning the installation, know the drive letter or path to the mounted CD-ROM device from which you will run the installation, the root password for the Unix server, and the information for the appropriate operating system as listed in the following sections:

- SYS userid
- password for SYS userid
- service name

## Installing on Unix Platforms

To install the BEA eLink Information Integrator software, you run the `install.sh` script. This script installs all the necessary software components.

Perform the following steps to install the eLink Adapter software on a supported Unix platform:

1. Log on as root to install the Information Integrator software.

```
$ su -  
Password:
```

2. Access the CD-ROM device.

```
# ls -l /dev/cdrom
```

```
total 0
```

```
brw-rw-rw-    1 root    sys      22,    0 April 14 10:55 clb0t010
```

3. Mount the CD-ROM.

```
# mount -r -F cdfs /dev/cdrom/clb0t010 /mnt
```

4. Change the directory to your CD-ROM device.

```
# cd /mnt
```

5. List the CD-ROM contents.

```
# ls
install.sh hp sun5x aix43
```

6. Execute the installation script.

```
# sh ./install.sh
```

7. The installation script runs and prompts you for responses.

### Listing 2-1 Install.sh Example

---

```
01) hp/hpux1020      02) hp/hpux11      03) ibm/aix43
04) sun5x/sol26

Install which platform's files? [01-4, q to quit, l for list]: 4
** You have chosen to install from sun5x/sol26 **

BEA eLink Information Integrator Release 1.0
This directory contains the Information Integrator for
SunOS 5.6 (Solaris 2.6) on SPARC.
Is this correct? [y,n,q]: y

To terminate the installation at any time
press the interrupt key,
typically <del>, <break>, or <ctrl+c>.

The following packages are available:

    1    infin          BEA eLink Information Integrator

Select the package(s) you wish to install (or 'all' to install all
packages) (default: all) [?,??,q]: 1

BEA eLink Information Integrator
(sparc) Release 1.0
Copyright (c) 2000, 1996-1999 BEA Systems, Inc., or its suppliers,
as applicable.
All Rights Reserved.
Distributed under license by BEA Systems, Inc.
BEA eLink is a trademark of BEA Systems, Inc.

Directory where Information Integrator files are to be installed
(Enter your Tuxedo directory path) [?,q]: /work/cmadm/tux65
```

Using /work/cmadm/tux65 as the Information Integrator base directory

Determining if sufficient space is available ...  
21098 blocks are required  
121658 blocks are available to /work/cmadm/tux65

```
Unloading /cmhome/dist/akita-2/sun5x/sol26/infin/INFINT65.Z ...
InfoInt/install.sql/NNF_GET_ALL_DSCR.sql
InfoInt/install.sql/NNF_GET_CASE.sql
InfoInt/install.sql/NNF_GET_CCNT.sql
InfoInt/install.sql/NNF_GET_CMPNT.sql
InfoInt/install.sql/NNF_GET_CNTL.sql
InfoInt/install.sql/NNF_GET_CNTL_NAM.sql
InfoInt/install.sql/NNF_GET_COL_ITEM.sql
InfoInt/install.sql/NNF_GET_CSTM_DT.sql
InfoInt/install.sql/NNF_GET_DEFAULT.sql
InfoInt/install.sql/NNF_GET_EXIT.sql
InfoInt/install.sql/NNF_GET_FIX.sql
InfoInt/install.sql/NNF_GET_FLAT_FMT.sql
InfoInt/install.sql/NNF_GET_ID.sql
InfoInt/install.sql/NNF_GET_INFIELDS.sql
InfoInt/install.sql/NNF_GET_JUSTIFY.sql
InfoInt/install.sql/NNF_GET_LAST.sql
InfoInt/install.sql/NNF_GET_LEN.sql
InfoInt/install.sql/NNF_GET_MATH_EXP.sql
InfoInt/install.sql/NNF_GET_NAME_VAL.sql
InfoInt/install.sql/NNF_GET_OUTFLD.sql
InfoInt/install.sql/NNF_GET_OUT_MSTR.sql
InfoInt/install.sql/NNF_GET_SUB.sql
InfoInt/install.sql/NNF_GET_SUBSTR.sql
InfoInt/install.sql/NNF_GET_TRIM.sql
InfoInt/install.sql/NNF_GET_USR_TYPE.sql
InfoInt/install.sql/NNR_GET_APP_MSG.sql
InfoInt/install.sql/NNR_GET_ARG.sql
InfoInt/install.sql/NNR_GET_BOOLEAN.sql
InfoInt/install.sql/NNR_GET_OPERNS.sql
InfoInt/install.sql/NNR_GET_RULES.sql
InfoInt/install.sql/NNR_GET_SUBACT.sql
InfoInt/install.sql/NNR_GET_SUBS.sql
InfoInt/install.sql/banner
InfoInt/install.sql/fmt_dlt_objs.sql
InfoInt/install.sql/fmt_grnt.sql
InfoInt/install.sql/fmt_meta.sql
InfoInt/install.sql/fmt_tbl_bld.sql
InfoInt/install.sql/inst_db.sh
InfoInt/install.sql/loadcomp.sh
InfoInt/install.sql/nn_post.sql
InfoInt/install.sql/nn_prdrop.sql
```

## 2 Installing BEA eLink Information Integrator on a Unix Platform

---

```
InfoInt/install.sql/nn_pre.sql
InfoInt/install.sql/nn_synonyms.sql
InfoInt/install.sql/rule_dlt_objs.sql
InfoInt/install.sql/rule_grnt.sql
InfoInt/install.sql/rule_meta.sql
InfoInt/install.sql/rule_prod.sql
InfoInt/install.sql/rule_tbl_bld.sql
InfoInt/install.sql/sqlload.sh
bin/AddDTfmt
bin/NNFie
bin/NNRDBARuleOwnership
bin/NNRTrace
bin/NNRie
bin/NNYORN
bin/NNcryptCfg
bin/apitest
bin/ii.cat
bin/ii.cfg
bin/iiserver
bin/msgtest
bin/ruletest
lib/libadk.so.1.0
lib/libnnNetExits.so
lib/libnncmpntmgr.so
lib/libnnfmgr.so
lib/libnnrmgr.so
lib/libnnrulesfmt.so
lib/libnnses.so
lib/libnnsesdbold.so
lib/libnnsq1.so
lib/libnntools.so
lib/libnnuserexit.so
sample/address.actions
sample/address.c
sample/address.data
sample/II_address_msg1.xml
sample/build_sample.cmd
sample/build_sample.sh
sample/Fieldtable.txt
sample/ii.cfg
sample/ud32.in

20830 blocks
... finished

Changing file permissions...
... finished
```

```
If your license file is accessible, you may install it now.  
Install license file? [y/n]: n
```

```
Please don't forget to use lic.sh located in your product bin  
directory to install the license file from the enclosed floppy.  
Refer to your product Release Notes for details on how to do this.
```

```
Installation of BEA eLink Information Integrator was successful
```

---

## Distribution Libraries and Executables

The BEA eLink Information Integrator CD-ROM contains the following libraries and executable programs. After installing the Information Integrator software, verify that these libraries and programs are installed on your system. Verify that the following files are installed by the Information Integrator software.

## Oracle Environment

**Table 2-1 Distribution Libraries and Executables for Oracle Environment**

<b>Directory</b>	<b>Files</b>
bin	AddDTfmt.exe
bin	apitest.exe
bin	dumpq.exe
bin	ii.cat
bin	ii.cfg
bin	iiserver.exe
bin	libadk.dll
bin	msgtest.exe
bin	NNcryptCfg.exe
bin	NNFie.exe
bin	NNRDBARuleOwnership.exe
bin	NNRie.exe
bin	NNRTrace.exe
bin	NNYORN.exe
bin	ruletest.exe
bin	sendBuf.exe
install.sql	banner
install.sql	fmt_dlt_objs.sql
install.sql	fmt_grnt.sql
install.sql	fmt_meta.sql
install.sql	fmt_tbl_bld.sql

**Table 2-1 Distribution Libraries and Executables for Oracle Environment**

<b>Directory</b>	<b>Files</b>
install.sql	inst_db.cmd
install.sql	loadcomp.cmd
install.sql	NNF_GET_ALL_DSCR.sql
install.sql	NNF_GET_CASE.sql
install.sql	NNF_GET_CCNT.sql
install.sql	NNF_GET_CMPNT.sql
install.sql	NNF_GET_CNTL.sql
install.sql	NNF_GET_CNTL_NAM.sql
install.sql	NNF_GET_COL_ITEM.sql
install.sql	NNF_GET_CSTM_DT.sql
install.sql	NNF_GET_DEFAULT.sql
install.sql	NNF_GET_EXIT.sql
install.sql	NNF_GET_FIX.sql
install.sql	NNF_GET_FLAT_FMT.sql
install.sql	NNF_GET_ID.sql
install.sql	NNF_GET_INFIELDS.sql
install.sql	NNF_GET_JUSTIFY.sql
install.sql	NNF_GET_LAST.sql
install.sql	NNF_GET_LEN.sql
install.sql	NNF_GET_MATH_EXP.sql
install.sql	NNF_GET_NAME_VAL.sql
install.sql	NNF_GET_OUTFLD.sql
install.sql	NNF_GET_OUT_MSTR.sql

