# Database Migration Guide for Siebel Billing Manager and Siebel Service Manager

Version 5.1.1

Date Published 9/30/2006

ORACLE

# Contents

# 1 Migrating to a New Version of Billing Manager or Service Manager

## Overview

This chapter provides instructions for migrating to a new version of Billing Manager or Service Manager. This involves:

- Backing up your existing database and checking logins and user objects
- Migrating your OLTP and OLSP databases
- Checking for errors and resetting permissions after database migration
- Un-deploying J2EE web applications and redeploying the new versions
- Migrating to a custom data source as necessary

Always check the release notes for the version you are upgrading to for any last minute information that may update or supersede information in this document.

The rest of this chapter describes the migration steps for each database platform for Billing Manager and Service Manager.

## MSSQL

This section describes how to migrate a Self-Service Microsoft SQL server database from one version to another on the Windows operating system.

### Migrating 5.1.0.0 to 5.1.1.0

This section describes how to migrate and existing Self-Service database from version 5.1.0.0 to version 5.1.1.0.

#### *To migrate the Self-Service database*

1. Change your working directory to <TBM_HOME>\db\mssql\migration\510to511.
2. Log into SQL Query Analyzer as CBM db user and execute these scripts in sequence:

   add_indexes_511.sql

### Migrating 4.5.3 to 5.1.0.0

This section describes how to migrate an existing Self-Service database from version 4.5.3 to version 5.1.0.

***To migrate the Self-Service database***

1. Set the ANT_HOME, JAVA_HOME and PATH environment variables at the command prompt, for example:

   C:\set ANT_HOME=C:\Siebel\CBM\ant

   C:\set JAVA_HOME=D:\bea81sp4\jdk142_05

   C:\set PATH=%JAVA_HOME%\bin;%ANT_HOME%\bin;%PATH%

2. Change your working directory to<TBM_HOME>\db\mssql\migration\453to510.

3. Change the following property values in the cbm510.properties file in the folder for your environment:

   DB_NAME

   DB_USERNAME

   DB_PASSWORD

   DB_TMPUSERNAME

   DB_TMPPASSWORD

   DB_SYSADMIN

   DB_ADMINPASSWD

   INSTALL_DB_HOME

4. Execute the ant build script at the command prompt, for example:

   ```
   C:\ant -f dbinstall.xml

   Execute this command on the same machine as the DB server.
   ```

# Migrating 4.5.2.0 to 4.5.3.0

This section describes how to migrate and existing Self-Service database from version 4.5.2.0 to version 4.5.3.1.

***To migrate the Self-Service database***

1. Change your working directory to <TBM_HOME>\db\mssql\migration\452to453.

2. Log into SQL Query Analyzer as CBM db user and execute these scripts in sequence:

   drop_tsm_schema.sql
   create_tsm_schema.sql
   create_tsm_const.sql
   create_migrate_data.sql

# Oracle on UNIX

This section describes how to migrate a Self-Service Oracle database from one version to another on a UNIX system.

## Migrating to 5.1.1.0

### *To migrate the Self-Service database*

1 Check for init<tbm_sid>.ora file for the following parameters:

query_rewrite_integrity = trusted

query_rewrite _enabled = true

compatible = 8.1.0.0.0 (or higher)

If these parameters are not present, add them accordingly to the init<tbm_sid>.ora file. This file is normally located in $ORACLE_HOME/dbs folder.

If "compatible = <version number>" is not set, be sure to specify the current Oracle release version number. You can find the version number by executing the following command in SQL*Plus after logging in as sys user:

select * from v$version;

After editing this file, restart the database before proceeding to the next step.

2 Log into the database server as oracle user.

3 Change your working directory to <TBM_HOME>\db\oracle.

4 Add permission to <tbm user>.

Log into SQL*Plus and execute the following command to grant permissions:

ORACLE_SID=<ORACLE_SID>

export ORACLE_SID
sqlplus -s /nolog
connect sys/<sys_password> as sysdba

grant create any index, query rewrite to <tbmuser>;

5 Execute the setup_schema.sh script, for example:

```
./setup_schema.sh
```

The following screen appears:

```
   -CBM Enhancement Main Menu
---------------------------------
      [1] Sign in Menu
      [2] Apply CBM enhancement

      [3] Initial data population
      [4] CBM Migration
```

```
      [Q] Quit
---------------------------------
Enter Your Selection:
```

6 Select Option 1. The login screen displays (examples of entries are also shown):

```
                SIGN IN MENU
-----------------------------------------------

Please enter Oracle SID        --> edx0
(Enter appropriate Oracle SID)

Enter Database Username        --> edx_dba
(Enter user name for the Self-Service database)

Enter Database Password        --> edx
(Enter the Password for the Self-Service database)

Please enter the password for SYS user ...>change_on_install
(Enter the Password for the "sys" user)
```

7 Selection Option 4 for Migration. The following menu appears:

```
Database Version Migration

-------------------------

1. Version 5.1.0.0 to 5.1.1.0

2. Version 4.5.3.1-4.5.3.8 to 5.1.1.0

3. Version 4.5.2.0 to 4.5.3.1

R. Exit to Main menu

Enter Your Option:
```

8 Select Option 1 (Version 5.1.0.0 to 5.1.1.0) to migrate the database from 5.1.0 to 5.1.1. The following warning prompt appears:

```
**Warning**

We strongly advise a full backup of your existing

database before applying the migration.

Do you have a backup (Y/N):Y
```

9 Select Option 2 (Version 4.5.3.1-4.5.3.8 to 5.1.1.0) to migrate the database from 4.5.3 to 5.1.1. The following warning prompt appears:

```
**Warning**

We strongly advise a full backup of your existing

database before applying the migration.

Do you have a backup (Y/N):Y
```

10 Select Option 3 (4.5.2.0 to 4.5.3.1) to migrate the database from 4.5.2.0 to 4.5.3.1. The following warning prompt appears:

```
**Warning**
We strongly advise a full backup of your existing
database before applying the migration.
Do you have a backup (Y/N):Y
```

## Migrating from 5.1.0.0 to 5.1.1.0 (CSM Only)

No migration.

## Migrating from 4.5.3.0 to 5.1.0.0 (CSM Only)

If you are installing Communications Service Manager, you must also run the following scripts to fully migrate to 5.1:

1 Enter the following:

```
cd <CBM_HOME>/db/tsm/oracle/migration/453to453x
sqlplus {user}/{password}@{tns_name}
SQL>@cmscase.sql;
SQL>@create_sequences.sql;
SQL>exit
```

Where {user} is the database user,{password} is the database password, and {tns_name} is the Oracle tns name.

