Installation Guide

Informatica Identity Resolution
(Version 9.0.1 SP4)
# Contents

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Introduction</strong></td>
<td>5</td>
</tr>
<tr>
<td>Support Matrix</td>
<td>6</td>
</tr>
<tr>
<td>Platforms</td>
<td>6</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>6</td>
</tr>
<tr>
<td>Databases</td>
<td>6</td>
</tr>
<tr>
<td>IIR Components</td>
<td>7</td>
</tr>
<tr>
<td><strong>Software Prerequisites</strong></td>
<td>8</td>
</tr>
<tr>
<td>Network Protocols</td>
<td>8</td>
</tr>
<tr>
<td>ODBC Drivers</td>
<td>8</td>
</tr>
<tr>
<td>Browsers</td>
<td>8</td>
</tr>
<tr>
<td><strong>Software Installation</strong></td>
<td>9</td>
</tr>
<tr>
<td>Overview</td>
<td>9</td>
</tr>
<tr>
<td><strong>Post Installation Steps</strong></td>
<td>10</td>
</tr>
<tr>
<td>Overview</td>
<td>10</td>
</tr>
<tr>
<td>Database configuration</td>
<td>10</td>
</tr>
<tr>
<td>Test ODBC connectivity</td>
<td>10</td>
</tr>
<tr>
<td>Select a Web-Browser</td>
<td>10</td>
</tr>
<tr>
<td>Regression Test</td>
<td>10</td>
</tr>
<tr>
<td>Windows</td>
<td>11</td>
</tr>
<tr>
<td>UNIX</td>
<td>11</td>
</tr>
<tr>
<td>Install additional client software</td>
<td>11</td>
</tr>
<tr>
<td>Miscellaneous Topics</td>
<td>11</td>
</tr>
<tr>
<td>Enabling UDB</td>
<td>11</td>
</tr>
<tr>
<td><strong>Database Configuration</strong></td>
<td>13</td>
</tr>
<tr>
<td>General Information</td>
<td>13</td>
</tr>
<tr>
<td>Terminology</td>
<td>13</td>
</tr>
<tr>
<td>ODBC</td>
<td>14</td>
</tr>
<tr>
<td>SQL scripts</td>
<td>14</td>
</tr>
<tr>
<td>Optional Synchronizer Objects</td>
<td>14</td>
</tr>
<tr>
<td>Oracle</td>
<td>14</td>
</tr>
<tr>
<td>Cost Based Optimization</td>
<td>14</td>
</tr>
<tr>
<td>Creating Users and Tables</td>
<td>15</td>
</tr>
<tr>
<td>Recommended Oracle Instance Parameters</td>
<td>18</td>
</tr>
<tr>
<td>UDB</td>
<td>18</td>
</tr>
<tr>
<td>Database Parameters</td>
<td>18</td>
</tr>
<tr>
<td>Collate Identity</td>
<td>18</td>
</tr>
<tr>
<td>Create a User</td>
<td>19</td>
</tr>
<tr>
<td>Creating Tables</td>
<td>20</td>
</tr>
<tr>
<td>Select Privileges</td>
<td>20</td>
</tr>
<tr>
<td>User Defined Function</td>
<td>20</td>
</tr>
<tr>
<td>ODBC / CLI Driver</td>
<td>21</td>
</tr>
<tr>
<td>Common Configuration Problems</td>
<td>21</td>
</tr>
<tr>
<td>DB2 on z/OS</td>
<td>23</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>23</td>
</tr>
<tr>
<td>Create a User</td>
<td>24</td>
</tr>
<tr>
<td>Define an ODBC Data Source</td>
<td>24</td>
</tr>
<tr>
<td>Permissions</td>
<td>24</td>
</tr>
<tr>
<td>Create Tables</td>
<td>25</td>
</tr>
</tbody>
</table>
Sybase ASE ................................................................. 26
Create a User ............................................................ 27
Create Tables ............................................................. 27
Set the Bulkcopy DB Option .......................................... 28
Define the ODBC Data Sources ..................................... 28
Generic ODBC Databases ............................................. 29

ODBC Configuration 30
Syntax ................................................................. 30
ODBC Drivers .......................................................... 31
Oracle ................................................................. 31
Microsoft SQL Server ................................................. 32
Sybase ................................................................. 32
UDB ................................................................. 32
Testing Connectivity .................................................. 33
Database Connections ................................................ 33

UNIX Notes 34
Establishing an environment ........................................ 34
Test the ssapid utility ................................................ 34
Uninstall .............................................................. 34

Index 35
Introduction

This manual describes the installation of Informatica’s Identity Resolution product.
Support Matrix

Not all combinations of Platform, Operating System and DBMS are supported. Please refer to the Product Availability Matrix for a definitive listing.

Platforms

IIR Client and Server software is supported on the following hardware platforms:

- HP/PA-RISC
- Intel x86 and x86-64
- Itanium
- RS/6000
- SUN/SPARC

Operating Systems

IIR Client and Server software is supported on the following Operating Systems:

- AIX 5.1 and above
- HP-UX 11.11 and above
- LINUX (libc 2.3)
- Solaris 8 and above
- Windows 98/NT 4 (Client software only)

Databases

IIR Servers support the following Database Management Systems:

- Oracle (version 8.1.7 and above)
- Microsoft SQL Server (version 2000 and above)
- UDB (version 8 and above)
- DB2 (version 7) on z/OS (source access only)
- Sybase ASE (version 12.5 and above)
- SQL databases with generic ODBC drivers (unsynchronized source access only).
IIR Components

IIR requires the installation of Server Components, Client Components and optionally Developer Components.

IIR Core modules are installed together with the IIR Server Components. They include the System Loader, Table Loader, Update Synchronizer and IIR Nucleus.

IIR Server Components include the Search Server, Console Server, Rulebase Server and Connection Server.

IIR Client Components include the Console Client, Search Client, Rulebase Editor and the Custom Search Client API.

The optional IIR Developer Components include C and Perl client program examples. They are installed together with the IIR Client Components.

Depending on the user’s requirements, the Core, Server and Client Components may be installed on one computer, or separated with the Core and Server Components on one computer and the Client Components on another.

The Core and Server Components are typically installed on the same computer as the user’s database.
Software Prerequisites

This chapter describes the software prerequisites for an IIR installation.

Network Protocols

The IIR Clients and Servers require a TCP/IP network connection.

DNS must be installed, configured and available. `/etc/hosts`, `/etc/resolv.conf` and `/etc/nsswitch.conf` (or their equivalents) must be correctly setup. Reverse name lookups must yield correct and consistent results.