**Table 2-1 Distribution Libraries and Executables for Oracle Environment**

<b>Directory</b>	<b>Files</b>
install.sql	NNF_GET_SUB.sql
install.sql	NNF_GET_SUBSTR.sql
install.sql	NNF_GET_TRIM.sql
install.sql	NNF_GET_USR_TYPE.sql
install.sql	NNR_GET_APP_MSG.sql
install.sql	NNR_GET_ARG.sql
install.sql	NNR_GET_BOOLEAN.sql
install.sql	NNR_GET_OPERNS.sql
install.sql	NNR_GET_RULES.sql
install.sql	NNR_GET_SUBACT.sql
install.sql	NNR_GET_SUBS.sql
install.sql	nn_post.sql
install.sql	nn_prdrop.sql
install.sql	nn_pre.sql
install.sql	nn_synonyms.sql
install.sql	rule_dlt_objs.sql
install.sql	rule_grnt.sql
install.sql	rule_meta.sql
install.sql	rule_prod.sql
install.sql	rule_tbl_bld.sql
install.sql	sqlload.cmd
lib	nncmpntmgr.dll
lib	nnfmgr.dll

**Table 2-1 Distribution Libraries and Executables for Oracle Environment**

<b>Directory</b>	<b>Files</b>
lib	nnNetExits.dll
lib	nnrmgr.dll
lib	nnrulesfmt.dll
lib	nnses.dll
lib	nnsesdbold.dll
lib	nnsq.dll
lib	nntools.dll
lib	nnuserexit.dll
lib	fmgr_mssql.dll
lib	nnsesmssql.dll
lib	rmgr_mssql.dll
lib	ses_mssql.dll

## SQL Server Environment

**Table 2-2 Distribution Libraries and Executables for SQL Server Environment**

<b>Directory</b>	<b>Files</b>
bin	AddDTfmt.exe
bin	apitest.exe
bin	dumpq.exe
bin	ii.cat
bin	ii.cfg
bin	iiserver.exe
bin	libadk.dll
bin	msgtest.exe
bin	NNcryptCfg.exe
bin	NNFie.exe
bin	NNRDBARuleOwnership.exe
bin	NNRie.exe
bin	NNRTrace.exe
bin	NNYORN.exe
bin	ruletest.exe
bin	sendBuf.exe
install.sql	banner
install.sql	fmt_dlt_objs.sql
install.sql	fmt_grnt.sql
install.sql	fmt_meta.sql
install.sql	fmt_tbl_bld.sql

**Table 2-2 Distribution Libraries and Executables for SQL Server Environment**

<b>Directory</b>	<b>Files</b>
install.sql	inst_db.cmd
install.sql	loadcomp.cmd
install.sql	NNF_GET_ALL_DSCR.sql
install.sql	NNF_GET_CASE.sql
install.sql	NNF_GET_CCNT.sql
install.sql	NNF_GET_CMPNT.sql
install.sql	NNF_GET_CNTL.sql
install.sql	NNF_GET_CNTL_NAM.sql
install.sql	NNF_GET_COL_ITEM.sql
install.sql	NNF_GET_CSTM_DT.sql
install.sql	NNF_GET_DEFAULT.sql
install.sql	NNF_GET_EXIT.sql
install.sql	NNF_GET_FIX.sql
install.sql	NNF_GET_FLAT_FMT.sql
install.sql	NNF_GET_ID.sql
install.sql	NNF_GET_INFIELDS.sql
install.sql	NNF_GET_JUSTIFY.sql
install.sql	NNF_GET_LAST.sql
install.sql	NNF_GET_LEN.sql
install.sql	NNF_GET_MATH_EXP.sql
install.sql	NNF_GET_NAME_VAL.sql
install.sql	NNF_GET_OUTFLD.sql
install.sql	NNF_GET_OUT_MSTR.sql

**Table 2-2 Distribution Libraries and Executables for SQL Server Environment**

<b>Directory</b>	<b>Files</b>
install.sql	NNF_GET_SUB.sql
install.sql	NNF_GET_SUBSTR.sql
install.sql	NNF_GET_TRIM.sql
install.sql	NNF_GET_USR_TYPE.sql
install.sql	NNR_GET_APP_MSG.sql
install.sql	NNR_GET_ARG.sql
install.sql	NNR_GET_BOOLEAN.sql
install.sql	NNR_GET_OPERNS.sql
install.sql	NNR_GET_RULES.sql
install.sql	NNR_GET_SUBACT.sql
install.sql	NNR_GET_SUBS.sql
install.sql	nn_post.sql
install.sql	nn_pre.sql
install.sql	rule_dlt_objs.sql
install.sql	rule_grnt.sql
install.sql	rule_meta.sql
install.sql	rule_prod.sql
install.sql	rule_tbl_bld.sql
install.sql	sqlload.cmd
lib	nncmpntmgr.dll
lib	nnfmgr.dll
lib	nnNetExits.dll
lib	nrmgr.dll

**Table 2-2 Distribution Libraries and Executables for SQL Server Environment**

<b>Directory</b>	<b>Files</b>
lib	nnrulesfmt.dll
lib	nnses.dll
lib	nnsesdbold.dll
lib	nnsql.dll
lib	nntools.dll
lib	nnuserexit.dll
lib	fmgr_ora7.dll
lib	nnsesora7.dll
lib	rmgr_ora7.dll
lib	ses_ora7.dll

# Preparing the Database Schema for Installation

## Creating Tablespaces

The database must have Information Integrator tablespaces created before you can install the database schema.

The size of your tablespaces depends on the numbers of rules and formats used at your site. See “Calculating Sizing Data” for specific sizing information.

You may want to place the tablespaces on different physical disks to balance I/O to avoid disk-access bottlenecks. You should separate data tablespaces and indexes by placing them on different disks or controllers. This optimizes index and data access parallelism.

### Oracle

To create Oracle tablespaces:

1. Create a dedicated Oracle instance where the Information Integrator database resides. Information Integrator is the default Oracle instance name used in the installation bat file.
2. Create the following tablespaces in the Oracle database:

<b>Table</b>	<b>Minimum Size</b>
TOOLS	1 MB
TEMP	10 MB
FORMATTER_DATA	20 MB
FORMATTER_INDEX	20 MB
RULES_DATA	20 MB
RULES_INDEX	20 MB

**Note:** For information on creating an Oracle database, refer to the Oracle installation documentation.

## Installing the Database Schema

**Note:** Make sure your Oracle environment is set up properly before performing these steps. See “Setting Up the Environment” for more information.

The `inst_db.sh` script creates the necessary tables and stored procedures in the Information Integrator database. The script sends the commands from the files in the `install.sql` directory.

1. Change to the `/opt/InfoInt/install.sql` directory.

2. To build the Information Integrator schema, execute the `inst_db.sh` script using the following syntax:

```
inst_db.sh SYS <SYS password> <servicename>
```

3. As the script runs, look for errors.
4. When the script completes the instantiation, a verification message appears.
5. For installation details and to see if there are errors, refer to the `inst_db.log` file. You should always check the log, even if the verification message says the instantiation completed successfully.

In the `inst_db.log` file, the error “table or view does not exist” does not indicate a problem with database instantiation. The database successfully instantiated if this is the only error you receive.

In addition to the `inst_db.log` file, an `NNFIE.log` may appear. This log only appears if there are failure messages; it will not exist otherwise.

## **2** *Installing BEA eLink Information Integrator on a Unix Platform*

---

# 3 Configuring BEA eLink Information Integrator

Configuring the environment for BEA eLink Information Integrator consists of the following basic tasks:

- Defining the Server
- Configuring BEA eLink Information Integrator

## Defining the Server

BEA eLink Information Integrator consists of a message format and routing server (IISERVER). This server identifies available service names and their associated application names and input formats.

Before running BEA eLink Information Integrator, you must identify the IISERVER server in the UBBCONFIG file. A sample UBBCONFIG file is provided on the installation CD-ROM. You can use this sample file as a base for creating your own UBBCONFIG file. Listing 3-1 shows the syntax for defining the IISERVER server in the UBBCONFIG file.

#### Listing 3-1 Syntax for IISERVER Server Definition in the UBBCONFIG File

---

```
*SERVERS
  IISERVER SRVGRP="identifier" SRVID="number"
  CLOPT= " -- -C configuration_file_name "
```

---

For information about the SRVGRP, SRVID, and CLOPT parameter syntax and definitions, refer to the *BEA TUXEDO Reference Manual*.