2 Enter the following:

```
cd <CBM_HOME>/db/tsm/oracle/migration/453to510
sqlplus {user}/{password}@{tns_name}
SQL>@csm_index.sql;
SQL>exit
```

Where {user} is the database user,{password} is the database password, and {tns_name} is the Oracle tns name.

# Oracle on Windows

This section describes how to migrate a Self-Service Oracle database to a newer version.

# Migrating 5.1.0.0 to 5.1.1.0

***To migrate the Self-Service database***

1. Check for init<tbm_sid>.ora file for the following parameters:

   query_rewrite_integrity = trusted

   query_rewrite _enabled = true

   compatible = 8.1.0.0.0 (or higher)

   If they are not present, add them accordingly to the init<tbm_sid>.ora file. This file is normally located in $ORACLE_HOME/dbs folder.

   If "compatible = <version number>" is not set, be sure to specify the current Oracle release version number. You can find the version number by executing the following command in SQL*Plus after logging in as sys user:

   select * from v$version;

   After editing this file, restart the database before proceeding to the next step.

2. Log into the database server as oracle user.
3. Change your working directory to <TBM_HOME>\db\oracle\migration\510to511.
4. Add permission to <tbm user>.

   Log into SQL*Plus and execute the following command to grant permissions:

   Set ORACLE_SID=<ORACLE_SID>
   sqlplus -s /nolog
   connect sys/<sys_password> as sysdba

   grant create any index, query rewrite to <tbmuser>;

5. Log into SQL*Plus as Billing Manager user and run the following script:

   ```
   sqlplus -s /nolog
   connect <tbm_user>/<tbm_user_password>
   spool migrate_510_511.log
   @add_indexes.sql

   spool off
   exit;
   ```

6. Check the migrate_510_511.log file for any errors.

# Migrating 4.5.3 to 5.1.1.0

***To migrate the Self-Service database***

1. Check for init<tbm_sid>.ora file for the following parameters:

   query_rewrite_integrity = trusted

query_rewrite _enabled = true

compatible = 8.1.0.0.0 (or higher)

If these parameters are not present, add them accordingly to the init<tbm_sid>.ora file. This file is normally located in $ORACLE_HOME/dbs folder.

If "compatible = <version number>" is not set, be sure to specify the current Oracle release version number. You can find the version number by executing the following command in SQL*Plus after logging in as sys user:

select * from v$version;

After editing this file, restart the database before proceeding to the next step.

2 Log into the database server as oracle user.

3 Add permission to <tbm user>.

Log into SQL*Plus and execute the following command to grant permissions:

```
Set ORACLE_SID=<ORACLE_SID>
sqlplus -s /nolog
connect sys/<sys_password> as sysdba
```

grant create any index, query rewrite to <tbmuser>;

4 Change your working directory to <TBM_HOME>\db\oracle\migration\453to510.

5 Log into SQL*Plus as Billing Manager user and run the following script:

```
Set ORACLE_SID=<ORACLE_SID>
sqlplus -s /nolog
connect <tbm_user>/<tbm_user_password>
spool migrate_453_510.log
@add_column.sql
```

6 Connect into database as the "SYS" db user, and run the following script:

```
connect sys/<sys_password> as sysdba
@temp_user_creation.sql
```

7 Connect to the database as the "CBM_SCRPT" user (the pass word is "cbm"), and run the following script:

```
connect CBM_SCRPT /cbm
@<TBM_HOME>\db\oracle\install_schema_migrate.sql

Where CBM_HOME is the absolute path to your CBM installation.
```

8 Connect to the database as the Billing Manager user and run the following script:

```
connect <tbm_user>/<tbm_user_password>
@rename_const_index.sql <tbm_user>
@move_tablespace.sql <tbm_user>

@move_indexes_to_tablespaces.sql <tbm_user>
@rebuild_index.sql <tbm_user>
```

9 Connect to the database as the "SYS" user, and run the following script:

```
connect sys/<sys_password> as sysdba
drop user CBM_SCRPT cascade;
spool off
       exit;
```

10 Check the migrate_453_510.log file for any errors.

11 Change your working directory to <TBM_HOME>\db\oracle\migration\510to511.

12 Log into SQL*Plus as Billing Manager user and run the following script:

```
Set ORACLE_SID=<ORACLE_SID>
sqlplus -s /nolog
connect <tbm_user>/<tbm_user_password>
spool migrate_510_511.log
@add_indexes.sql

spool off
exit;
```

13 Check the migrate_510_511.log file for any errors.

## Migrating 4.5.3.0 to 4.5.3.x

***To migrate the Self-Service database***

1 Change your working directory to <TBM_HOME>\db\oracle\migration\453to453x.

2 Log into SQL*Plus as Billing Manager user and run the migration script, for example:

```
Set ORACLE_SID=<ORACLE_SID>
sqlplus -s /nolog
connect <tbm_user>/<tbm_user_password>
spool migrate_453_453_x.log
@addindexes.sql;
spool off
exit;
```

3 Check the migrate_453_453_x.log file for any errors.

## Migrating 4.5.2.0 to 4.5.3.0

***To migrate the Self-Service database***

1 Change your working directory to <TBM_HOME>\db\oracle\migration\452to453.

2 Log into SQL*Plus as the Billing Manager user and run the migration script, for example:

```
Set ORACLE_SID=<ORACLE_SID>
sqlplus -s /nolog
connect < tbm_user>/<tbm_user_password>
spool migrate_452_453.log
```

```
@drop_tsm_schema.sql;
@create_tsm_schema.sql;
@create_tsm_const.sql;
@create_migrate_data.sql;
spool off
exit;
```

3 Check the migrate_452_453.log file for any errors.

## Migrating from 5.1.0.0 to 5.1.1.0 (CSM Only)

No migration.

## Migrating from 4.5.3 to 5.1 (CSM Only)

If you are installing Communications Service Manager, you must also run the following scripts to fully migrate to 5.1:

1 Enter the following:

cd <CBM_HOME>\db\tsm\oracle\migration**453to453x**
sqlplus {user}/{password}@{tns_name}
SQL>@**cmscase.sql**;
SQL>@**create_sequences.sql**;
SQL>exit

Where {user} is the database user,{password} is the database password, and {tns_name} is the Oracle tns name.