ODBC Drivers

IIR uses the ODBC protocol to access SQL databases. ODBC drivers for databases used as either a source or target must be pre-installed and configured according to the manufacturer’s specifications.

The only exception is the ODBC driver provided by Oracle Corp. It does not need to be installed because IIR ships with a custom driver named `ssaoci` for Oracle access, which works on both Windows and UNIX platforms. That is, IIR does not use the ODBC driver provided by Oracle.

Browsers

The HTTP server generates pages compatible with Netscape 4, Internet Explorer 5 and Firefox 1.0 (or later versions)
Software Installation

Overview

The installation process is divided into four main steps:

- Install the License Server,
- Obtain product license file(s) from Informatica,
- Install your licensed software, and
- Follow the product specific installation instructions to complete the installation process.

Instructions for the first three steps are found in the INFORMATICA IR PRODUCT INSTALLER manual. The Post Installation steps are described in the following chapters of this manual.
Post Installation Steps

Overview

The IIR server and client software components were installed by running the Informatica IR Product Installer. Once copied to disk, various customizations must be performed to complete and verify the installation.

IIR post-installation tasks include:

- Creation of database objects (schema, tables, etc.) as described in the Database Configuration section of this manual.
- Configuration and testing of ODBC connections as described in the ODBC Configuration section of this manual.
- Running a quick regression test to verify a successful server installation.
- Optionally install additional client software on other machines.
- Configuration of environment variable, #SSAWORKDIR#. This is the path to a directory that is to contain the log files. In case of Client-Only installation, before launching the UI, this variable needs to be modified to point to #SSAWORKDIR# of Server Components installation.

Database configuration

Ensure that the tasks described in the Database Configuration section of this manual have been completed.

Test ODBC connectivity

During installation you may have been prompted to set up and test an ODBC configuration file unless you already had an odbc.ini file configured from a previous installation.

If a new odbc.ini file was created, its validity must be tested by following the steps described in the Testing Connectivity section of the ODBC Configuration chapter.

Select a Web-Browser

On UNIX platforms, edit $SSABIN/ssabrz and uncomment one of the statements that sets the environment variable SSABRZ to an appropriate Web browser.

Regression Test

A short test is run to verify the installation by loading a predefined System and verifying its search and synchronization results.

Once these tests have finished successfully, IIR is ready for user defined systems. For more detailed information on running the Console Client/Server software, refer to the relevant sections in the OPERATIONS manual.

Before conducting the Regression test, ensure that the license server is running. Refer to the instructions in INFORMATICA IR PRODUCT INSTALLER manual to start the license server.
Windows

The Informatica IR Product Installer creates two shortcuts in the Informatica program group that are used to run the regression test.

First, start IIR Server (Configure mode) followed by the IIR Console Client (Admin mode). Follow the prompts and provide a user-id, password and service name (previously configured in odbc.ini).

UNIX

IIR Server processes are launched in Configure Mode in order to run a predefined System and verify its search and synchronization results.

First, establish environment variables as described in the Establishing an Environment section of the UNIX Notes chapter of this manual.

Bring the IIR Servers up in Configure Mode. For Sybase ASE:

\$SSABIN/idsup -inosync.dat -t$SSATOP/ids/systems

For other DBMSs:

\$SSABIN/idsup -itests.dat -tSSATOP/ids/systems

Start a Console Client in Admin mode. If running the Console Client on a Win32 computer, start the client as described in the Windows section above. If running the Console Client on the Unix computer, run the following script:

\$SSABIN/idsconc -a

When the Console Client starts for the first time, it will prompt for the user-id, password and service name of the Rulebase to be used for the test.

Install additional client software

Once the server installation is complete, additional client software may be installed on other machines, if necessary.

Miscellaneous Topics

Enabling UDB

Support - post-priori The Informatica IR Product Installer creates a shortcut named "IIR Server - Start" for launching the IIR Server. If UDB support was chosen, this shortcut will launch a batch file that starts the IIR Server (idsup.bat) in a UDB command window (d2cmd.exe); For example:

C:\SQLLIB\bin\db2cmd.exe /C C:\InformaticaIR\bin\idsup.bat
However, if UDB support was not chosen, the batch file is launched in a "normal" Windows Command Prompt (cmd.exe). For example

```
C:\WINNT\system32\CMD.EXE /C C:\InformaticaIR\bin\idsup.bat
```

If UDB support is required at a later stage, this shortcut must be manually modified to launch the db2cmd.exe instead of cmd.exe.
Database Configuration

This chapter describes the steps required to prepare a database for use by IIR software. Before starting, the following questions must be answered:

Which databases contain the User Source Tables to be indexed by IIR (known as a source-database)?

Which database is designated to contain the IIR Tables (known as a target-database)?

Is the Update Synchronizer to be used? The Update Synchronizer allows the IIR Tables to be kept synchronized with the User Source Tables (see the INTRODUCTION or OPERATIONS manual for more information).

General Information

Terminology

Database objects (tables, indexes, triggers, etc.) accessed and created by IIR fall into two logical groups:

- Source Database
- Target Database

The source and target databases may be the same physical database, or they may be different, but we make a distinction between them for this discussion.

Source Database (SSA_SRC)

The Source Database contains User Source Tables (UST). These are owned and maintained by an existing application system. IIR is used to create fuzzy indexes for these tables.

IIR accesses the USTs using a user id that has been given SELECT privileges on those tables. This user id is known as the SSA_SRC user id in the following discussion. The SSA_SRC user id is specified in the select_by or merge_from clauses to tell IIR how to connect to the Source Database.

If the IIR indexes are to be synchronized with changes made to the USTs using triggers, the SSA_SRC user id will need privileges to create triggers on the USTs. The triggers will store information about UST rows that have been added, deleted or updated into a Transaction Table. This table lives in the Source Database in SSA_SRC’s schema.

Target Database (SSA_TGT)

The Target Database contains Identity Tables (IDT) and Identity Indexes (IDX). They are created and maintained by IIR utilities using the SSA_TGT user id.

If the IDTs and IDXs are to be synchronized, a Two-Phase-Commit table is created in the SSA_TGT schema.

The Table Loader and Synchronizer use the SSA_SRC user id to access the USTs and Transaction Table, and the SSA_TGT user id to update the Target Database.
ODBC

IIR uses the ODBC protocol to access source and target databases. Only a limited subset of ODBC verbs are required for unsynchronized source access, so most generic ODBC drivers will suffice.

Synchronized source access requires triggers to be created on the source tables.