CLOPT= " -- -C configuration\_file\_name"  
specifies the eLink Information Integrator configuration file.

## Configuring BEA eLink Information Integrator

The `ii.cfg` file controls the operation of the eLink Information Integrator server (IISERVER). Following are the sections of the configuration file and the parameters you can define for each section. A sample configuration file is provided in the “Optional Parameters” section.

**Note:** `ii.cfg` is a generic filename. You can name this file anything you choose, but the filename must match the `-C configuration_file_name` parameter you specify in the TUXEDO UBBCONFIG file. (See Defining the Server for instructions on configuring the IISERVER server in the UBBCONFIG file.)

The eLink Information Integrator configuration file is divided into the following sections:

- ◆ **SERVER**  
Contains the general parameters required during server startup.
- ◆ **SERVICE**  
Contains a list of services to be performed and the parameters required for each service.

## Defining the SERVER Section

The syntax for the SERVER section of the eLink Information Integrator configuration file is as follows:

### Listing 3-2 Syntax for SERVER section

---

```
*SERVER
  DB_USERID="userID"
  DB_PASSWORD="password"
  DB_LOCATION="location"
  DB_INSTANCE="instance"
  #   MS SQL Server=4
  #   Oracle=8
  DB_TYPE="integer"
  MSG_CATALOG="catalog file name"
  MINMSGVLV="integer"
  MAXMSGLEVEL="integer"
```

---

## Required Parameters

The following parameters must be defined in the SERVER section:

### DB\_USERID

User ID for logging into the database.

### DB\_PASSWORD

Password for logging into the database.

### DB\_LOCATION

Location of the database.

### DB\_TYPE

Numeric representation of the database type. May be "0" for default database type.

### MSG\_CATALOG

Name of the TUXEDO catalog file. This file is installed as `ii.cat`.

### Optional Parameters

The following parameters are optional in the SERVER section:

#### DB\_INSTANCE

Indicates the instance of the database in MS SQL Server. Required if using an SQL Server database.

#### MAXMSGLVL

Indicates the maximum trace level range to be used by the eLink Information Integrator for logging trace messages.

#### MINMSGLVL

Indicates the minimum trace level range to be used by the eLink Information Integrator for logging trace messages.

## Defining the SERVICE Section

The syntax for the SERVICE section of the eLink Information Integrator configuration file is as follows:

### Listing 3-3 Syntax for SERVICE section

---

```
*SERVICE
  NAME="service name"
  APPL_NAME="application name"
  INPUT_FORMAT="input format"
  FML32_VAL_FLD="field name"
```

---

### Required Parameters

The following parameters must be defined in the SERVER section:

#### NAME

The service name being advertised.

#### APPLICATION\_NAME

The application name associated with the service, as defined in Rules. See “Working with Rules” for more information.

**INPUT\_FORMAT**

The input format required by the application, as defined in Rules. See “Working with Rules” for more information.

## Optional Parameters

**FML32\_VAL\_FLD**

Defines the field name that contains the message data. Used only if the input message is an FMLVO buffer.

## Defining the IIRESET Service

BEA eLink Information Integrator includes a service called IIRESET. This service is used to remove all cached information and reload this information from the database.

The syntax for defining the IIRESET service in the eLink Information Integrator configuration file is as follows:

### Listing 3-4 Syntax for the IIRESET service

---

```
*SERVICE
NAME=I IRESET
APPL_NAME=DUMMY
INPUT_FORMAT=DUMMY
```

---

## Required Parameters

The following parameters must be defined in the SERVICE section for the IIRESET service:

**NAME**

IIRESET, to indicate that the IIRESET service is being advertised.

**APPLICATION\_NAME**

The name of the application associated with this service.

**INPUT\_FORMAT**

The input format required for this application.

### Optional Parameters

There are no optional parameters for defining the IIRESET service.

## Sample eLink Information Integrator Configuration File

This section contains a sample eLink Information Integrator configuration file.

### Listing 3-5 Sample configuration file

---

```
*SERVER
DB_USERID=bea
DB_PASSWORD=bea
DB_LOCATION=dall
DB_INSTANCE=dall
# MS SQL Server = 4
# Oracle = 8
DB_TYPE=4
MSG_CATALOG=ii.cat
MINMSGLEVEL=0
MAXMSGLEVEL=1

*SERVICE
NAME=DebugII
APPL_NAME=BW_debug_appl
INPUT_FORMAT=BW_debug_in

*SERVICE
NAME=DebugIIExplode
APPL_NAME=BW_debug_appl
INPUT_FORMAT=BW_explode1_I

*SERVICE
NAME=IITEST
APPL_NAME=BW_debug_appl
INPUT_FORMAT=II_test1_in_I
```

---

# The Database Session Configuration File

The Database Session configuration file (`sqlsvses.cfg`) contains information about the database and database server used for eLink Information Integrator executables.

**Note:** The default location of the `sqlsvses.cfg` file is the local directory in which the executable is invoked. However, the location can be modified and centralized to another location by setting the environment variable `NN_CONFIG_FILE_PATH` to look for the file in a different directory.

The syntax for the Database Session configuration file is as follows:

## Listing 3-6 Syntax for `sqlsvses.cfg`

---

```
<sessionname>:<servername>:<userID>:<password>:<databasename>
```

---

The character length for the parameters in the `sqlsvses.cfg` file is dependent on your server platform and operating system. Line size in the `sqlsvses.cfg` file is limited to 1024 bytes. Each parameter must be separated by a colon.

## Required Parameters

The following parameters must be defined in the `sqlsvses.cfg` file:

### SESSIONNAME

Database session name to be used by eLink Information Integrator executables. This can be any string as long as it is unique within the file.

### SERVERNAME

Server where the eLink Information Integrator database resides.

### USERID

Database user ID.

**PASSWORD**

Database password.

**DATABASENAME**

Name of the database where the eLink Information Integrator tables reside.

**Note:** The DATABASENAME parameter is not used for Oracle environments.

There must be a colon after the password in Oracle environments, even though the last parameter is not used.

## Optional Parameters

There are no optional parameters for the `sqlsvses.cfg` file.

## Sample Database Session Configuration File

A sample `sqlsvses.cfg` file is included with the eLink Information Integrator software. To customize this file for your environment, uncomment the line relating to your database (SQL Server or Oracle) and edit the appropriate parameters (server, user ID, password, etc.).

### Listing 3-7 Sample Database Session configuration file

---

```
# -----  
#  
# Sqlsvses.cfg - Defines sessions to be used to access the Formatter  
# database. The format of each line depends on the database in which  
# the Formats are to be stored.  
  
#  
# Four fields are required in order to specify a connection to a  
# Oracle database. Five fields are required to specify a connection  
# to a SQLServer database.  
  
#  
# Entries can be commented by placing a "#" in column 1.
```

```
# -----
#
# Format of lines for the various supported database types:
#
# MS SQLServer:
# SessionName:dBaseServer:dBaseUserid:dBasePassword:dBaseInstance
#
# Oracle:
# SessionName:dBaseService:dBaseUserid:dBasePassword:
# (note that the fifth field is blank, so the last colon is required)
#
# -----
#
# The SessionName "import" is required to run the Message Definition
# Importer Tool (MsgDefAdmin).
#
#
# -----
#
# Example SessionNames (uncomment the SessionNames needed):
#
# MS SQL Server database: import:Server:Userid:Password:Instance
# (change Server, Userid, Password, and Instance above)
#
# Oracle database (final colon required on each line):
# import:Service:Userid:Password:
# (change Service, Userid, Password above)
# ----- ( End of File ) -----
```

---

### **3** *Configuring BEA eLink Information Integrator*

---

# 4 Working with Formatter

This chapter discusses the following topics:

- Formatter Overview
- Importing and Exporting Formats
- Testing Formats

## Formatter Overview

The BEA eLink Information Integrator Formatter is a graphical user interface that allows you to provide the format definition data and store the information in a relational database. The eLink Information Integrator server then uses this definition data to translate messages from one format to another. The server can handle multiple message format types from multiple data value sources, and can convert and parse messages. Messages can be converted from any described format to any other described format.

Two executables, `apitest` and `msgtest`, are delivered with the eLink Information Integrator. These two executables show how to invoke the public functions and serve as tools for validating format definitions. The `apitest` executable parses an input message and displays a hierarchical representation of the parse tree. The `msgtest` executable reformats an input message into an output message.

A command line tool, `NNFie`, is also delivered with eLink Information Integrator. This tool allows you to export format definitions from a database to an export file and to import from the export file into a database.

## Fields and Input Controls

Information contained within a structured input message can be broken into individual fields using input controls. Input controls define how to parse an individual field. Defined by a unique name and control information used to define their beginning and end (input control), fields are cohesive parts of a message representing some type of information.