2 Enter the following:

cd <CBM_HOME>\db\tsm\oracle\migration**453to510**
sqlplus {user}/{password}@{tns_name}
SQL>@**csm_index.sql**;
SQL>exit

Where {user} is the database user,{password} is the database password, and {tns_name} is the Oracle tns name.

# Oracle 5.1 to 5.1.1 - Hierarchy Node Name Migration

In some versions of CBM/CSM prior to 5.1.1, business structure hierarchy names and business structure folder names were not properly trimmed, so some of these names may have been written to the database with leading and/or trailing spaces.

There could also be a name with leading or trailing spaces and the same name without those spaces; in this case, when the leading and trailing spaces are removed, there will be a duplicate name.

Leading and trailing spaces in these names cause several problems in CBM/CSM, including incorrect node counts and failures to expand, collapse, or work with the affected hierarchy or folder.

CBM/CSM version 5.1.1 trims these names correctly, so this will not be a problem going forward.

The scripts described in this section are intended to clean up any leading and trailing spaces in your existing database. We strongly urge you to run these scripts to make sure that your data is clean going forward with CBM/CSM 5.1.1.

The process of migrating hierarchy node names includes:

1. Running the Duplicate Reporting script to report any duplicates which will result from trimming leading and trailing spaces.
2. Fixing any reported duplicates either manually or by running another script which can eliminate duplicates automatically.
3. Running the Duplicate Reporting script again to make sure any duplicates are resolved.
4. Running the Trim Spaces script to trim leading and trailing spaces.

These scripts only need to be run once, on the migration of your database to 5.1.1; they do not need to be run going forward.

All scripts are located in your CBM_HOME/db/oracle/migration/510to511 folder (or TSM_HOME/db/oracle/migration/510to511 folder).

The following steps assume you are working in that directory, and that you have used the default names for the Oracle SID, username, and password; please adjust these to match what you have used.

***Before you begin:***

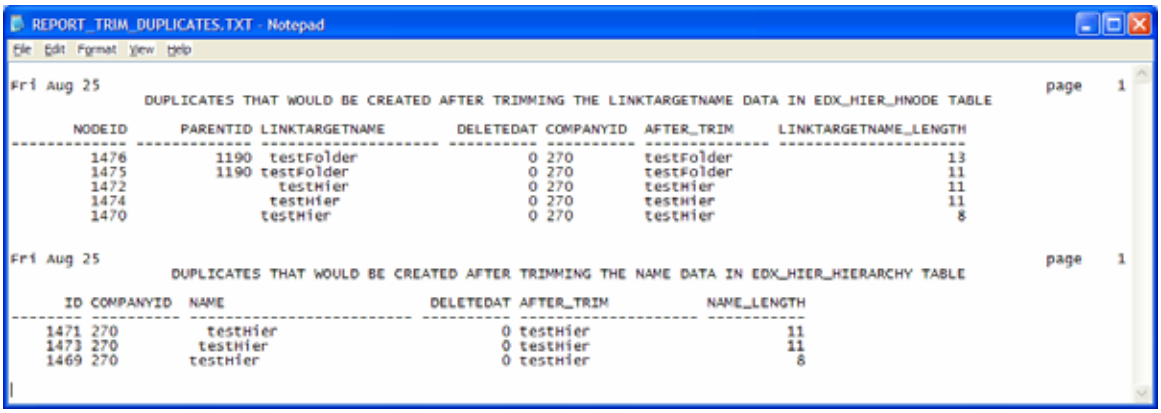
Before running these scripts, back up the EDX_HIER_HNODE and EDX_HIER_HIERARCHY tables. A backup will be invaluable in the unlikely event that a problem occurs.

***To migrate hierarchy node names to trim leading and trailing spaces***

1. Enter SQL*Plus as the CBM or CSM database user, and run the Duplicate Reporting script:

   ```
   sqlplus edx_dba/edx@edx.db @report_trim_duplicates.sql
   ```

   This script makes no changes to your database; it only reports any duplicates that would be created if leading and trailing spaces were removed. Look in the same directory as the script for the report file, which will be named REPORT_TRIM_DUPLICATES.TXT. Here is a sample of what that file might look like:




Look for duplicates reported in either the EDX_HIER_HNODE table or the EDX_HIER_HIERARCHY table. Note that each hierarchy name (NAME column of the EDX_HIER_HIERARCHY table) must be unique. In the EDX_HIER_HNODE table, each child of any given parent must have a unique link target name (LINKTARGETNAME column of the EDX_HIER_HNODE table). Duplicate link target names are allowed if they have different parents or are in different companies, however, a folder named Employees (for example) can exist as a child of a folder named Accounting, and another folder named Employees can also exist as a child of a folder named HR.

2 If the script does not report any duplicates, you are ready to trim leading and trailing spaces. Skip to Step 4.

If you have duplicates, you can choose one of two ways to fix them:

(a) Manually fix the duplicates yourself. This is the best solution, as you can control exactly what to do with the name and can make sure that the name provided is appropriate for the particular context of the particular business structure. For example, a name with leading and trailing spaces could have been used as the name for a hierarchy that has many business structures underneath it. There could be another hierarchy with the same name but with no leading or trailing spaces, which could have other structures under it. Each of these hierarchies needs a unique name so it can be used properly. Only you can determine what the particular names ought to be, or if one or the other of the duplicates should be deleted entirely.

To fix the duplicates yourself, you need to change the reported duplicate names in two tables: the NAME column in the EDX_HIER_HIERARCHY table and the LINKTARGETNAME column in the EDX_HIER_HNODE table. Use your preferred database tool (or SQL*Plus) to make these changes rather than the CBM/CSM UI because the nodes might not appear correctly (or at all) in the UI.

(b) Fix the duplicates automatically. To do this, run the following script:

```
sqlplus edx_dba/edx@edx.db @handle_hier_dup.sql
```

This script finds any duplicates and renames them by appending "- dup" + the node id. For example, if you have three duplicate names "Name" with node ids 1, 2, and 3, they will be named "Name- dup1", "Name- dup2", and "Name- dup3". Please note that these names are assigned randomly; there is NO implication that any particular one is in fact the original record or the one which should be saved. Only you can determine the correct course of action to handle each particular duplicate.

We recommend that you use the automatic fix script ONLY in situations where the only duplicates

are not in the production environment, and where the names displayed really don't matter. Even in that case, you will probably want to go back and handle the particular cases that the script finds. The script gives you a report of any names it has changed to assist you in finding these cases.