Target support requires tables to be created and mass loaded using native DBMS utilities. These capabilities require specific SQL dialects and DBMS extension features to be used. IIR currently supports the use of Oracle, Microsoft SQL Server, UDB/DB2 and Sybase ASE for synchronized source / target databases.

SQL scripts

SQL scripts are provided as a way of documenting the minimum set of privileges required. Since they are generic samples, they must be customized before use to meet site specific standards.

The scripts use a default userid of $SSA$, hereafter known as the "SSA userid". If a different userid is to be used, the scripts must be edited appropriately.

It is expected that the installation will be performed by an experienced DBA with the appropriate authority to create users and assign privileges.

The SQL scripts are located in database specific directories in the server installation’s \%SSATOP\\dbscript\\<db> directory, where <db> is ora, msq, syb or udb.

Please copy the scripts to another location prior to customizing them, since the original scripts may be overwritten by updated versions when a Fix CD is applied.

Optional Synchronizer Objects

The creation of Synchronizer objects for some databases is optional. However, if you intend to run the standard regression test as part of the installation process (which is highly recommended to confirm that your installation is working correctly), the Synchronizer objects must be created because the regression test will exercise the Update Synchronizer.

Oracle

IIR supports Oracle version 8.1.7 and above, although v9.2 (or above) is required to access Unicode data.

Cost Based Optimization

IIR has been designed to use the Cost Based Optimizer (CBO). Some Oracle features that IIR uses only work with the CBO (such as Index Organized Tables). Update Synchronization also requires the use of CBO when accessing the User Source Tables, as the SQL is too complicated to be well optimized by the Rule Based Optimizer (RBO).

If your existing applications have been optimized to use the RBO, you can enable CBO just for IIR by setting the database initialization parameter Optimizer_Mode=Rule and collecting statistics for the USTs. When IIR accesses the USTs, it specifies an optimizer hint of CHOOSE, so that if statistics exist, it will use the CBO. Existing applications will continue to use the RBO due to the default set by the initialization parameter.
Creating Users and Tables

The Update Synchronizer requires access to various Oracle features: triggers, procedures, sequences and tables. These are created by the supplied installation scripts.

The number of database objects created and privileges required depend on whether or not the database is a source or target database, and whether synchronization is required. The following table shows these dependencies:

<table>
<thead>
<tr>
<th>DB Type / Use</th>
<th>Privileges</th>
<th>IIR Objects</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source / NOSYNC</td>
<td>CREATE SESSION</td>
<td>IDS_FDT_META</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td>SELECT ON TABLE (UST)</td>
<td>IDS_FDT_RECID</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_CONV</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td>Target / NOSYNC</td>
<td>CREATE SESSION</td>
<td>IDS_FDT_META</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE</td>
<td>IDS_FDT_RECID</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td>UNLIMITED TABLESPACE</td>
<td>IDS_RB_GROUPS</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_CONV</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td>Source / NOSYNC</td>
<td>CREATE SESSION</td>
<td>IDS_FDT_META</td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE</td>
<td>IDS_FDT_RECID</td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td></td>
<td>UNLIMITED TABLESPACE</td>
<td>IDS_RB_GROUPS</td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE SEQUENCE</td>
<td>IDS_UPD_SYNC_TXN</td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE PROCEDURE</td>
<td>&lt;Triggers&gt;</td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE ANY TRIGGER</td>
<td></td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td></td>
<td>SELECT ON TABLE</td>
<td></td>
<td>updsyncu.sql</td>
</tr>
<tr>
<td>Target / SYNC</td>
<td>CREATE SESSION</td>
<td>IDS_FDT_META</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE TABLE</td>
<td>IDS_FDT_RECID</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td>UNLIMITED TABLESPACE</td>
<td>IDS_RB_GROUPS</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_2PC</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_NSA</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_CONV</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>idstbora.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>idstbora.sql</td>
</tr>
</tbody>
</table>

1 This is not intended to be an exhaustive list. Only major components are listed. Refer to the scripts for a definitive list of database objects created at install time. The OPERATIONS manual, Database Object Names section provides further details.
Creating Users on Source Databases

Create a new SSA_SRC user using the supplied script `idsuseru.sql` in each Source database. This is the set of privileges required:

```
CREATE USER ssa IDENTIFIED BY ssa
DEFAULT TABLESPACE users;
GRANT CREATE SESSION TO ssa;
GRANT SELECT ON <UST> TO ssa;
```

If the source tables are to be synchronized, the following privileges are also required:

```
GRANT CREATE TABLE TO ssa;
GRANT UNLIMITED TABLESPACE TO ssa;
GRANT CREATE SEQUENCE TO ssa;
GRANT CREATE PROCEDURE TO ssa;
GRANT CREATE ANY TRIGGER TO ssa;
```

The script must be modified to specify a user id, password and default table space. Replace the strings `#UID#`, `#PWD#` and `#TS#` with the selected values.

The script may be modified to specify additional privileges and/or restrict the amount of table space quota. The default is to grant unlimited table space quota.

The IIR System Loader creates triggers on the User Source Tables when it starts extracting data from them. It connects to the Source database using the SSA_SRC user id. As the source tables are usually owned by another user, the SSA_SRC user id must have `CREATE ANY TRIGGER` privilege (at least while the Table Loader runs).

The SSA_SRC user id used to read User Source Tables requires `SELECT` privileges on those tables. Refer to the Design Guide manual, User Source Table section for details.

Creating Users on Target Databases

If the IIR tables are to exist in a separate database from the User Source tables create a new SSA_TGT user "ssa" in the target database using the supplied script `idsuseri.sql`. The set of privileges required are:

```
CREATE USER ssa IDENTIFIED BY ssa
DEFAULT TABLESPACE users;
GRANT CREATE SESSION TO ssa;
GRANT CREATE TABLE TO ssa;
GRANT UNLIMITED TABLESPACE TO ssa;
```

The script must be modified to specify a user id, password and default table space. Replace the strings `#UID#`, `#PWD#` and `#TS#` with the selected values.

The script may be modified to specify additional privileges and/or restrict the amount of table space quota. The default is to grant unlimited table space quota.
Creating IIR Control Objects

To create the IIR control objects, run the `idstbora.sql` script using the SSA_SRC or SSA_TGT userid/password (as appropriate) for each

- Source database that will be synchronized, and
- Target database

Creating Synchronizer Objects on Source Databases

In each source database containing User Source Tables to be synchronized (even if the target tables are in the same database),

Run the `updsyncu.sql` script with the SSA_SRC user id.