Each field has an associated parse (input) control that describes how to identify the field in the message. Input control information includes the data type for the field, tags preceding and/or following the field, the length of the field, the number of times the field repeats within a message, and literals. Repetition count indicates how many times a certain field will appear in a message.

eLink Information Integrator supports several data types including ASCII string, ASCII numeric, and binary.

Tags are sets of bits or characters explicitly defining a string of data. For example, `<DATE>` and `</DATE>` might mark the beginning and end of a date field in a message.

Literals are symbols used in programming languages. For example, a literal can represent numbers or strings that provide an actual value instead of representing possible values. Literals might contain only ASCII values and are often used as delimiters to separate fields in a message.

Regular Expressions (REs) are strings expressing rules for string pattern matching. Within input parse controls, use REs to match ASCII field data in input fields. Instead of searching for a defined literal, you can use an RE to search for complex string patterns in field data. String-matching capabilities comply with the POSIX 1003.2 standard for regular expressions.

## Output Controls

For each field in an input message you want to appear in an output message or use to affect a resulting field in an output message, you must have a matching output format control. Output controls specify how to get a starting value for the output field, what data type transformation to perform, and what formatting operations to perform (for example, prefix, suffix, trim).

Defined in much the same way as parse controls, output controls contain additional information such as the type of mathematical operation, prefix and suffix data, pad characters, and default value.

## Formats

Simple formats are defined by grouping fields and their parse or output format controls. Messages are described using individual data fields. However, there can be several layers of complexity in a format definition before the actual field values within a message can be determined.

Formats can be one of two types: flat or compound. Flat formats only contain fields and their input or output format controls. Compound formats contain one or more formats, each of which can be either flat or compound.

Input formats (flat or compound) contain fields and their parse (input) controls, and are used to parse messages so they can be reformatted according to output formats (flat or compound).

The eLink Information Integrator server uses the format descriptions defined in the Formatter graphical user interface to recognize and parse input messages and reformat output messages. The server uses these descriptions to interpret the values in incoming messages and to construct outgoing messages.

Possible transformations the eLink Information Integrator server can perform include:

- Adding, removing, or rearranging data, literals, tags, and delimiters (delimiters are logically cohesive sequences of characters forming a field terminator or format terminator)
- Converting between data types

- Inserting literals into output
- Inserting headers and trailers (including control characters) around any field
- Performing arithmetic operations on numeric data

# Importing and Exporting Formats

This section describes how to use the command line utility NNFile to import and export formats.

## NNFile

NNFile exports format definitions from a database to an export file and imports from the export file into a database. The Unix command for running NNFile is as follows:

```
NNFile
```

**Note:** To use NNFile, Unix users must have write permissions to the current directory.

The NT command for running NNFile.exe is as follows:

```
NNFile.exe
```

The export file for NNFile is not interchangeable with the files created by the GUI. NNFile, NNRie, and `sqlsvses.cfg` must be in the same directory as NNFile.

**Warning:** Do not name components the same with only a change in case to identify them. For example, do not name one format “f1” and another format “F1”. You must make each item unique using something other than case differences.

**Note:** File names (including absolute paths) for both import and export must be no longer than 255 characters.

Listing 4-1 shows the syntax for the NNFile command

**Listing 4-1 Syntax for NNFie**

```

NNFie
((-C <command file name>)
(-i <import file name> [-T] [ -o|-g|-n|-4]
[-s <session name>])

(-e <export file name> [-m <format name>+] [-q "comment"]
[-Q <Comment file name>] [-w <number>] [-s <session name>])
(-t <import file name> [-s <session name>])
(-I <import file name> [-s <session name>]))

```

In the above example:

- [ ] represents optional
- () represents grouping
- | represents XOR
- + represents one or more
- <> means replace with user-provided data

You must keep the NNFie options in the correct position. For example, the following command is correct:

```
>nnfie -e myfile -m myformatname -s nnfie
```

The following command is incorrect because of the position of the options:

```
>nnfie -e my file -s nnfie -m myformatname
```

Table 4-1 lists the NNFie options and their definitions.

**Table 4-1 NNFie Options**

Name	Mandatory/Optional	Description
-C [<command file>]	Optional	<p>Alternate command file name; default file is NNFie.cmd. If this option is provided, NNFie reads command line options from a file instead of the command line.</p> <p><b>Note:</b> Command line option -C puts import/export command options in a text file. Do not use quotation marks around names (e.g., format name, session name, etc.) in the text file. Also, do not use back slashes in command lines.</p>

**Table 4-1 NNFile Options**

Name	Mandatory/Optional	Description
-i [<import file>]	Mandatory for Import	This parameter is required to import data from the named file and is mutually exclusive from -e. The named file default is NNFile.exp. If you use the command line option -i, then the following options are available to you: [-T] [-o -g -n -4]. These additional options are described below the parameters table.
-e [<export file>]	Mandatory for Export	This parameter is required to export data from the named file and is mutually exclusive from -i. The named file default is NNFile.exp. If you use the command line option -e, then the following options are available to you: -q, -Q, -w, and -m. These additional options are described below the parameters table.
-s [<session name>]	Optional	Name of session in sqlsvses.cfg. Defaults to NNFile.
-I<import file name>	Mandatory	Writes description of all conflicts in import file to NNFile.log.
-t <import file name>	Mandatory	Writes an inventory of the import file to NNFile.log.

### NNFile Import Syntax

Listing 4-2 shows the syntax for executing an import command.

#### Listing 4-2 Syntax for NNFile Import

---

```
$ NNFile -i [<file name>] [-s <session name>]
```

---

NNFile stores error messages in the `NNFile.log` file. If a component fails to import, the line containing an error from the export file is written to `NNFile.err`.

Table 4-2 lists the NNFile import options and their definitions.

**Table 4-2 NNFile Import Options**

Import Options	Mandatory/Optional	Description
-T	Optional	Loads import file as one transaction. If an import failure for one component is detected, then the entire import is rolled back. The default behavior is a transaction boundary for each component.
-o	Optional	Overwrites all conflicts and replaces all components of same name with those in the export file.
-g	Optional	Ignores all conflicts and uses existing component definitions.
-n	Optional	Implements the interactive conflict resolution option. NNFile defaults to -n if no options are selected.
-4	Optional	Use R4_0 conflict resolution if a component in the export file conflicts with current data in the database. Do not import the new component but flag it in the error file and do not import any components that rely on the conflicting component.

## NNFile Export Syntax

Listing 4-3 shows the syntax for executing the command to export an entire database.

### Listing 4-3 Syntax for NNFile Export (entire database)

```
$ NNFile -e [<export file name>] [-s <session name>]
```

Listing 4-4 shows the syntax for executing the command to export a single format.

### Listing 4-4 Syntax for NNFile Export (single format)

```
$ NNFile -e [<export file name>] [-m <format name>] ] [-s <session name>]
```

Listing 4-5 shows the syntax for executing the command to export multiple formats.

### Listing 4-5 Syntax for NNFie Export (multiple formats)

---

```
$ NNFie -e [<export file name>] [-m <format name> <format name>
... ] [-s <session name>]
```

---

Table 4-3 lists the NNFie export options and their definitions.

**Table 4-3 NNFie Export Options**

Export Options	Mandatory/Optional	Description
-q	Optional	Adds comments within quotes to top of the export file.
-Q	Optional	Adds contents of <comment file> to top of export file.
-w	Optional	Sets maximum line length in export file. Default value is 80.
-m [<message type>]	Optional	Specifies the message type to export. By default, exports all messages types within the specified application group.

## Conflict Resolution

A conflict occurs when an imported component does not match an existing component in the database of the same name and type.

**Note:** NNFie is not designed to import or export databases that are corrupt or have unresolved issues with the data.

## Inventory Components

When a component is overwritten, the component definition within the export file is imported into the database. To implement the batch overwrite conflict resolution, type the following syntax:

```
NNFie -i <filename> -o
```

When a component is either ignored or skipped, the component in export file **is not** imported into the database. To implement the batch Ignore/Skip conflict resolution, type the following syntax:

```
NNFie -i <filename> -g
```

Components that are skipped may cause the import of supporting components that will not be used. Overwriting existing components may cause existing components not to be used. This does not affect the integrity of the database, but requires use of a clean-up utility.

When the system identifies a conflict in interactive mode, it displays a description of both the existing and import components, and you are presented with three options: Overwrite, Ignore, and Rename the imported component. If you select the Rename option, all references to this component within the export file are updated.

To implement the interactive conflict resolution option, type the following syntax:

```
NNFie -i <filename> -n
```

To implement conflict resolution used in release 4.0, where no conflicting components are imported, type the following syntax:

```
NNFie -i <filename> -4
```

If you do not choose a conflict resolution option, the interactive resolution is used as the default. All conflicts and their resolution are reported to the NNFie.log file.