3 Repeat Step 1 until all duplicates have been resolved. Once all duplicates have been resolved (or if it did not find any duplicates), go to Step 4.

4 Only when all duplicates have been resolved (or if it did not find any duplicates) and the report in Step 1 returns no rows, run the Trim Spaces script:

sqlplus edx_dba/edx@edx.db @trim_spaces.sql

Be sure to run this script even if you did not find any duplicates in previous steps. Leading and trailing spaces are not removed from the names in the database until this script is run.

The Trim Spaces script:
(a) Trims leading and trailing spaces on the hierarchy name in the EDX_HIER_HIERARCHY table;
(b) Trims leading and trailing spaces on the link target name in the EDX_HIER_HNODE table; and
(c) Commits all changes.

If any unique constraint errors occur while running the Trim Spaces script, there were some unresolved duplicates in the DB. Go back to Step 1 and follow these steps to resolve the duplicates.

# 2 Migrating To A New Version of eaDirect

## Overview

This chapter describes how to migrate to a new version of eaDirect. This involves:

- Backing up your existing database and checking logins and user objects
- Backing up your data directories **`AppProfiles, Data, Input`** and **`Output`**
- Migrating your eaDirect database
- Checking for errors and resetting permissions after database migration
- Undeploying J2EE web applications and redeploying the new versions
- Migrating to a custom data source as necessary

**CAUTION:** The installation and configuration examples shown in this guide use default eaDirect pathnames, privileges, and permissions. If you choose not to accept the default values, make sure your values are consistent on all servers across your installation of eaDirect.

**NOTE:** When migrating Billing Manager (TBM/eBilling), eaDirect is now called Platform Services, and is located in `TBM_HOME/estatement` instead of `/opt/EDCSbd`.

## Migrating an Existing Oracle Database

To migrate an existing eaDirect database to a newer version, you must run the database setup shell script **`edx_admin.sh.`** The script's Main Menu has a Database Version Migration option that lists available migration paths.

### *To migrate an eaDirect database for Oracle on UNIX*

1. Switch user to the **`DB Admin`** user, in this example **`oracle`**. Oracle requires the administrative user in order to create files.

   ```
   su – oracle
   ```

2. Make sure that the Oracle listener is running for your eaDirect database.
3. Change directory to your eaDirect database home directory, for example:

   ```
   cd /opt/EDCSbd/db/oracle
   ```

4. Start the **`edx_admin`** shell script with the command:

   ```
   ./edx_admin.sh
   ```

   The Server Administration Main Menu appears.

```
edocs eaDirect Server Administration Main Menu
 ------------------------------------------------------
 [1] Sign in Menu
 [2] Capture Database File Locations
 [3] Install edocs e-Statement
 [4] Initial Data Population
 [5] Database Version Migration
 [Q] Quit
 ------------------------------------------------------
 Enter Your Selection: 5
```

5 Select Option 5, **Database Version Migration**.

6 Enter the username, password, and database name for the eaDirect database.

**CAUTION:** When creating an Oracle database, limit its name to eight characters. Defining or entering an Oracle SID with more than eight characters causes the Oracle database configuration to fail.

The following example shows sample values for **Oracle** on **Solaris**. Specify suitable values for your platform.

```
 Please enter Oracle SID        --> edx_dba
 Enter Database Username        --> edx
 Enter Database Password        --> edx0
```

Press **Enter** to return to the main menu.

7 From the Database Version Migration menu, select the version you are migrating from and migrating to, for example **40To42**.

**CAUTION:** Be sure to select the correct version for your existing installation.

8 Indicate whether you have already done a full backup of your database.

```
We strongly advise taking full backup of your existing database before applying
the migration

Do you have a backup (Y/N): Y
```

If you have not backed up your database, you are asked whether you want to continue the migration process.

```
Still continue with the migration (Y/N): Y
```

9 Enter your choice to abort the migration process or continue. If **no**, you are returned to the Database Version Migration menu where you can select the option to return to the previous menu.

If **yes**, migration begins. Select 'Q' to end the database migration session.

If you choose to continue with the database migration, the migration script **migrate.sh** runs for your version upgrade.

```
*********************************************
Migration process started
*********************************************
```

10 If prompted, enter the number of partitions for the Index database table:

```
Enter the partition count...([1|4|12] Default is 1)> 1
```

```
The following message appears:
```

```
This migration requires the creation of 2 new tablespaces!>
Please enter valid paths and ensure at least 200mb disk space...>
```

Database files can reside wherever you want them to. If you plan to use only one disk location, you can simply specify:

```
$ORACLE_HOME/oradata
```

Depending on your platform, you may then be prompted to enter multiple locations for your **database control** file. This file allows access to the database, so you should specify backup locations in case a database control file is deleted or corrupted. Consult your Oracle documentation for details.

You may also be prompted to enter suggested **mount points** for your database. Consult your Oracle documentation for details.

The configuration process checks the validity of the specified locations and displays a success message if no problems are encountered.

11 A successful database migration displays the following message:

```
Migration successful
```

12 The script prompts you to check your session log for errors. For example:

```
*********************************************
Please check the log file for any errors...
Log filename: /opt/EDCSbd/db/oracle/migrate0411100446.log
*********************************************
Ending migration process
```

13 From the Database Version Migration menu, select **Return to previous menu (R)**.

14 From the edocs Server Administration Main Menu, select **Quit (Q)**.

### *To migrate an eaDirect database for MSSQL on Windows*

Run the **edx_pkg_versioning.sql** and **pwc_runtime.sql** scripts located under <CBM_HOME>\estatement\db\mssql:

1 Launch a Query Analyzer and login as the database schema owner, for example edx_dba.

2 Open the above file within the Query Analyzer.

3 Click the green arrow to compile.

# Checking for Errors and Resetting Permissions

After migrating a database from a previous release of eaDirect, you must:

- Check **.log** files for errors
- Reset permissions for edocs directories and files

## Checking .log files for errors

After migrating an eaDirect database, you should check the following log files for errors:

- **scan.log**
- **migrate<version>.log**

In the context of migrating an eaDirect database, a **normal error** is one that does not cause the upgrade process to fail, such as trying to drop an object that does not exist. An **abnormal error** is one that can cause the database migration process to fail, which can have a cascading effect. A single abnormal error can lead to many other abnormal errors as the database migration process proceeds.