**Note:** The `updsyncu.sql` script should never be run against a UST database after the synchronizer has been used to synchronize that UST database. Doing so will result in a loss of synchronization. The Synchronizer can detect this situation and will issue an error message if this occurs.

Create public synonyms for the objects created in (i) to enable them to be shared. This is done by running `updsyncg.sql` using a DBA user id (or any user id that has `CREATE PUBLIC SYNONYM` authority). Before running this script, ensure that the SSA_SRC user id specified in the script (defaults to "ssa") is correct.

Creating Synchronizer Objects on Target Databases

In each target database to be synchronized, run the following script using the SSA_TGT user id:

- `updsynci.sql`

Creating IIR Utilities on Target Databases

In each target database, run one of the following scripts (dependent on your Oracle version) using the SSA_TGT user id:

- `idsconv8.sql` (for Oracle 8), or
- `idsconv9.sql` (for Oracle 9 and above).

Creating PL/SQL API components on Target Databases (optional)

Prior to running these scripts, each one must be edited to adjust the `CONNECT` and `SPOOL` statements. `#SSA#` must be replaced by the connection string of the target database (e.g. `ssa/ssa@service`). `#SSAWORKDIR#` is the path to a directory that is to contain the log file.

Run the following scripts in each target database that will be searched using the PL/SQL Search API.

- `%SSABIN%\ssasep.sql`
- `%SSABIN%\ssaseb.sql`
- `%SSABIN%\ssasec.sql`

Refer to the *Calling from PL/SQL* section of the DEVELOPER GUIDE manual for complete instructions.
Recommended Oracle Instance Parameters

OPTIMIZER_DYNAMIC_SAMPLING (10g) should be set to 0 to avoid dynamic statistics collection during queries. Not doing so is likely to impact query execution times, particularly during AnswerSet creation.

Set MAX_COMMIT_PROPAGATION_DELAY to 0 to avoid delays in synchronizing AnswerSets committed on one RAC instance and read from another.

UDB

UDB must be installed prior to the installation of IIR.

Two sets of scripts are provided and the appropriate set should be used based on your database platform. The scripts are for

- UDB Version 8 or later (*.sql)

Database Parameters

IIR requires the following database configuration parameters to be set:

APPLHEAPSZ to at least 1024

MAXAPPLS to at least 256

IIR requires the following database manager configuration parameters to be set:

UDF_MEM_SZ to at least 256. Note: with DB2 version 8, the UDF_MEM_SZ configuration parameter became ASLHEAPSZ.

Collate Identity

When using UDB version 8, the COLLATE IDENTITY option must be enabled at database creation time.

The Collation Sequence can be changed from its default value (System) to the required one (Identity), with the Create Database Wizard or the command line.

Using the Create Database Wizard

At Step 6 (Specify the locale for this database), check Identity instead of System, which is the default.

Using the command Line

CREATE DATABASE SSA_IDS ON C: USING CODESET 1252 TERRITORY AU COLLATE USING IDENTITY;
Create a User

Create a new user "ssa" with password "SSA". As a UDB database server uses operating system authentication, the user must be created using operating systems tools. The details of this process vary from platform to platform. See your System Administrator for assistance.

The SQL scripts mentioned in the subsequent sections assume the userid is 'ssa'. If this userid is changed each script will also need to be modified.

UDB has two bulk load mechanisms, LOAD and IMPORT. LOAD is the faster mechanism. By default IIR uses the LOAD mechanism but this requires that the IIR user has one of the following privileges:

- sysadm
- dbadm

If you wish to use the IMPORT utility instead of the LOAD facility, the Loader-Definition option of Conventional-Path must be specified in your SDF.

<table>
<thead>
<tr>
<th>DB Type / Use</th>
<th>Privileges</th>
<th>IIR Objects</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source / NOSYNC</td>
<td>CONNECT</td>
<td>-</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td>Target / NOSYNC</td>
<td>CONNECT</td>
<td>IDS_FDT_META</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_FDT_RECID</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>Relate</td>
</tr>
<tr>
<td>Source / SYNC</td>
<td>CONNECT</td>
<td>IDS_FDT_META</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_FDT_RECID</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_TXN</td>
<td>updsyncu.701</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
<td>&lt;Rulebase&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>Relate</td>
</tr>
<tr>
<td>Target / SYNC</td>
<td>CONNECT</td>
<td>IDS_FDT_META</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_FDT_RECID</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbubd.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_2PC</td>
<td>updsynci.701</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_NSA</td>
<td>updsynci.701</td>
</tr>
<tr>
<td></td>
<td>LOAD</td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>Relate</td>
</tr>
</tbody>
</table>
Creating Tables

In each database that will hold either IIR Tables or User Source Tables run the `idstbudb.xxx`, the `updsyncu.xxx` and the `updsynci.xxx` scripts with the SSA userid to create the IIR control tables and the Update Synchronizer tables. For example, the `idstbudb.sql` script would be run as:

```
db2 -v -t -f idstbudb.sql -z idstbudb.log +o
```

when creating tables on a UDB or DB2 (z/OS) database.

Note: Unlike Oracle, the `udbsyncu.sql` and `udbsynci.sql` scripts need to be run against all UDB databases regardless of whether they are source or target, synchronized or not. Otherwise, errors will occur at bind time.

Note: The `udbsyncu.sql` script should never be run against a UST database after the synchronizer has been used to synchronize that UST database. Otherwise a loss of synchronization will result.

Select Privileges

The userid used to access User Source Tables must be granted `SELECT` privileges on those tables. Refer to the DESIGN GUIDE manual, `User-Source-Table` section for details.

If the source tables are to be synchronized, then the userid must be granted `ALTER` privileges as well.

User Defined Function

Install the User Defined Function (UDF) on all source and target databases.

To allow IIR to synchronize with the source database, it is necessary for a source environment to be configured to run triggers and stored procedures/user defined functions.

Note: The installation process on UNIX platforms requires the use of some environment variables which are defined later in the install process. This step should be deferred until you are instructed to perform this step in the `UNIX Installation` section of the manual. It is documented here to ensure that the UDF installation procedure is in only one place in the manual.

Windows

Copy the dynamic load library `ssaudf16.dll` from the `lib` directory on the installation CD to the UDB server’s shared library directory (which is called `function` and is normally found under `sqlib`).

```
copy %SSALIB\ssaudf16.dll %DB2PATH%\function
```

UNIX

Copy the shared library `ssaudf16` found in the `lib` directory of the installation to UDB’s shared library directory. (You can pick up the environment variable `DB2DIR` by running `db2profile`.) For example:

```
cp $SSALIB/ssaudf16 $DB2DIR/function
```
ODBC / CLI Driver

A UDB/DB2 CLI driver must be installed. Refer to the relevant UDB/DB2 manuals for full details.