### Example

```
Literal:"MyLiteral" conflicts with an existing Formatter element!  
literalLength ( existing=2 | incoming=3 )  
Overwrite, Ignore or Rename component (OIR): R  
Please enter new component name: MyLiteral_NewValue
```

## Troubleshooting Export Failures

You can take inventory of the components contained in an NNFie export file. This inventory option produces a component inventory listing in the file named `NNFie.log`. The command syntax is as follows:

```
NNFie -t <filename>
```

You have the option of identifying all conflicts without importing any data. This test import function allows you to verify the contents of export files within working databases, thus facilitating easy validation for archiving. Any conflicts are recorded in the `NNFie.log` file. To use this conflict report, type the following syntax:

```
NNFie -I <filename> -k
```

### Troubleshooting Import Failures

If `NNFie` fails to import from a given export file, view the `NNFie.log` to determine the cause for import failure.

The following types of errors can cause an import to fail:

- Conflict errors — data already exists in the database that conflicts with imported data.
- Nonconflict error for a specific component — This error message should be complete without any specific component information:
- Nonconflict error for a specific Formatter element -- This error message contains both Formatter component identification and the data that is being imported:
- Conflict error for a specific Formatter element -- The data being imported conflicts with data already existing in the database. View the data and either remove the conflicting data in the destination database, or fix the data in the originating database, re-export the data, and import the newly exported data.

---

# Testing Formats

Two programs you can use to test formats are provided with eLink Information Integrator. These programs are:

- APITEST
- MSGTEST

## APITEST

The `apitest` executable outputs the structure and contents of a message parsed by the eLink Information Integrator server.

The syntax for `apitest` is as follows:

---

### Listing 4-6 Syntax for APITEST

---

```
apitest[-d[<filename>]]  
-d :parse debug on
```

---

The `-d [filename]` parameter sets debugging mode to parse for this run of `apitest`. `[filename]` specifies an optional file where debug information is written. If `[filename]` is not specified, debug information is written to the screen (STDOUT).

## Using apitest

To run `apitest`:

1. At the command line prompt, type `apitest`.
2. At the prompt, *Enter the input file name:*, type the name of the file in this directory that contains the message to be parsed and reformatted.

3. At the prompt, *Enter the input format name:*, type the name of the input format that will be read from the NNF-FMT table in the database identified in the sqlsvses.cfg file.

## MSGTEST

The `msgtest` executable uses input and output formats, delimiters, and other control information read from the database to parse and reformat an input message read from a file. The information needed by `msgtest` must be placed in the database using the Formatter graphical user interface.

### Listing 4-7 Syntax for MSGTEST

---

```
msgtest[-li][-lo][-if][-nv][-d[<filename>]][-dcp]
[-dcm][-dco]]

-li:          loud input
-lo:          loud output
-lf:          loud formatted value
-nv:          no validation
-d:           debug on (debug parse only if -dcp and -dcm and
-dco not specified)
-dcp:         debug parse on
-dcm:         debug map on
-dco:         debug output on
```

---

The `-d [filename]` parameter sets debugging mode to parse for this run of `msgtest`. `[filename]` specifies an optional file where debug information is written. If `[filename]` is not specified, debug information is written to the screen (STDOUT).

## Using msgtest

To run `msgtest`:

1. At the command line prompt, type `msgtest`.
2. At the prompt, *Enter the input file name:*, type the name of the file in this directory that contains the message to be parsed and reformatted.

3. At the prompt, *Enter the output file name:*, type the name of the file that will contain the reformatted message.
4. At the prompt, *Enter the input format name:*, type the name of the input format that will be read from the NNF-FMT table in the database identified in the `sqlsvses.cfg` file.
5. At the prompt, *Enter the output format name:*, type the name of the output format that will be read from the NNF\_FMT table in the database identified in `$msgtest<myFormatterTest.txt>`.

To run `msgtest` more than once using the same information, create a text file. Listing 4-8 shows an example command line using a `msgtest` text file.

---

**Listing 4-8 Syntax for calling a MSGTEST text file**

---

```
$ msgtest<myFormatterTest.txt>
```

---

The file `myFormatterTest.txt` contains:

`ascii_string`

The input file name containing the message.

`output_AS1`

The output file name that will contain the translated message.

`AS_IF`

The input format to be read from the database.

`AS_NA1_OF`

The output format to be read from the database.

## Configuration File

Before running test executables, verify that the `sqlsvses.cfg` file includes the database name and server name information used to execute this program. This file must also be in the same directory as the executable program.

For test executables, the session name to be entered in the `sqlsvses.cfg` file is `new_format_demo`.

### **Listing 4-9 Syntax for new\_format\_demo**

---

```
new_format_demo:MyServerName:MyUserName:MyPasswordName:  
MyDatabaseName
```

---

For more information on the `sqlsvses.cfg` file, see “Configuring BEA eLink Information Integrator.”

# 5 Working with Rules

This chapter discusses the following topics:

- Rules Overview
- Defining Users in Rules
- Importing and Exporting Rules

## Rules Overview

The BEA eLink Information Integrator Rules Engine is a graphical user interface that enables you to allow you to provide the rule definition data and store the information in a relational database. The eLink Information Integrator server then uses this definition data to evaluate messages and react to the evaluation results.

Rule definitions describe how to parse a message using the format parameters (specified in the Formatter graphical user interface) against the rules defined for the message. The rules definitions include subscriptions and the actions to perform if the rule hits (evaluation criteria are true).

The Rules Consistency Checker utility checks the validity of the rule definition data in the relational database. As rule definition data is built and maintained, users should run the consistency checker periodically to ensure data integrity.

The NNRie tool delivered with eLink Information Integrator is a command line tool that can be used to export rule definitions from a database to a file, and to import the exported file into a database.

## Application Groups

Application groups are logical divisions of rule sets for different business needs. You can define unlimited application groups. For example, you might want the rules for the accounting department and the application development department separated into two groups. You could define Accounting as one application group, Application Development as another, and then associate rules with each group as appropriate.

## Message Types

Message types define the layout of a string of data. Each application group can contain several message types, and a message type can be used with more than one application group. Message types are defined by the user. When using Formatter, a message type is the same as an input format name. This format name is used by Information Integrator to parse input messages for Rules evaluation.

## Rule Names

When a rule is created, each rule is assigned a rule name and associated with an application group and message type. Each rule is uniquely identified by its application group/message type/rule name.

The following items must be defined for each rule:

- Evaluation criteria — An expression that contains arguments and operators
- Subscription information — Subscriptions, actions, and options

## Expressions, Arguments, Boolean, and Rules Operators

An expression (evaluation criteria) contains a list of fields, associated operators, and associated comparison data (either static values or other fields) connected with Boolean operators. An argument contains the combination of a field name, Rules comparison operator, and static value or other field name. Field names depend on the message type (input format name) and they are defined using the Formatter graphical user interface. Rules comparison operators are already defined within Rules. Field comparisons can be made against static data or other field values. Arguments are linked together with Boolean operators ‘&’ (AND) and ‘|’ (OR) and parentheses can be used to set the evaluation priority.

## Subscriptions, Actions, and Options

When a rule evaluates to true, it is considered a “hit.” If the rule does not evaluate to true, it is considered a “no-hit.” When a rule hits, you can retrieve associated subscriptions to be taken by the application. These subscriptions are the actions or commands, and the associated parameters are the options to execute them.

Subscriptions are lists of actions to take when a message evaluates to true. Each rule must have at least one associated subscription. Subscriptions are uniquely identified within an application group/message type pair by a user-defined subscription name. You can define as many subscriptions as you need. Each action within a subscription is defined by an action name. The action does not need to be unique, since all actions are intended to be executed in sequence. A single subscription can be shared by multiple rules. In this case, the shared subscription would be retrieved only once no matter how many of its rules hit.

An action has a list of one or more associated options. An option consists of an option name-value pair. The user defines all action names and option name-value pairs.

## Defining Users in Rules

This section provides instructions for creating user accounts in eLink Information Integrator Rules

# Oracle

During installation, a role is created for each user: `II_USER`. To access databases, users must be created and associated with the `II_USER` role using the procedures described in the following sections.

## Creating Users

After you install eLink Information Integrator, you must create user names or assign user roles in your database. User names identify individual users to the database. Listing 5-1 shows the syntax for creating users.

---

### Listing 5-1 Syntax for creating users

---

```
create user USERNAME identified by PASSWORD;
```

---

`USERNAME` and `PASSWORD` are required parameters.

Once a user has been created, you must define the synonyms for that user. Connect to the database using the newly created `USERNAME` and `PASSWORD`; then, run the SQL\_Plus executable `nn.synonyms.sql`, found in the `install.sql` directory. Listing 5-2 shows the syntax for defining synonyms.