Although there is no easy way to differentiate between normal and abnormal errors, there is a way to check whether the database upgrade was successful. Typically, if the process does not flag invalid objects or there are no violations of referential constraints, then the operation was successful.

### Resetting Permissions

After migrating your database, confirm that all the eaDirect directories, with the exception of the **/db** directory and its subdirectories, reflect the same directory and file ownership as that of the application server owner.

For database files and directories under **$EDX_HOME**, eaDirect uses the default owner and group permissions.

# Updating Database Passwords for Your Application Server

## UNIX

If the database password for an existing database has changed, UNIX administrators can update your eaDirect environment with the application server script **edx_config.** Run this script **on your application server** using the instructions in Capturing UNIX Environment Data with edx_config. Enter your new password at the prompt.

Once the script has completed, check to see that the following files have been updated with your new password:

- **edx_env**
- **edx_sqlldr.config**
- **ws_config.xml** (WebSphere) or **config.xml** (WebLogic)
- JVM settings for system properties in your application server console

### Windows

If the database password for an existing database has changed, Windows administrators can update your eaDirect environment with the application server script **edx_load.config.bat.** Edit this script **on your application server** using the instructions in Editing Windows Environment Data with edx_env.bat and edx_load.config.bat to have the correct user, password and database alias. This file is then sourced at application server startup.

# Migrating J2EE Applications

When migrating to a new version of Siebel Self-Service software, you must first un-deploy J2EE applications running on your application server. Once you have migrated your database, you should then redeploy the new versions of each eaDirect web application, including eaDirect and samples. These sample applications provide examples of how to use the current features of eaDirect. You must:

- See the *Release Notes* for your version for important information about specific migration requirements and settings.
- See *Deploying and Customizing J2EE Applications* for information about the components that make up the J2EE and Web applications, and the procedures for recreating EAR and WAR files.
- Merge your custom code into the sample application, and rename it. For more information, see *Deploying and Customizing J2EE Applications*.
- Redeploy your custom application.

For more information on deployment, see the eaDirect *Deploying and Customizing J2EE Applications Guide.*

**CAUTION:** If you receive error messages when starting up your application server after migration, check to make sure you have undeployed all old versions of edocs J2EE web applications and redeployed the new versions. Manually editing the config.xml files for deployment is not recommended.

# Migrating To a Custom Data Source EJB

**eaDirect 3.4 and later versions** allow you to specify a datasource EJB for each eaDirect application (DDN) you create in the Command Center. When creating an eaDirect application in the Command Center, a datasource refers to an EJB in your application (EAR file) that specifies summary information and location of your document data.

**CAUTION:** Please consult the Release Notes for information on migrating from one particular version to another, as migration may require several steps.

During database migration, the `edx_admin.sh` script maps DDNs to the default datasource **`edx/ejb/EdocsDataSource`**, packaged in **`ear-eStatement.ear`**. For more information about datasource mapping, see the eaDirect *Production Guide*.

To change the datasource value to point to a custom datasource or to another datasource in a different deployed application, you can edit and run the SQL script **`update_data_source.sql`** on the database server. This script is located in the `/migration/<version>` subdirectory of the database directory in your eaDirect installation.

**TIP:** You do not need to run this script if you are using the default datasource.

You need to edit **`update_data_source.sql`** to replace the placeholder values for datasource name and document definition name (DDN) with your own values. Since each DDN may now have its own datasource, you can copy and paste the update sequence for each DDN. Save and close **`update_data_source.sql`** and run this SQL script to update your DDNs.

**CAUTION:** Consult the Release Notes for your version and platform to locate the correct version of update_data_source.sql.

# 3 Migration Notes

These migration notes were originally published in the eaDirect release notes beginning with Version 4.2.1.9.

## eaDirect

- If migrating from a version prior to eaDirect 4.0 and using Oracle, please follow the instructions in the document Index_Table_Migration.txt located in $EDX_HOME in addition to following the normal migration procedures.
- If you are migrating from eaDirect 4.1 and using Oracle, please run the following command after migrating to eaDirect 4.2 (if necessary).

  ```
  ./migrate_4x_patch.sh
  ```

  NOTE: This script prompts for ORACLE_SID, database username and database password.
- It is important to monitor the amount of database space used during the migration. The eaDirect/eaPay migration involves several database tables and their data. The size of the tables varies from small to huge. Depending on the volume of the tables, during the migration the database may use a huge amount of temporary space, undo space (applicable to Oracle), redo space (applicable to Oracle and DB2), and table/index data space. It is recommended that you monitor the space growth and pre-allocate enough space before migration and/or during the migration. It is advisable to allocate a big enough chunk of space just to make sure the migration doesn't fail due to insufficient space.
- For the Oracle database, if you see that any stored object (package, procedure, function) is "invalid" after the migration, it may not be due to the migration. To determine if this is a critical issue, recompile the "invalid" object manually. For example, if DDN_MGR package becomes invalid, log on to the database as the database owner and compile it as follows:

  ```
  SQL> alter package ddn_mgr compile;
  ```

  If the above compile command runs successfully, then this is not an issue and you have resolved it. Otherwise, it may be a critical issue and you should contact edocs Technical Support.