**UNIX**

Edit the `db2cli.ini` file, which can be found in `$INSTHOME/cfg`. Add the following lines:

```ini
[data-source]
DBALIAS=alias
```

where `data-source` is the name of your data source and `alias` is the database alias as you would specify it on a `db2 connect` statement.

**Windows**

Go to START menu and select Settings and Control Panel

Select Data Sources (ODBC) (found under Administrative Tools)

Push Add…

Select IBM DB2 ODBC from the list of possible database sources

Input your data source name, database alias and an optional description. Press OK

**Common Configuration Problems**

When using UDB for the first time, you may encounter some problems as a result of an incomplete UDB installation.

**SQL Code -444**

If the Update Synchronizer reports:

```
SQL07323N An error occurred in a triggered SQL statement in trigger "SSA.IDS0000U"
Information returned from the error includes SQLCODE "-444",
SQLSTATE "42724" and message tokens
"*\_VARCHAR|SQL050704134911400|...qllib/function". SQLSTATE=09000
```

Run a command manually to get the reason code:

```
db2 values ids_comp_varchar (SMALLINT(3),'fred','tony','henry'
 ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' ,'' )
```

Check that the UDF has been copied in the function directory and that the library path is correctly set to include the correct `lib` directory.
When the following error is encountered:

```
SQL1224N A database agent could not be started to service a request,
or was terminated as a result of a database system shutdown or a force command.
```

First, verify that:

- The database manager has been started on the database server
- The database manager was not stopped
- The database agent was not forced off by the system admin
- The database manager has not already allocated the maximum number of agents
- The database agent was not terminated due to an abnormal termination of a key database manager process.

The following instructions allow you to connect to your local machine as though it was a remote client.

On the client:

Catalog the node, for example:

```shell
code db2 catalog tcpip node <NodeName> remote <hostname> server db2cDB2
```

The server name must match the name of the DB2 connection port service which you specified in `/etc/services`.

Verify that the catalog command worked properly:

```shell
db2 list node directory
```

Catalog the database as follows:

```shell
db2 catalog db <database name> as <database alias>
db2 uncatalog db <database name>
db2 catalog db <database alias> as <database name> at node <nodename>
```

(this will prevent the SQL1334N error that can occur when the database name is the same as the database alias)

Stop and start DB2.

```shell
db2stop
db2start
```
DB2 on z/OS

IIR supports DB2 on z/OS as a source database only. Two User Defined Functions (UDF) are provided in a dataset file named `ssa.xmit.ids901.ssaufdf`. This file is not supplied as part of the standard release package and must be requested from Global Customer Support if required. Transfer the dataset to your mainframe system using ftp in binary mode. For example:

```
ftp> quote site lrec=80 recfm=fb blksize=3120 vol=ssawrk cylinders
ftp> bin
ftp> put ssa.xmit.ids901.ssaufdf 'ssa.xmit.ids901.ssaufdf'
```

**Note:** When using the z/OS ftp client, use the LOCSITE command instead of "quote site".

The dataset is provided in TSO TRANSMIT format and must be unpacked using the TSO RECEIVE command. For example:

```
receive indsn('ssa.xmit.ids901.ssaufdf') da('ssaudf.load')
```

Alternatively, the TSO command may be run from JCL by invoking the TSO interpreter `IKJEFT1A`. For example:

```
//SSAINS JOB (1),IEBCOPY,NOTIFY=&SYSUID
//STEP1 EXEC PGM=IKJEFT1A
//SYSTSFRMT DD SYSOUT=* 
//SYSTSIN DD *
RECEIVE INDSN('SSA.XMIT.IDS901.SSAUDF')
DA('SSAUDF.LOADD') VOLUME(SSA001)
/*
/*
```

Now copy the two PDS members (named `SSAUDF` and `II4TOUL`) into the appropriate DB2 load library (probably called `DB2HLQ.RUNLIB.LOAD`) so that they can be called by DB2.

Microsoft SQL Server

IIR supports Microsoft SQL Server 2000 and above.

Microsoft SQL must be installed prior to the installation of IIR.

Two sets of scripts are provided and the appropriate set must be used depending on your Microsoft SQL Server version:

- Microsoft SQL Server 2000 (*.sql)
- Microsoft SQL Server 2005 or 2008 (*.msq)
Create a User

Create a new user named "ssa" with a password of "ssa". The user must not be a database owner. A database owner is any member of the

- `sysadmin` server role, or
- `db_owner` fixed database role. Database owners are mapped to the special user called `dbo` and any tables created by them will appear in the database as if they were created by user `dbo`. If the ssa user is a database owner, the IIR will not be able to find any database objects that it creates.

The ssa user will need to create tables and stored procedures. Use Enterprise Manager to set up the necessary authority by right clicking on Database | Properties | Permissions. Check "Create Table" and "Create SP".

The IIR Table Loader creates triggers on the User Source Tables. Therefore, the ssa userid used to access the USTs requires the `db_ddladmin` role to allow it to issue DDL commands.

SQL Server 2005 users will need to create a schema named ssa and define it to be the default schema for user ssa.

Define an ODBC Data Source

An ODBC Data Source must be defined on the machine that the Search Server will be running on to allow it to connect to SQL Server. It is defined with the ODBC Data Source Administrator, which is usually started from Control Panel | Administrative Tools | Data Sources (ODBC).

To create an ODBC Data Source, select the "User DSN" tab and click Add. If an "SQL Server" driver is not on the list of available drivers, ensure that Microsoft SQL Server client software has been installed on the machine.

Most defaults are suitable, but you must:

- Specify a Data Source Name. This name is known as the "Service Name" in IIR terminology.
- Specify the "SQL Server authentication" option, connecting as user ssa.
- Specify a "default database" for the user.
- Ensure that "Use regional settings when outputting currency, numbers, dates and times" is not checked.
- Verify that the user can connect by clicking "Test Data Source…"

All access to database servers is via ODBC, with the exception of the IIR Table Loader that bypasses ODBC to directly load data into the Microsoft SQL Server. ODBC Data Sources need to be created for each database server / database that you wish to access.