---

### Listing 5-2 Syntax for synonyms.sql

---

```
@nn.synonyms.sql
```

---

## Granting Roles to Users

Users must be given permissions to access the database data. You can either grant permissions to an individual user or create roles with certain permissions and associate users with specific roles. `II_USER` is a role created by the eLink Information Integrator installation process. Listing 5-3 shows the syntax for granting user roles.

**Listing 5-3 Syntax for granting user roles**

---

```
grant II_USER to USERNAME;
```

---

The II\_USER role is granted to the user identified by USERNAME.

## SQL Server

Users must have login accounts and a user name in each database they want to access. Adding login accounts, database users, and object permissions can be done by the system administrator, security officer, or database owner. A single person can occupy one or more of these roles. Check with your database administrator for information about your specific environment.

### Creating Login Accounts

Login accounts give users access to the SQL Server. They are created by the system administrator or security officer using the `sp_addlogin` system procedure. Listing 5-4 shows the syntax for `sp_addlogin`.

**Listing 5-4 Syntax for `sp_addlogin`**

---

```
sp_addlogin loginName, password [, defdb [, deflang [,  
full-name]]]
```

---

`loginName`

The login name being added.

`password`

The password associated with the login name being added.

`defdb`

Specifies a default database for the user.

deflang

Specifies the default language to use.

full-name

The full name of the user who owns this account.

Login accounts only give access to the SQL Server. To access a database, a login must be assigned to a user name to the databases the user wants to access.

### Assigning Users to a Database

To use a database, the server login must be associated with a user name in the database. Users can be added to a database by the database owner (DBO) using the `sp_adduser` system procedure. This procedure must be run from the database in which the user is to be added. Listing 5-5 shows the syntax for `sp_adduser`.

---

#### Listing 5-5 Syntax for `sp_adduser`

---

```
sp_adduser loginName [, nameInDB] [, group]
```

---

loginName

The user's server login account.

nameInDB

The name for the user in the database. Defaults to the loginName if not specified.

group

Enables the DBO to add the user to an existing group in the database. If not specified, the user is placed in the default group, PUBLIC.

### Defining User Groups

Each user added to the database must be granted permissions to access objects within that database, unless they are the database owner. During installation, the group `II_USER` is created for eLink Information Integrator users. To access eLink Information Integrator databases, users must be linked to the `II_User` group.

Users can be added using either the `sp_adduser` or `sp_changegroup` system procedures. Listing 5-5 describes the syntax for `sp_adduser`.

**Listing 5-6 Syntax for `sp_changegroup`**

---

```
sp_changegroup groupName, userName
```

---

`groupName`

Name of the group to which the user is added.

`userName`

The database user name.

## Importing and Exporting Rules

This section gives instructions for using the command line utility NNRie to import and export rules.

### NNRie

NNRie is a command line tool that you can use to export rule definitions and orphan subscriptions (subscriptions that are not associated with a rule) from a database to a file and to import the exported file into a database. To use NNRie, Unix users must have write permissions to the current directory. Listing 5-7 shows the syntax for NNRie.

### Listing 5-7 Syntax for NNRie

---

```

NNRie ((-C [<command file name>] |
      -v |
      (-i <import file name>|-e <export file name>
      [[-a <appname> [...]] [-m <msgname>] [...]] [-r
      <rulename>] [...]] [-S <subname>] [...]]
      [-T [<trace file name>] ]
      [-l [<conflict report file name>] ]
      [-t [<inventory report file name>] ]
      [-f [<failure file name>] ]
      [-s <session name>]
      [-o]
      [-c <database configuration file name>]))

```

---

Table 5-1 lists NNRie options and their definitions.

**Table 5-1 NNRie Options**

Name	Mandatory/Optional	Description
-C [<command file>]	Optional	Alternate command file. The default is NNRie.cmd. If this option is provided, NNRie reads command line options from a file instead of a command line. If -C is present, NNRie expects the other parameters to be in the command file named in the same format as the command line.
-V (version)	Optional	Shows program version information only and does no processing.
-i [<import file>]	Mandatory for Import	Indicates the program should import data from the named file. This parameter is required to import data and is mutually exclusive with -e. This parameter may be followed by the name of a file that contains the import data. The referenced file must have been created with the NNRie -e option. The default file name is NNRie.exp.
-e [<export file>]	Mandatory for Export	Indicates the program should export to the named file. This parameter is required to export data, and is mutually exclusive with -i. This parameter may be followed by the name of a file to hold the export data. The default file name is NNRie.exp.

**Table 5-1 NNRie Options**

Name	Mandatory/Optional	Description
-s <session name>	Optional	The session name corresponding to the session identifier in the configuration file (See the -c option below). The default session tag is "nnrmie".
-o (overwrite flag)	Optional	The default behavior is off (do not overwrite). If this parameter is present during export, it overwrites the export file. If this parameter is present during import, and a rule or subscription defined in the import file already exists in the importing database, the old rule is overwritten with the new definition if you have update permission. If you do not have update permission, an error is noted and the rule is replaced. If not overwriting rules, any rule that cannot be processed because it already exists in the importing database is noted.
-c <config file>	Optional	Indicates the name of the configuration file the program should read to load its session data for access to a database. The default configuration file is <code>sqlsvses.cfg</code> .
-a <application group>	Optional	Identifies the application group to export. If a value for this parameter is not identified, all application groups are exported. This parameter can be repeated as many times as necessary to define multiple application groups to export.
-m <message type>	Optional	Specifies the message type to export. This parameter also requires the -a parameter to be set. The default behavior is to export all message types within the specified application group. This parameter can be repeated as many times as necessary to define multiple message types within the same application group.
-S <subscription name>	Optional	Specifies the name of the subscription to export. This parameter also requires the -e, -a, and -m parameters to be set. This parameter can be repeated as many times as necessary to export multiple subscriptions.
-r <rule name>	Optional	Specifies the name of the rule to export. This parameter also requires the -a and -m parameters to be set. The default behavior is to export all rules within the specified application group and message type. This parameter can be repeated as many times as necessary to define multiple rules within the same application group and message type.

**Table 5-1 NNRie Options**

Name	Mandatory/Optional	Description
-t [<inventory filename>]	Optional	Creates an inventory of an export file in NNRie.log (does no processing).
-T [<trace file name>]	Optional	Specifies the name of the trace file. Default trace file is NNRieT.log.
-O	Optional	Completely overwrites imported message types (import only). The default behavior is off (do not overwrite).
-l [<conflict report filename>]	Optional	Reports on any import conflicts. The default behavior is off (does no processing). Default file is NNRie.log.
-g	Optional	Ignore and do not import any conflicting rules and subscriptions.
-n	Optional	Implement interactive conflict resolution. The default behavior is on. MVS default is off.
-q <comments in double quotes>	Optional	Includes comments in an export file.
-Q <comments file name>	Optional	Includes a file of comments in an export file. No default.
-f [<failure file>]	Optional	Specifies the failure file that contains lines not imported. The default file is NNRie.err.

**Note:** If there are no -a, -m, -r, or -S options, the entire database exports.

### NNRie Import Syntax

Listing 5-8 shows the syntax for importing a rule.

#### Listing 5-8 Syntax for importing a rule

---

```
$ NNRie -i [<file name>] [-s <session name>]
```

---

If the file fails to import, an error message is generated and NNRie errors out.

## NNRie Export Syntax

Listing 5-9 shows the syntax for exporting an entire database

### **Listing 5-9 Syntax for exporting an entire database**

---

```
$ NNRie -e [<export file name>] [-s <session name>]
```

---

Listing 5-10 shows the syntax for exporting a single application group.

### **Listing 5-10 Syntax for exporting a single application group**

---

```
$ NNRie -e [-a <app group name>]
```

---

The application group name exports and then each message type within the application group exports. The message type export includes all subscriptions and rules in the specific application group/message type. This procedure is followed for each application group if multiple application group names are given.

Listing 5-11 shows the syntax for exporting a message type for an application group.

### **Listing 5-11 Syntax for exporting a message type for an application group**

---

```
$ NNRie -e [-a <app group name>][-m <msgtype name>]
```

---

The application group name and message type name exports, then the rules export with the links to subscriptions. All subscriptions in the application group/message type export, whether they are linked to rules or not. If multiple message type names are given, the subscriptions and rules for each message type export.

Listing 5-12 shows the syntax for exporting a single application group.

### Listing 5-12 Syntax for exporting a single rule

---

```
$ NNRie -e [-a <app group name>] [-m <msgtype name>] [-r <rule name>]
```

---

The rule's application group name and message type name exports. All subscriptions linked to the rule export with permissions, actions, and options and then the rule information exports with permissions, expressions, and links to subscriptions. If multiple rule names are given, the subscriptions linked to each rule export with no duplicates, and then the rules export.

Listing 5-13 shows the syntax for exporting a single application group.