- eaDirect 4.0 through 4.2 require a separate indexing DDF to be used exclusively with an application's Indexer job. eaDirect indexes all fields in the DDF published for Indexer (and displays all fields in the hit list). The DDF should contain all the fields, group fields, and sub-documents that require indexing. The eaDirect 4.0 through 4.2 database migration migrates the last indexed table structure. The fields defined in the DDF used for the Indexer job must correspond to the migrated index table structure. After you have migrated the database, adding, removing, or renaming fields in the DDF requires that you manually modify your database tables. For a new application, ideally you should define every field you want indexed when you originally create the DDF. Once you run the Indexer job with this DDF, however, adding, removing, or renaming fields in the DDF requires that you manually modify your database tables. For information on how to manually modify the eaDirect database tables to add, remove, or rename existing fields please consult edocs Technical Services.
- As of 4.x, the database migration migrates existing Optional Fields specified in the Index tables. For new Indexer jobs created in 4.x, ten optional fields are created by default. You no longer have the option of specifying how many optional fields to include.
- If you have enabled Charting for Command Center Bill Statistics when using WebLogic prior to 4.0, you need to copy *javachart.jar* included in the *ear-easample.ear* and then edit *edx.config* to add the *javachart.jar* to the CLASSPATH. For specific details, see the WebLogic product notes in the release notes.
- If there is a failed IXLoader task in 3.4.x, then after migrating to 4.2 the instance should be cancelled and re-run, **not** retried from the existing failed instance.
- When you migrate to 4.2 from any prior release, you must delete and recreate any detail extractor jobs (and their schedules). Existing data from an old job can still be seen and is not affected. But the job itself, if you want to run it again, needs to be deleted and recreated to be run successfully. You must use the same detail extractor job name when recreating it, or it reprocesses index files that have already been processed and generate unique constraint errors.
- When you migrate to 4.2, do not remove any of the data files in application data directories. If you do, you will not be able to view previously indexed statements.
- *script.js* has moved. If you are unable to view statements, change the path of this file to *enrollment/scripts/script.js*. More information about *script.js* is provided below. (Note: this only applies for migration from 2.x to 4.2, not 3.x to 4.2.)
- Upon a migration from 2.1.x, both user statistics report and bill statistics report WILL NOT bring any information back. This is due to new tables introduced since 3.0. The reports no longer use the *rpt_activity* table; after migration, the reports include data from 3.x and forward only.
- Migration to eaDirect 4.2 on AIX is from versions 2.2.4.1 and higher. For migration of versions prior to 2.2.4.1 on the AIX platform, please contact Technical Support.
- If eaDirect processed an Email Job on the same day you migrate from a version prior to 3.4, wait at least 1 day before running the Email Job again to set the IVNScanner's "Scan Number of Days" parameter low enough so it doesn't pick up the older, already processed IVNs.
- When migrating a MSSQL database to 4.2 from a version older than 4.0 or 4.1, you must invoke the DBConfigTool more than once. You must run the tool for each step in the process. Database migration is not complete until the final step, "40 or 41 to 4.2" migration, has finished successfully.

- As mentioned in the *Installation and Configuration Guide*, if you are not using the default datasource you need to run the *update_data_source.sql* script for your database. This only applies for those migrations from a version prior to 3.4. The locations for each file are:
  - Oracle: <eaDirect-home-directory>/db/oracle/migration/32to34/
  - SQL Server: <eaDirect-home-directory>\db\mssql\migration\32to343\
  - DB2: <eaDirect-home-directory>/db/db2/migration/32to343/
- When migrating from versions prior to 3.2, if the user had used restricted characters as part of their password, they will not be able to login in 4.2. They include:

  ```
  !@#$%^&*()_+[]\|}{;'":,./?><
  ```

- When migrating your Web applications to eaDirect 4.2, you must upgrade ejb-application.jar to the latest version included in the eaSample sample application provided with eaDirect.
- When migrating to 4.5.2.5 and above, and using the Oracle database, please do the following two tasks:run $TBM_HOME/estatement/db/oracle/migrate_4x_patch.sh.
  - run the following from the directory $TBM_HOME/db/oracle/migration/452to4521:

    ```
    sqlplus {user}/password}@{tns_name}
    SQL>@upgrade_db.sql
    SQL>exit
    ```

    where {user} is the database user, password is the database password, and {tns_name} is the Oracle database tns name.
- If migrating using an Oracle database, please run:
  - For 4.5.2 or 4.5.2.4 to 4.5.3:

    ```
    sqlplus {user}/{password}@{tns_name}
    SQL>@AddTSMSchema.sql
    SQL>@MigrateData.sql
    SQL>exit
    ```

    where:

    {user} is the database user.
    {password} is the database password.
    {tns_name} is the Oracle tns name.

    Note that above sql scripts are found in $TBM_HOME/db/oracle/migration/452to453
  - For 4.5.2.x or 4.5.3 to 4.5.3.1:

    Opt/Run estatement/db/oracle/migrate_4x_patch.sh

## eaPay

- AVS service is an extension to the existing credit card validation process. If the biller has enabled the AVS, they need to specify the AVS enabled Verisign account at the time of Verisign gateway setup. For the migration, AVS is set to true (enabled) by default, so the biller needs to update the created Verisign credit card gateways to specify the AVS enabled Verisign account information. (If the biller does not want it, they can disable the AVS and update the Verisign gateway configuration.) As part of the migration process, the biller needs to send out emails to their customers about this new service and informing them to update their credit card accounts with address information exactly as it appears on the bill, otherwise the transactions will fail.
- The following APIs from the IPaymentAccountAccessor interface and PaymentAccountAccessorImpl and SSOPaymentAccountAccessor classes have been removed and are no longer supported:

```
PaymentAccountQueryResult getPaymentAccount(String uid, String pid,
ExtendedAccountQueryParams extAccQueryParams)
```

```
PaymentAccountQueryResult getPaymentAccounts(String uid,
ExtendedAccountQueryParams extAccQueryParams)
PaymentAccountQueryResult getPaymentAccounts(IAccount iaccount,
ExtendedAccountQueryParams extAccQueryParams)
PaymentAccountQueryResult getPaymentAccounts(String biller, PaymentType type,
PaymentAccountState state, int maxCount, ExtendedAccountQueryParams
extAccQueryParams)
List getPaymentAccountsToBeActivated(String ddn, PaymentType type, int maxCount)
PaymentAccountQueryResult getPaymentAccounts(String biller, PaymentType type,
String notifyStatus, int maxCount, ExtendedAccountQueryParams extAccQueryParams)
PaymentAccountQueryResult getExpiredPaymentAccounts(String ddn, PaymentType
pmtType, List status, java.util.Date expireDate, ExtendedAccountQueryParams
extAccQueryParams)
```

## SDK

The following are some general migration issues with the APIs for the edocs SDK:

- The *Logout.java* class must implement the method getUserIdAttribute of the com.edocs.app.LoginApp class. The method should retrieve the user's unique id. It is implemented by the *Login.java* class also.
- Email Notification Job users: The Account Resolver code which is used to return email addresses needs to be rewritten using the *SampleAccountResolver.java* provided in this release. The method *search* has replaced the *getEmailAddressesByAccountNumber*. Also note that every email must resolve at least twice or the system may throw an ArrayOutOfBoundsException.