Permissions

The ssa user must have select permission on all User Source Tables that your systems will access. To grant select permission, launch Enterprise Manager, locate the table(s) under the databases folder, right click on Properties for the table(s), then the Permissions tab. Check the select permission for the ssa user.
The synchronizer places triggers on the User Source Tables (UST), which execute the stored procedure `IDS_UPD_SYNC`. All users updating the UST must have execute permission on this stored procedure. To grant "execute" permission, launch Enterprise Manager, drill down to the database, locate the stored procedure `IDS_UPD_SYNC` in the list of stored procedures, right click on it and select Properties and then press the Permissions tab. Check the "exec" permission for the user.

<table>
<thead>
<tr>
<th>DB Type / Use</th>
<th>Roles</th>
<th>IIR Objects</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source / NOSYNC</td>
<td>db_datareader</td>
<td>-</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td>Target / NOSYNC</td>
<td>db_datareader</td>
<td>IDS_FDT_META</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td>db_datawriter</td>
<td>IDS_FDT_RECID</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td>db_ddladmin</td>
<td>IDS_RB_GROUPS</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>Relate</td>
</tr>
<tr>
<td>Source / SYNC</td>
<td>db_datareader</td>
<td>IDS_FDT_META</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td>db_datawriter</td>
<td>IDS_FDT_RECID</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td>db_ddladmin</td>
<td>IDS_RB_GROUPS</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_TXN</td>
<td>mqsyncu.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC</td>
<td>mqsyncep.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Triggers&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Relate</td>
</tr>
<tr>
<td>Target / SYNC</td>
<td>db_datareader</td>
<td>IDS_FDT_META</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td>db_datawriter</td>
<td>IDS_FDT_RECID</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td>db_ddladmin</td>
<td>IDS_RB_GROUPS</td>
<td>idstbmsq.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC</td>
<td>mqsyncri.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_2PC</td>
<td>mqsyncri.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>relate</td>
</tr>
</tbody>
</table>

Create Tables

Two sets of scripts are supplied for use with Microsoft SQL. One set is for Microsoft SQL 2000, and ends with the extension " .sql". The other set is for Microsoft SQL 2005 and 2008 and ends with the extension " .msq".

Several scripts must be customized and run to create control tables for IIR.
In each database that will hold either IIR Tables or User Source Tables run the `idstbmsq.sql` script with the `ssa` userid, password and service:

```
osql -i idstbmsq.sql -U <userid> -P <password> -S <server>
```

where `-U` represents the SSA userid, `-P` is the corresponding password and `-S` is the Microsoft SQL Server name.

Do this in each database containing User Source Tables to be synchronized (even if the IIR Tables are in the same database).

Ignore the following warning:

Warning: The table 'IDS\_FDT\_META' has been created but its maximum row size (9171) exceeds the maximum number of bytes per row (8060). INSERT or UPDATE of a row in this table will fail if the resulting row length exceeds 8060 bytes.

Customize the `msqsynccp.sql` script by modifying the INSERT statement at the end of the script to change the database name from "master" to the database containing the tables created by `msqsyncu.sql`. The stored procedure is run by the users who update the source tables that have IIR triggers on them. As these users may or may not have the same default database as SSA, the database is explicitly identified in the stored procedure.

Run the `msqsynccu.sql` and `msqsynccp.sql` scripts with the SSA userid in each Source database.

Run the `msqsyncli.sql` script with the SSA userid in each Target database that will be synchronized.

Creating PL/SQL API components on Target Databases (optional)

The procedure outlined below has to be followed on target database that will be searched using the PL/SQL Search API.

Copy the DLL `ssasexp.dll`, to the directory containing the standard SQL server DLL files (`C:\Program Files\Microsoft SQL Server\MSSQL\Binn` for MS SQL 2000). Copy `ssaiok.dll` and `ssasec.dll` into a directory on the MS SQL server’s default path.

To install the DLL, from the `%SSATOP%\MSSQL` directory run:

```
osql -U sa -P password -S service -n -i ssasexp.sql
```

**Note:** `msqsynccu.sql` should never be run against a UST database after the synchronizer has been used to synchronize that database. Doing so will result in a loss of synchronization.

Sybase ASE

IIR supports Sybase ASE 12.5 and above. Sybase must be installed prior to the installation of IIR.
Create a User

Create a new SSA_SRC user in each Source database. The supplied script, `sybuseru.sql`, may be used for this purpose.

This user must **not** be a database owner. A database owner is any member of the

- `sysadmin` server role, or
- `db_owner` fixed database role.

Database owners are mapped to the special user called `dbo` and any tables created by them will appear in the database as if they were created by user `dbo`. If the `ssa` user is a database owner, the IIR will not be able to find any database objects that it creates.

Create Tables

Several SQL scripts must be customized and run to create control tables for IIR.

- In each database that will hold either IIR Tables or User Source Tables run the `idstbsyb.sql` script with the SSA_SRC userid, password and service:

  
  ```
  isql -i idstbsyb.sql -U <userid> -P <password> -S <server>
  ```
  
  where `-U` represents the SSA userid, `-P` is the corresponding password and `-S` is the Sybase ASE Server name.

Do this in each database containing User Source Tables to be synchronized (even if the IIR Tables are in the same database).

Ignore the following warning:

```
Warning: Row size (9163 bytes) could exceed row size limit, which is 8108 bytes.
```

- Customize the `sybsyncp.sql` script by modifying the two INSERT statements at the end of the script to change the database name and user from "master.ssa" to the database and user containing the tables created by `sybsyncu.sql`. The stored procedure is run by the users who update the source tables that have IIR triggers on them. As these users may or may not have the same default database as SSA, the database is explicitly identified in the stored procedure.

- Run the `sybsyncu.sql` and `sybsyncp.sql` scripts with the SSA_SRC userid in each Source database.

- Run the `sybsynci.sql` script with the SSA_SRC userid in each Target database that will be synchronized.

**Note:** `sybsyncu.sql` should never be run against a UST database after the synchronizer has been used to synchronize that database. Doing so will result in a loss of synchronization.