### Listing 5-13 Syntax for exporting more than one rule

---

```
$ NNRie -e [-a <app group name>][-m <msgtype name>][-r <rule name> <rule name>...]
```

---

### Listing 5-14 Syntax for exporting a single subscription

---

```
$ NNRie -e [-a <app group name>][-m <msgtype name>][-S <subscription name>]
```

---

No rule information exports. The application group and message type name information exports and then the subscription information exports without the rule name. If multiple subscriptions are given, each subscription exports.

## Command Line Functions

The NNRie command line functions are described below.

To overwrite component by component, enter the following syntax:

```
NNRie -i <filename> -o
```

To run the batch Ignore/Skip conflict resolution, enter the following:

```
NNRie -i <filename> -g
```

To run the interactive conflict resolution option, enter the following:

```
NNRie -i <filename> -n
```

To run the “check only”, conflict reporting only option, enter the following:

```
NNRie -i <filename> -l (Writes to NNRie.log)
NNRie -i <filename> -l MyCLog.txt (Writes to MyCLog.txt)
```

To import and totally overwrite the application group message type pair in the database, enter the following syntax:

```
NNRie -i NNRie.exp -O
```

To trace the command that is about to be executed and save to a log file, enter the following:

```
NNRie -i NNRie.exp -T (Writes to NNRieT.log)
NNRie -i NNRie.exp -T trace.log (Writes to trace.log)
```

To produce an inventory of an export file, enter the following:

```
NNRie -t NNRie.exp (Writes to NNRie.log)
NNRie -i NNRie.exp -t inv.log (Writes to inv.log)
```

To add comments to the header of the Export file, enter the following:

```
NNRie -e <filename> -q "additional comment between quotes"
```

To add a file of comments to the header of the Export file, enter the following:

```
NNRie -e <filename> -Q <comment file>
```

To implement the batch Overwrite conflict resolution, enter:

```
NNRie -i <filename> -o
```

To implement the batch Ignore/Skip conflict resolution, the syntax is:

```
NNRie -i <filename> -g
```

To implement the interactive conflict resolution option, the syntax is:

```
NNRie -i <filename> -n
```

To implement the conflict report option, the syntax is:

```
NNRie -i <filename> -l <optional filename>
```

If no conflict resolution option is chosen, the interactive resolution is used as the default.

The user should be able to replace an entire application group/message type pair by entering the following command:

```
NNRie -i NNRie.exp -O
```

This command deletes each message type from the database that it encounters in the import file and all the rules and subscriptions under it before importing new information. If it fails to delete because of rights violations or other problems, it returns an error message and does not import the new information.

If NNRie is not designed to import or export databases that are corrupt or have unresolved issues with the data.

## Conflict Resolution

This section describes the files provided with eLink Information Integrator to resolve import and export conflicts.

### NNRie.log File

If NNRie is unable to import an application group, message type, rule or subscription, the corresponding import information is written to the `NNRie.err` file. The `NNRie.err` file can be modified to fix the problem with the component and then used to perform the import operation again. The reason for the component's failure to import is written to the `NNRie.log` file. Listing 5-15 shows an example of a `NNRie.log` file.

---

#### Listing 5-15 Sample `NNRie.log` file

---

```
Conflict with Subscription: 'S3'  
  App Name: 'MsgTest'  
  Msg Name: 'MsgTest'  
Subs in import file:  
  Owner: 'Public'  
  Comment: 'New Checking'
```

```
Subs in Database:
  Owner: 'PUBLIC'
  Comment: ''
  Conflict Exists in : Comment
```

---

### Trace.log File

The `Trace.log` file contains a progress report of the import.

```
NNRie -i NNRie.exp -T trace.log
```

This command shows, line by line, what will be imported. If a process fails, the log stops within the errant process.

The following letters define import and export components:

- A = Application group
- M = Message Type
- R = Rule
- n = permission (either rule or subscription)
- S = subscription - written to file or added to database
- C = action - written to file or added to database
- P = option - written to file or added to database
- s = subscription - read from file
- c = action - read from file
- p = option - read from file
- l = Subscription linked to a rule in the database

### Inventory Export File

The inventory export file provides a tool to determine the items contained within an export file. The default log file is `NNRie.log`.

```
Example NNRie -t NNRie.exp (Writes to NNRie.log)
```

```
Example NNRie -i NNRie.exp -t MyInv.log (Writes to MyInv.log)
```



# A Error Messages

This document contains the following descriptions of error, informational, and warning messages that can be encountered while using BEA eLink Information Integrator.

## NNFie Error Messages

<b>-4001: NNFIEE_FILE_EXISTS</b>	<b>ERROR: Given file already exists (so will not replace it)</b>
	<b>DESCRIPTION</b> The specified export file name already exists.
	<b>ACTION</b> Remove the file or specify a different export file name.
<b>-4002: NNFIEE_NO_IMPORT_FILE</b>	<b>ERROR: No import files by the given name exist</b>
	<b>DESCRIPTION</b> The specified import file does not exist.
	<b>ACTION</b> Create the file or check the accuracy of the input file name.
<b>-4003: NNFIEE_FAILED_TO_READ_FROM_IMPORT_FILE</b>	<b>ERROR: Failed to read from the import file</b>
	<b>DESCRIPTION</b> The file cannot be read.
	<b>ACTION</b> Check for the existence of the file or possible access problems.

## A Error Messages

---

<b>-4004: NNFIEE_FAILED_TO_READ_FROM_IMPORT_FILE</b>	<b>ERROR: Failed to separate and get/return a piece of the input data</b>
	<b>DESCRIPTION</b> The import file has been corrupted.
	<b>ACTION</b> Restore or recreate the file.
<b>-4005: NNFIEE_BAD_FILE_STREAM</b>	<b>ERROR: Bad file stream</b>
	<b>DESCRIPTION</b> Unable to obtain the required file stream.
	<b>ACTION</b> Check for the existence of the import/export file.
<b>-4006: NNFIEE_NAME_PROPERTY_CONFLICT</b>	<b>ERROR: Conflict with an existing Formatter element with the same name</b>
	<b>DESCRIPTION</b> A format component being imported conflicts with an existing component of the same name.
	<b>ACTION</b> If importing into a populated format database, rename the existing component and import again, or change the incoming component name in the source database and re-export.
<b>-4007: NNFIEE_INVALID_IE_MODE</b>	<b>ERROR: Invalid import/export mode; valid: EXPORT_BY_NAME, EXPORT_ALL, IMPORT)</b>
	<b>DESCRIPTION</b> An invalid mode has been specified on the command line or in the command file.
	<b>ACTION</b> Check the arguments passed to NNFie for correctness.
<b>-4008: NNFIEE_ATTEMPTING_TO_REEXPORT</b>	<b>ERROR: Attempting to re-export an element that has been exported</b>
	<b>DESCRIPTION</b> A component has been defined that references itself.
	<b>ACTION</b> Remove the circular reference to this component.
<b>-4010: NNFIEE_INVALID_FORMATTER_ELEMENT</b>	<b>ERROR: Invalid Formatter element type</b>

	<b>DESCRIPTION</b>	An unknown format component has been found. The file is corrupt.
	<b>ACTION</b>	Recover or recreate the export file.
<b>-4011: NNFIEE_INVALID_NNFIE_FILE</b>	<b>ERROR: Invalid NNFie file. Make sure the file was generated by NNFie</b>	
	<b>DESCRIPTION</b>	The specified file is incompatible.
	<b>ACTION</b>	Recreate or recover the export file.
<b>-4012: NNFIEE_INVALID_VERSION_NO</b>	<b>ERROR: Invalid NNFie version number</b>	
	<b>DESCRIPTION</b>	The version number found in the file is not supported.
	<b>ACTION</b>	Recreate the file using a supported version.
<b>-4013: NNFIEE_FAILED_TO_INVENTORY</b>	<b>ERROR: Failed to add to the I/E inventory</b>	
	<b>DESCRIPTION</b>	NNFie was unable to register the component as exported or imported.
	<b>ACTION</b>	Rerun the import/export.
<b>-4014: NNFIEE_NO_FORMATS_TO_EXPORT</b>	<b>ERROR: No formats to export</b>	
	<b>DESCRIPTION</b>	The format database does not contain any valid formats to export.
	<b>ACTION</b>	Create valid formats.
<b>-4015: NNFIEE_NOTHING_TO_IMPORT</b>	<b>ERROR: Nothing to import</b>	
	<b>DESCRIPTION</b>	The import file does not contain any format information.
	<b>ACTION</b>	Create an export file from a database that contains formats.
<b>-4018: NNFIEE_NNFIEERR_ALREADY_EXISTS</b>	<b>ERROR: NNFieerr already exists</b>	

## A Error Messages

---

	<b>DESCRIPTION</b>	The error file NNFile.err exists.
	<b>ACTION</b>	Remove the file NNFile.err and rerun.
<b>-4019:NNFIEE_IE_FILE_ALREADY_EXISTS</b>	<b>ERROR: I/E File already exists</b>	
	<b>DESCRIPTION</b>	The specified output file already exists.
	<b>ACTION</b>	Use a new export file name; move or rename the existing export file.
<b>-4020: NNFIEE_FAILED_TO_OPEN_DBMS_SESSION</b>	<b>ERROR: Failed to open DBMS session</b>	
	<b>DESCRIPTION</b>	NNFile was unable to use the Format Manager library.
	<b>ACTION</b>	Check the correctness of the installation of BEA eLink Information Integrator.
<b>-4021: NNFIEE_FAILED_TO_OPEN_FMGR</b>	<b>ERROR: Failed to initialize Formatter Manager</b>	
	<b>DESCRIPTION</b>	NNFile was unable to use the Format Manager library.
	<b>ACTION</b>	Check the correctness of the installation of BEA eLink Information Integrator
<b>-4022: NNFIEE_INVALID_CNTL_TYPE</b>	<b>ERROR: Invalid control type</b>	
	<b>DESCRIPTION</b>	An unknown format control has been found. The file is corrupt.
	<b>ACTION</b>	Recover or recreate the export file.