The following are the deprecated APIs from previous releases:

### Deprecated Classes

```
com.edocs.jndi.cda.CDADataSourceFactory
com.edocs.jndi.cda.CDADB2Factory
com.edocs.jndi.cda.CDAJDBCFactory
com.edocs.jndi.cda.CDAOracleFactory
```

```
com.edocs.enrollment.HttpRequestParameters
com.edocs.enrollment.user.IAdmissionFactory
com.edocs.enrollment.user.IAuthenticationFactory
```

## Deprecated Interfaces

```
com.edocs.enrollment.user.IAccountEnumeration
com.edocs.enrollment.user.IAdmission
com.edocs.enrollment.user.IAdmissionHome
com.edocs.enrollment.user.IAuthentication
com.edocs.enrollment.user.IAuthenticationHome
```

## Deprecated Fields

```
com.edocs.app.App.METHOD
```

## Deprecated Methods

```
com.edocs.enrollment.user.IAuthentication.authenticate(NameValue[])
com.edocs.enrollment.user.IAccount.close()
com.edocs.enrollment.user.IAdmission.createAccount(NameValue[])
com.edocs.enrollment.user.IAdmission.deleteAccount(IAccount)
com.edocs.enrollment.user.IAdmission.fetchAccount(NameValue[])
com.edocs.app.verify.Verify.getAccountListSize(String)
com.edocs.enrollment.user.IAccount.getBillers()
com.edocs.enrollment.user.AccountBean.getBillers()
com.edocs.app.user.User.getDocument(String, String[], String, String, String,
String)
com.edocs.app.user.User.getDocumentDistributedInputStream(String, String[],
String, String, String, String)
com.edocs.enrollment.user.IAccountResolver.getEmailAddressesByAccountNumber(String
, String)
com.edocs.enrollment.user.IAccount.getExactValue(String, String)
com.edocs.enrollment.user.AccountBean.getExactValue(String, String)
com.edocs.enrollment.user.IAccount.getValue(String, String)
com.edocs.enrollment.user.AccountBean.getValue(String, String)
com.edocs.enrollment.user.IAccountResolver.isAccountValid(String, String)
com.edocs.enrollment.user.IAccount.isValid(NameValue[])
com.edocs.enrollment.user.AccountBean.isValid(NameValue[])
com.edocs.enrollment.user.IAuthentication.reAuthenticate(Handle, NameValue[])
com.edocs.enrollment.user.IAccountResolver.search(Attributes, String[])
com.edocs.enrollment.user.IAdmission.selectAccounts(NameValue[])
com.edocs.app.user.User.sendSortedDocument(String, String[], String, String,
String, String, SortParams, boolean, boolean, HttpServletRequest, Writer)
com.edocs.common.web.validation.ValidatorBean.setIllegalParameterValues(String[])
com.edocs.common.web.validation.ValidatorBean.setLegalParameterNames(String[])
com.edocs.enrollment.user.IAdmission.updateAccount(IAccount, NameValue[])
com.edocs.app.user.User.updateSummaryInfo(String, Name, String, String, String)
com.edocs.common.web.validation.ValidatorBean.validateParameterNames(ServletReques
t)
```

```
com.edocs.common.web.validation.ValidatorBean.validateParameterValues(ServletReque
st)
```

# 4 Account Data Migration

## Migrating data from Index (estatement) tables

This document describes a set of migration scripts that can populate specific Billing Manager tables (defined by the Professional Services, or PS, team) from statement account numbers stored in index (estatement) tables.

This process requires extracting account numbers from index table and then populating necessary Billing Manager tables according to the format described below.

### 1 (Step 1) Extracting account numbers from index tables

Extract account numbers (Z_PRIMARY where Z_CONTEXT == null) from index tables. (Professional services should clarify whether we need to scan all index tables or only the latest index table to get account numbers.)

Note: Account numbers can have duplicate entries, so remove duplicate while extracting.

Figure 1 shows sample data and its format in index tables (using simulated data). Siebel Professional Services provides actual index table format and sample data as required, but for this example the Z_PRIMARY and Z_CONTEXT columns are sufficient.

| Z_PRIMARY | Z_DOC_DATE | Z_DOC_ID | Z_IVN | Z_CONTEXT | BillDate | PhoneSummary | MobileCount | MobileNum | GroupSu |
|---|---|---|---|---|---|---|---|---|---|
| 80011008 | 6/17/2005 | ivn-1/po-0/bc-6253 | 1 | <NULL> | 05/04/04 | 600 | 2 | 407-113-5451 | 900 |
| 382-473-4937 | 6/17/2005 | ivn-1/po-1003165/l | 1 | 06122165 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 382-215-2803 | 6/17/2005 | ivn-1/po-1025259/l | 1 | 06122165 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 13875658 | 6/17/2005 | ivn-1/po-1036405/l | 1 | <NULL> | 05/04/04 | <NULL> | 1 | 784-749-3874 | <NULL> |
| 784-749-3874 | 6/17/2005 | ivn-1/po-1038357/l | 1 | 13875658 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 14464399 | 6/17/2005 | ivn-1/po-1066157/l | 1 | <NULL> | 05/04/04 | <NULL> | 1 | 748-373-7646 | <NULL> |
| 748-373-7646 | 6/17/2005 | ivn-1/po-1067647/l | 1 | 14464399 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 03286008 | 6/17/2005 | ivn-1/po-1124948/l | 1 | <NULL> | 05/04/04 | 600 | 20 | 847-349-4100 | 900 |
| 847-349-4100 | 6/17/2005 | ivn-1/po-1134924/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4101 | 6/17/2005 | ivn-1/po-1156027/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4102 | 6/17/2005 | ivn-1/po-1166414/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4103 | 6/17/2005 | ivn-1/po-1175641/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 31569801 | 6/17/2005 | ivn-1/po-121372/b | 1 | <NULL> | 05/04/04 | 600 | 2 | 494-392-9463 | 900 |
| 847-349-4104 | 6/17/2005 | ivn-1/po-1227056/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4105 | 6/17/2005 | ivn-1/po-1236289/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4117 | 6/17/2005 | ivn-1/po-1248870/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4106 | 6/17/2005 | ivn-1/po-1251577/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 494-392-9463 | 6/17/2005 | ivn-1/po-125282/b | 1 | 31569801 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4107 | 6/17/2005 | ivn-1/po-1269076/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4118 | 6/17/2005 | ivn-1/po-1334215/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4108 | 6/17/2005 | ivn-1/po-1336922/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4109 | 6/17/2005 | ivn-1/po-1355443/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |
| 847-349-4110 | 6/17/2005 | ivn-1/po-1368166/l | 1 | 03286008 | <NULL> | <NULL> | <NULL> | <NULL> | <NULL> |

Fig. 1 - Sample data in index table.