The number of database objects created and privileges required depend on whether or not the database is a source or target database, and whether synchronization is required. The following table shows these dependencies:
<table>
<thead>
<tr>
<th>DB Type / Use</th>
<th>Privileges</th>
<th>IIR Objects 2</th>
<th>Created by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source / NOSYNC</td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Target / NOSYNC</td>
<td>CREATE TABLE</td>
<td>IDS_FDT_META</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_FDT_RECID</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td>CREATE PROCEDURE</td>
<td>IDS_FDT_META</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_FDT_RECID</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_TXN</td>
<td>sybsyncu.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_IDENT</td>
<td>sybsyncu.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Triggers&gt;3</td>
<td>Table Loader</td>
</tr>
<tr>
<td>Source / NOSYNC</td>
<td>CREATE TABLE</td>
<td>IDS_FDT_META</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE PROCEDURE</td>
<td>IDS_FDT_RECID</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_TXN</td>
<td>sybsyncu.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_IDENT</td>
<td>sybsyncu.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDLs&gt;</td>
<td>relate</td>
</tr>
<tr>
<td>Target / SYNC</td>
<td>CREATE TABLE</td>
<td>IDS_FDT_META</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td>CREATE PROCEDURE</td>
<td>IDS_FDT_RECID</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_RB_GROUPS</td>
<td>idstbsyb.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_2PC</td>
<td>sybsynci.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS_UPD_SYNC_NSA</td>
<td>sybsynci.sql</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;Rulebase&gt;</td>
<td>Console</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDTs&gt;</td>
<td>Table Loader</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;IDXs&gt;</td>
<td>Table Loader</td>
</tr>
</tbody>
</table>

**Set the Bulkcopy DB Option**

The Target database must have the `select into/bulkcopy/pllsort` option enabled. This can be done using the `sp_dboption` utility.

**Define the ODBC Data Sources**

An ODBC Data Source must be defined on the machine that the Search Server will be running on, one for each Sybase ASE server that the Search Server will connect to.

---

2 This is not intended to be an exhaustive list. Only major components are listed. Refer to the scripts for a definitive list of database objects created at install time. The OPERATIONS manual, `Database Object Names` section provides further details.

3 In the case of Sybase ASE, Triggers are not automatically added by the Table Loader. Instead, Table Loader writes the required Trigger code to a text file. The DBA is responsible for incorporating this generated code into the Source Table’s triggers.
**Windows**

On Windows an ODBC Data Source is defined by using the ODBC Data Source Administrator, which is usually started from Control Panel | Administrative Tools | Data Sources (ODBC).

To create an ODBC Data Source, select the "User DSN" tab and click Add.

Select "Adaptive Server Enterprise" from the list of available drivers and click Finish. (If an "Adaptive Server Enterprise" entry is not in the list of available drivers, ensure that Sybase ASE client software has been installed on the machine.)

In the tabbed dialog which is presented next, most defaults are suitable, but you must:

- Specify a Data Source Name. This name is known as the "Service Name" in IIR terminology.
- Specify the "Server Name (ASE Host Name)".
- Specify the "".
- Verify that the user can connect by clicking "Test Data Source…"

All access to database servers is via ODBC, with the exception of the IIR Table Loader that bypasses ODBC to directly load data into the Microsoft SQL Server. ODBC Data Sources need to be created for each database server / database that you wish to access.

**Unix**

On Unix an ODBC Data Source is defined by modifying the file `odbc.ini` in the directory `$SYBASE` (i.e. the top level directory where the Sybase ASE client software is installed, e.g. `/opt/sybase`). An entry must be added to this file for each Data Source to be used. Here is a sample entry:

```
[sybase_dsn]
description=Sybase ASE 15.0 On SSA55
server=SYB150LINX86
port=5000
```

`sybase_dsn` is the Data Source name.

`server` and `port` define the connection parameters to be used by the ASE ODBC Driver to connect to the ASE Server.

**Generic ODBC Databases**

Generic ODBC drivers may be used to access ODBC compliant databases for unsynchronized source access.

The ODBC drivers for your DBMS must be installed and configured appropriately prior to the installation of IIR. `%SSABIN%\odbc.ini` must be configured to include an entry for each Data Source Name. Refer to section `ODBC Configuration` on page 30.
ODBC Configuration

IIR uses ODBC to access source and target databases. At run time, the database layer will attempt to load an appropriate ODBC driver for the type of database to be accessed. The name of the driver is determined by reading a file named odbc.ini and locating a configuration block matching the database service specified in the connection string. For example, the database connection string odb:99:scott/tiger@ora920 refers to a service named ora920.

A configuration block for ora920 might look like this. The service name appears in square brackets:

```
[ora920]
ssadriver = ssaoa19
ssauunixdriver = ssaoa19
server = ora920.mydomain.com
```

The odbc.ini file is located by searching a list of directories, specified using environment variables, in the following order:

1. ODBCINI
2. HOME
3. SSABIN

We recommend creating the primary configuration file in SSABIN and using ODBCINI or HOME as overrides, if necessary.

Syntax

A configuration block has the following form:

```
[Service_Name]
DataSourceName = ODBC_DSN
ssadriver = ODBC_Driver
ssauunixdriver = ODBC_Driver
server = Native_DB_Service
```

where

Service_Name is the name of the service used in the IIR connection string. That is, the name following the @ symbol. Service_Name is an arbitrary name (user selectable).

ODBC_DSN is the Data Source Name (DSN) defined in your ODBC configuration. It is an optional parameter. If not specified, ODBC_DSN defaults to Service_Name. All database drivers, except Oracle, require a valid DSN. Therefore, if this parameter is omitted, Service_Name must be a valid DSN.

ODBC_Driver is the name of the ODBC driver used to communicate with the DSN. The driver name must not include a path or extension (or lib prefix on UNIX). The directories containing the driver and its dependent libraries must be included in your path. The ssadriver tag is used to specify the name of the Windows version of the driver, whereas ssauunixdriver is used on UNIX platforms.
Native_DB_Service is the name of the database service configured in your DBMS setup. IIR specifies the Native_DB_Service name when running database utilities (such as the command line SQL processor and Mass-Loader) which connect to the database using proprietary DB protocols instead of ODBC. For Oracle databases, use the Net*8 TNS alias name defined in tnsnames.ora. For UDB, specify a Database Alias from the list displayed by db2 list database directory and for MSQ/Sybase, use the server name associated with the -S parameter of the bcp utility.

A sample configuration file named odbc.ini.ori is found in the bin directory of the Server components installation.

ODBC Drivers

Most DBMSs provide a generic ODBC driver for data access. In some cases, drivers are not available on all platforms. For example, Oracle does not provide an ODBC driver for UNIX platforms, despite shipping one for Windows. Microsoft’s ODBC Driver and Driver Manager are only available on the Windows platform. UDB provides drivers for both Windows and UNIX (although named differently), but no Driver Manager.

To solve these problems, IIR provides

• a Driver Manager on all platforms, and
• Oracle drivers for the Windows and UNIX platforms.

Installing and configuring ODBC drivers is platform and database dependent. Unless the driver is provided by IIR (as is the case for Oracle), you should follow the instructions provided by your database manufacturer in order to install them.

On Windows platforms, use Control Panel | Administrative Tools | Data Sources (ODBC) to create a DSN and associate it with a driver and database server.