# Information Integrator Server Error Messages

<b>500: Unknown TUXEDO Buffer Type</b>	<b>ERROR: Unknown TUXEDO buffer type [%1]</b>
<b>DESCRIPTION</b>	The BEA eLink Information Integrator server received a request contained in an unsupported TUXEDO buffer type. The supported types are FML, FML32, STRING, and CARRAY. VIEW's and user-defined types are not supported
<b>ACTION</b>	Change the request to include supported buffer types only.
<b>501: Unknown Error</b>	<b>ERROR: An unknown error has occurred</b>
<b>DESCRIPTION</b>	An unknown internal error occurred.
<b>ACTION</b>	Try reinstalling BEA eLink Information Integrator on the server.
<b>502: Unknown Service Name</b>	<b>ERROR: An unknown service name [%1%2] was requested</b>
<b>DESCRIPTION</b>	A service alias that is not defined in the configuration file was used to contact the BEA eLink Information Integrator server.
<b>ACTION</b>	Change the service alias used to contact the server, or define the specified service alias in the configuration file.
<b>503: Configuration Error</b>	<b>ERROR: Configuration error</b>
<b>DESCRIPTION</b>	An error was encountered while processing the BEA eLink Information Integrator server configuration file.
<b>ACTION</b>	Check the TUXEDO user log file for the exact details of the error.



# B Calculating Sizing Data

This section describes the formulas for calculating space requirements in Formatter and Rules.

## Formatter Space Requirements

To calculate the total database space required for Formatter, you must estimate the number of objects used by Formatter. This calculation is the same for Oracle and Microsoft SQL Server databases.

Estimate the number of:

- Fields (fields are in flat input, flat output and compound formats)
- Literals
- Input controls
- Name/Value input field validation parameter pairs
- Output format controls
- Flat input formats
- Flat output formats
- Compound formats

Use the key below to interpret the formulas in the following sections:

F = Number of fields

D = Number of delimiters

## **B** *Calculating Sizing Data*

---

IPC = Number of input parse controls

NV = Name/value input field validation parameter pairs

OFC = Number of output format controls

FIF = Number of flat input formats

FOF = Number of flat output formats

CF = Number of compound formats

FSpace = Total space, in bytes, needed for Formatter

## **Oracle**

Use the following formula to determine the space needed for Formatter:

```
(1819 + (164 x F)) +  
(1819 + (52 x D)) +  
(1819 + (164 x IPC)) +  
(82 x NV) +  
(1819 + (563 x OFC)) +  
(1819 + (60 x FIF)) +  
(1819 + 66 x FOF)) +  
(1819 + (60 x CF)) =  
FSpace
```

## **Microsoft SQL Server**

Use the following formula to determine the space needed for Formatter:

```
(1821 + (168 x F)) +  
(1821 + (56 x D)) +  
(1821 + (200 x IPC)) +  
(84 x NV) +  
(1821 + (615 x OFC)) +  
(1821 + (88 x FIF)) +  
(1821 + (100 x FOF)) +  
(1821 + (84 x CF)) =  
FSpace
```

# Rules Space Requirements

To calculate the total database space required for Rules, you must estimate the number of objects used by Rules. This calculation is the same for Oracle and Microsoft SQL Server databases.

Estimate the number of:

- Application Groups
- Message types within each application group
- Rule names within each message type
- Expressions within each rule name
- Subscriptions within each rule name
- Actions within each subscription

Use the key below to interpret the formulas in the following sections:

AG = Number of application groups

MT = Number of message types within an application group

R = Number of rule names within each message type

A = Number of expressions within each rule

S = Number of subscriptions within each rule

AO = Number of actions within each subscription

RSspace = Total space, in bytes, needed for Rules

### **Oracle**

Use the following formula to determine the space needed for Rules:

$$\begin{aligned} & (1318 + (43 \times AG)) + \\ & (1322 + (13 \times MT)) + \\ & (1322 + (535 \times R)) + \\ & (1322 + (216 \times A)) + \\ & (1322 + (240 \times S)) + \\ & (1322 + AO) = \\ & RSpace \end{aligned}$$

### **Microsoft SQL Server**

Use the following formula to determine the space needed for Rules (the key to formula expressions follows the DB2 formula):

$$\begin{aligned} & (1322 + (47 \times AG)) + \\ & (1330 + (17 \times MT)) + \\ & (1330 + (601 \times R)) + \\ & (1330 + (242 \times A)) + \\ & (1330 + (260 \times S)) + \\ & (127 + AO) = \\ & RSpace \end{aligned}$$

# C Changing Database Sort Order

This section describes how to change the database sort order on Microsoft SQL Server systems.

## Microsoft SQL Server

The default sort order for Microsoft SQL Server is dictionary case-insensitive. For the Information Integrator server to take full advantage of case-sensitive naming conventions and operations, the sort order on the SQL Server must be changed to a binary sort order. Changing the sort order to binary provides a performance advantage over dictionary sort order.

Changing the sort order requires preparation and an understanding of SQL Server. Changing the sort order on SQL Server is easiest on an initial server install. It is more difficult on a server that is established with applications other than Information Integrator running. In ALL cases, make full backups of all databases, especially the master database, before proceeding.

## New Install of Microsoft SQL Server

The SQL Server setup program prompts for the sort order and the character set to use. Change the sort order to binary and proceed with the installation as usual.

## Upgrade Microsoft SQL Server from Version 4.2 and Earlier

The default sort order is set correctly for Information Integrator in SQL Server versions 4.2 and earlier. If your installation of SQL Server 6.5 is an upgrade from 4.2, changing the sort order may not be necessary. To verify that the sort order is set to binary, log on to the server and execute the `sp_helpsort` system stored procedure. This returns the character set along with the sort order being used by the server. If the sort order is not set to binary, follow the instructions in the next section to change the sort order on an established server.

### Established Microsoft SQL Server

Changing the sort order on an established server requires preparation and planning. If your server runs applications other than Information Integrator, verify that changing the sort order on the server will not affect the other applications also residing on that server.

Before performing this procedure, refer to Chapter 3 (*Rebuilding the Master Data* section) in the *Microsoft SQL Server Administrator's Companion*.

#### To change the sort order on the server:

1. Back up *all* databases on the server, especially the master database. This allows you to rebuild the server to its initial settings, if necessary.
2. BCP all of the data from all tables in all databases, since the database backup file contains the sort order used at the time of the backup. When the sort order is changed on the server, the backups are invalid.
3. To export the DDL from the databases, use the Enterprise Manager tool and do the following:
  - Choose Objects→Generate SQL scripts.
  - Check the All Objects box in the Scripting Objects section.
  - Generate a fill for each database.

4. To generate a script for every login and user on the server, use the Enterprise Manager tool and do the following:
  - Choose Objects→Generate SQL scripts.
  - Check all the boxes in the Security section.
  - Generate the script.
5. Shutdown the SQL Server.
6. To change the sort order on the SQL Server:
  - Choose the Setup icon for SQL Server. Two screens appear: a welcome screen, and then a screen indicating that SQL Server is already installed.
  - Choose Continue on each of these screens. The next screen contains a set of radio buttons.
  - Select Rebuild Master Database, then choose Continue. A warning screen appears indicating that rebuilding the master database will destroy all data.
  - Choose Resume. A screen appears that lets you change the sort order.
  - Select the sort order box, select the binary sort order option, and choose Continue. The setup program rebuilds the system databases using the specified sort order.
7. Restart the SQL Server.
8. Login to the server using the SA user ID.
9. Execute the `sp_helpsort` stored procedure to verify that the sort order was changed to binary.
10. Recreate the database devices and databases.
11. Run the script that was generated in Step 3 for each database.
12. Run the script that was generated in Step 4 for logins and users.
13. BCP the data back into the tables.