### 2 (Step 2) Insert account numbers (extracted from index tables) into Billing Manager tables.

You must update the following tables for each account number qualified from Step1.

***EDX_BSL_AMF_BACCOUNT***

***EDX_HIER_HNODE***

Before updating the above two tables with account number information, you must load the static data (pre-defined) for the dependant tables of above two. Listed below are the dependent tables that need to be populated along with their values.

**Step 2.1 – Loading static data**

**EDX_BSL_CMF_COMPANYPROFILE**

This table will have only one row.

```
ID = [sequence]
NAME = __default__
```

**EDX_BSL_CMF_COMPANY**

This table will have only one row.

```
ID = [sequence]
PROFILEID =  [EDX_BSL_CMF_COMPANYPROFILE.ID]
```

**EDX_HIER_HTYPE**

This table will have only two rows as described below.

Row1

```
ID = [sequence]
CODE = “BILLING”
NAME = “Billing Hierarchy”
DESCRIPTION = “Billing hierarchy”
```

Row2

```
ID = [sequence]
CODE = “BUSINESS”
NAME = “Business Hierarchy”
DESCRIPTION = “Business hierarchy”
```

**EDX_HIER_HIERARCHY**

This table will have only one row.

```
ID = [sequence]
HTYPE = [EDX_HIER_HTYPE.ID where CODE=”BILLING”]
COMPANYID = [EDX_BSL_CMF_COMPANY.ID]
NAME= “Billing Accounts”
DISPLAYNAME = “Billing Accounts”
CREATEDAT = timestamp (I think this is java current time in millisec  from
1/1/1970, 00:00:00. so we need to calculate this dynamically)
CREATEDBY = “siebeladmin”
DELETEDAT = 0
PUBLISHEDAT = 0
```

**EDX_HIER_HNODE**

```
NODEID = [sequence]
PARENTID = null
HIERARCHYID =  [EDX_HIER_HIERARCHY.ID]
LINKTARGETURI = “edx:hierarchy:folder:31b8b6086207ef7f:1c2fdde:1048bb8e8ac:-
```

```
8000:root”
LINKTARGETID = “root”
ENTITYISCONTAINER = 1
PATH = 1 [NODEID]
LINKTARGETNAME = “Billing Accounts”
CREATEDAT = timestamp (same as in EDX_HIER_HIERARCHY)
CREATEDBY = “siebeladmin”
MODIFIEDAT = timestamp
MODIFIEDBY = “siebeladmin”
DELETEDAT = 0
LINKTARGETTYPE = “com.edocs.common.hierarchy.api.IHierarchyFolder”
```

**Step 2.2 - Updating Account # related tables;**

**EDX_BSL_AMF_BACCOUNT**

Each account number extracted from index tables should be inserted into this table according to the below format.

```
ID = [sequence]
NAME = “<<account #>>” [account # is the value extracted from index table
(Z_PRIMARY)]
TYPE = “ACCOUNT”
CREATEDAT = timestamp (same as in EDX_HIER_HIERARCHY)
CREATEDBY = “siebeladmin”
```

For example, after adding two accounts, EDX_BSL_AMF_BACCOUNT looks like;

| | nodeID | name | description | type | createdAt | createdBy | modifiedAt | modifiedBy |
|---|---|---|---|---|---|---|---|---|
| | 1 | 80011008 | | ACCOUNT | 1119035648547 | siebeladmin | <NULL> | <NULL> |
| ✎ | 2 | 29006120 | | ACCOUNT | 1122252381160 | siebeladmin | <NULL> | <NULL> |
| * | | | | | | | | |

Fig. 2 – sample rows in edx_bsl_amf_baccount table.

***EDX_HIER_HNODE***

For each row inserted to EDX_BSL_AMF_BACCOUNT table, there should be corresponding entry in this table. The format is as follows:

```
NODEID = [sequence]
PARENTID = [NODEDID in the very first row - root node]
HIERARCHYID =  [EDX_HIER_HIERARCHY.ID]
LINKTARGETURI = “edx:amf:billingaccount:<<account #>>”
LINKTARGETID = “<<account #>>”
ENTITYISCONTAINER = 1
PATH =  [root node’s NODEID.current row NODEID]
LINKTARGETNAME = “<<account #>>”
CREATEDAT = timestamp (same as in EDX_HIER_HIERARCHY)
CREATEDBY = “siebeladmin”
MODIFIEDAT = timestamp
MODIFIEDBY = “siebeladmin”
DELETEDAT = 0
LINKTARGETTYPE = “ACCOUNT”
```

For example, after adding two accounts, EDX_HIER_HNODE looks like:

| | nodeID | parentID | hierarchyID | linkTargetURI | linkTargetID | entit | path | linkTargetName | alias | description |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | <NULL> | 1 | edx:hierarchy:folder:31b8b6086207 | root | 1 | 1 | Billing Accounts | <NULL> | <NULL> |
| | 2 | 1 | 1 | edx:amf:billingaccount:80011008 | 80011008 | 1 | 1.2 | 80011008 | <NULL> | <NULL> |
| | 3 | 1 | 1 | edx:amf:billingaccount:29006120 | 29006120 | 1 | 1.3 | 29006120 | <NULL> | <NULL> |
| * | | | | | | | | | | |

continued...

| entit | path | linkTargetName | alias | description | createdAt | createdBy | modifiedAt | modifiedBy | deletedAt | linkTargetType |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Billing Accounts | <NULL> | <NULL> | 1119035648172 | siebeladmin | 1119035648172 | siebeladmin | 0 | com.edocs.common.hierarchy.api.IHierarchyFolder |
| 1 | 1.2 | 80011008 | <NULL> | <NULL> | 1119035648813 | siebeladmin | 1119035648813 | siebeladmin | 0 | ACCOUNT |
| 1 | 1.3 | 29006120 | <NULL> | <NULL> | 1122252381300 | siebeladmin | 1122252381332 | siebeladmin | 0 | ACCOUNT |

Fig. 3 – Sample rows in edx_hier_hnode table.

***NOTE:***

- All unique keys *[sequence]* should be generated using "***hibernate_sequence***" sequence.
- Need to write errors to a log file. Shows the execution summary at the end of the log file:

  How many records inserted successfully.
  How many records failed to insert.

***ASSUMPTION:***

1. The Professional Services team agreed to populate all other Billing Manager tables by a PS effort, for example, migrating enrolled users, mapping enrolled user ids to correct nodeid in HNODE table.