Oracle

The IIR Oracle driver is named %SSABIN%\ssaoci{8|9}.dll (or $SSABIN/libssaoci{8|9}.s{o|o}1) on UNIX. This driver works "out-of-the-box". There are no special setup requirements, other than adding configuration blocks to your odbc.ini file.

The ODBC_Driver name may be either ssaoici8 or ssaoici9. The former must be used with Oracle 8 client libraries and does not support Unicode or LOB data. The latter is used with Oracle 9 (or above) client libraries and supports both Unicode and LOB data.

Oracle 10g

When using the ssaoici9 driver with Oracle 10g client software, the connectivity test (see Testing Connectivity below) may fail on some UNIX platforms. This occurs because the driver has been linked with libclntsh.so.9.0, which is not distributed with Oracle 10g. Oracle normally provides backward compatibility by adding symbolic links to redirect requests for older versions of the library to the current version. Unfortunately, by default, this practice is restricted to minor versions only (e.g. 9.0 ⇒ 9.2). To overcome the problem, locate the appropriate Oracle lib directory (lib, lib32 or lib64) and add a symbolic link. This allows the dependency for libclntsh.so.9.0 to be satisfied by the current Oracle 10 client library (which is compatible). For example,

```
cd $ORACLE_HOME/lib32
ln -s ./libclntsh.so libclntsh.so.9.0
```
Example Configuration

[ora10g]
ssadriver = ssacoi9
ssaunixdriver = ssacoi9
server = ora10g.mynet8tns.name

Microsoft SQL Server

Microsoft provides a Windows ODBC driver named sqlsrv32. It is configured by adding a new Data Source Name (DSN) using Control Panel | Administrative Tools | Data Sources (ODBC). Refer to the appropriate Microsoft manuals for specific details.

The ODBC_Driver name is sqlsrv32 and the Native_DB_Service is the server name (-S parameter of the osql and bcp utilities).

[production]
DataSourceName = msg2003
ssadriver = sqlsrv32
server = mySQLServer

The SQL Server Native Client (SQLNCLI - sqlncli.dll) may be used as an alternative driver. The SQLNCLI and the SQL Server Management Studio can be installed using the SQL Server Tools MSI. The Management Studio deploys bcp but does not deploy the SQL Server database engine.

Sybase

Sybase provides ODBC drivers, named sybdrvodb, for multiple platforms. Refer to the appropriate Sybase manuals for installation specifics.

A sample odbc.ini configuration block appears below:

[production]
DataSourceName = ase150
ssadriver = sybdrvodb
ssaunixdriver = sybdrvodb
{server = mySybaseServer

UDB

IBM provides ODBC drivers for both Windows and UNIX platforms, named db2cli and db2 respectively. Basic installation instructions are found in the ODBC / CLI Driver section of the UDB Pre-requisites chapter this manual. Refer to your UDB manual set for full details.

A sample odbc.ini configuration block appears below:

[test-udb]
DataSourceName = udb8
ssadriver = db2cli
ssaunixdriver = db2
server = UDB_database_alias
Drivers on AIX platforms

On AIX platforms the UDB driver is \texttt{libdb2.a(shr.o)} for 32 bit installations and \texttt{libdb2.a(shr_64.o)} for 64 bit installations.

Testing Connectivity

The \texttt{dblist} utility may be used to test your ODBC configuration by connecting to a database whose connection string is provided with the \texttt{-d} parameter. An example of the output associated with a successful connection appears below:

```bash
$SSABIN/dblist -c -dodb:99:ssa/SSA@ora920
Maximum connections per module: 1024
Linked databases: odb: sdb:
Driver Manager: 'Informatica ODBC Driver Manager <revision>'
ODBC Driver: 'ssaoci9 SSADB8 2.7.0.00GCCLIBC2Feb 1 2010 14:22:57'
DBMS Name: 'Oracle DBMS (9.2.0.6.0)'
Native DB type: 'ora'
*** Connection successful ***
```

Database Connections

IIR uses a modular architecture to overcome connection limits. The database module, named \texttt{libssadbm.so}, supports a fixed number of connections, typically 1024. However, it may be copied using the naming convention outlined below, to provide additional connections. \texttt{n} copies of the module will support \texttt{n} times the connections supported by a single copy. The copies must be named \texttt{libssadbmnnn.so} (or \texttt{.sl} on HP-UX PA-RISC machines) where \texttt{nnn=001, 002, ...}

To determine how many connections a single module provides, run \texttt{$SSABIN/dblist -c}. For example:

```bash
$SSABIN/dblist -c
Maximum connections per module: 1024
Linked databases: odb: sdb:
```

Based on the example above, if support for 3000 connections is required, two additional copies should be made:

```bash
cp $SSABIN/libssadbm.so $SSABIN/libssadbm001.so
cp $SSABIN/libssadbm.so $SSABIN/libssadbm002.so
```
Establishing an environment

All scripts require a valid environment. To establish one, please perform the following steps:

1. Start a fresh Bourne shell. For example,
   sh
2. Now source `iss` to establish the server’s environment variables:
   . `$SSATOP/env/iss`

   POSIX compliant shells may terminate the script when any undefined environment variables are referenced. If this occurs, add the command `set +u` as the second line of the script and source it again.

Test the ssapid utility

Some IIR processes extract process-id numbers using the supplied utility `$SSABIN/ssapid`. Verify that this utility works by running it as follows:

```bash
$SSABIN/ssapid $$
```

You should see output containing a process-id and some additional system dependant information. For example:

```
735024109 -bash
```

If nothing is shown then the utility did not work. One likely reason for this is that a non-standard `ps` utility was found in your search `PATH` before the standard UNIX `ps` utility. If the system does not have a `ps` utility that supports the `-p` option, then the `ssapid` script should be modified to suit your `ps` utility. Please contact Informatica Corporation Technical Support if this is the case.

Uninstall

On UNIX platforms, run the script `$SSATOP/uninstall` to remove the installation directory. Individual sub-products cannot be removed separately.

Note: All servers and utilities must be shut down prior to uninstalling the software.
Index

AIX, 5
Bind, 20
Configure Mode, 10
DB2, 5, see UDB
HP-UX, 5
Microsoft SQL Server, 5, 22
ODBC, 5, 23, 27, 28
    configuration, 29
Solaris, 5
SQL Code
   -1224, 21
   -444, 20
SSA_SRC
    user id, 12
SSA_TGT
    user id, 12
ssapid, 33

UDB, 5, 10, 17
User Defined Function, 19