Deltek Maconomy®

BPM Admin Guide

November 25, 2022

Contents

Overview

How this Document is Organized

This document is organized so that you can go quickly to the section you need. Use Ctrl+F to easily find the term you are searching for, or see the table below to find the section to help you.

Section	Description	Use it when…
Install	Steps to enable BPM in Maconomy and People Planner, as well as install SAP Business Object, and set up Pentaho.	▪ You must perform an installation and enable BPM in Maconomy and People Planner.
Upgrade	Steps to upgrade BPM, including ETL, universes and documents, and customizations.	▪ You must upgrade your BPM system.
Setup and Configuration	Provides setup and configurations for your BPM system.	▪ You need to set up particular aspects of your system.
Customizations	Details all the ways you can extend your BPM system through different customizations and localization.	▪ You want to maximize your Maconomy system through the various customizations and extensions available. ▪ You need localization steps.
System Maintenance and Regular Use	Includes details for performing overall maintenance on your BPM system.	▪ You need to perform routine system admin duties, such as running regular maintenance.
Other Setup	Miscellaneous setup material.	▪ Review non-typical procedures, such as USync installation, and using a non-standard web server port.
Troubleshooting	Includes troubleshooting as well as frequently asked questions for key areas of BPM.	▪ You have issues or questions when running BPM or related systems or processes.

INSTALL

Business Performance Management (BPM) Reporting consists of a collection of universes and standard reports. Universes provide actuals, budget amounts, and dimensions for producing standard reports and ad hoc queries in Business Objects Web Intelligence. All standard reports run on Maconomy's transactional database through one or more universes.

This section provides steps for installing BPM Reporting and setting up Maconomy. This process includes:

- Installing and configuring SAP BusinessObjects.
- Importing universes and reports that are contained in Lifecycle Management Business Intelligence Archive Resource (LCMBIAR) files.
- Managing users and single sign-on settings.
- Entering the necessary setup data in Maconomy.

    This data is required for the universes and reports to work correctly. Detailed explanations of the setup procedures are included in appendices to this document.

For specific requirements for hardware, see the documentation for BusinessObjects, Pentaho, Maconomy, Oracle, and SQL Server.

> **Tip:** This document does **not** contain information about hardware, sizing, performance tuning, or maintenance.

> **Note: Access Control**
>
> In BPM, access control applies in the same way as in Maconomy. In BPM Analysis, access control is not applied on dimension tables. The reason is that the analysis universes are used in another why where fact tables are typically involved in queries. When doing so, access control applies in the same way as in Maconomy.

Installation Files

SAP BusinessObjects

SAP BusinessObjects is required for *BPM Reporting*, *BPM Analysis*, and *BPM People Planner*.

Installation files for SAP BusinessObjects (including SAP LiveOffice), are located on the download server:

\\dl\applications\Released\BPM\Business Objects\BO BI 4.2\4.2 SP 8

Pentaho Data Integrator

Pentaho Data Integrator is required for *BPM Analysis* and *BPM People Planner*.

Installation files for Pentaho Data Integrator are located on the download server:

\\dl\applications\Released\BPM\Pentaho\Pentaho Data Integration\Pentaho-9.3-InstallationBundle\

Reporting Packing Unit (RPU)

The installation package containing BPM is comprised in a zipped folder called a *Reporting Packing Unit* (RPU*)*.

RPUs are available in zipped format on the DSM under the matching Maconomy version as a sub-product.

An RPU is named after the version and solution it contains. For example,

```
rpu.21.sp101.cu001.1.s-std.zip
```

The general format of the folder name is:

```
rpu.<main version>.<service pack>.<cumulative update>.<build number>.<solution>.zip
```

The solutions available are:

- Standard
- PSO
- CPA

A packing unit has the following content:

lcmbiars	This folder contains a LCMBIAR file for Oracle and one for MS SQL Server:
	Example format:
	```bpm-<product>.<main version>.<sp>.<cu>.<build number>.<solution>.<database>.lcmbiar```
	Furthermore, it contains a file `bpmtoc.txt,` which is a table-of-content file listing the CMS folders, documents, and universes the bpm LCMBIAR files contain.
css	This folder contains the file `bpmdefault.css` which provides WebI document the default BPM style. The file is only necessary to install if changes need to be made to the style of documents (see Customizing Document Styles).

dashboards	This folder contains the `JSP` files and additional files to be installed on the web server in order for BPM Reporting Dashboards to function (see Customizing BPM Dashboards).
dbcheckscripts	This folder contains SQL scripts that can be executed on the Maconomy database in order to check that Maconomy setup supports BPM. The scripts are divided into two sub-folders - one for Oracle and one for MS SQL Server.
documentation	This folder contains detailed documentation of the BPM universes. The documentation consists of an HTML file for each universe. This HTML file can be viewed in a browser and lists the folders, dimension objects, measure objects, attribute objects, and filters that the universe offers. For each object, a description is provided.
etl	This folder contains XML files for the ETL of BPM Analysis and BPM People Planner, respectively.
	Example format:
	`bpm-analysis-etl.<main version>.<service pack>.<cumulative update>.<build number>.xml`
performanceviews	This folder contains drop and create scripts for the performance views of BPM Reporting. Furthermore, it contains scripts for installing the flush mechanism of the shared pool. Both performance views and the flush mechanism are installed using MConfig. It also contains drop scripts of the old MV-named performance views. The files in the folder are merely for documentation or relevant if manual installation or deinstallation is needed.

Licenses

SAP BusinessObjects installation and Pentaho Installation require license keys and license files, respectively.

All licenses can be requested by contacting **Deltek Distribution** at HQ-Distribution@deltek.com.

Certified Versions

All certified versions of SAP BusinessObjects, Pentaho, and other BPM related tools and platforms are stated in the Deltek Maconomy Supported Products document. Specific updates are announced in TechNews.

New Installation

Workflow

This process of new installation is divided into four main steps, described below.

1	• Enable BPM in Maconomy
2	• Enable BPM in People Planner
3	• Install SAP BusinessObjects
4	• Set up Pentaho Data Integrator
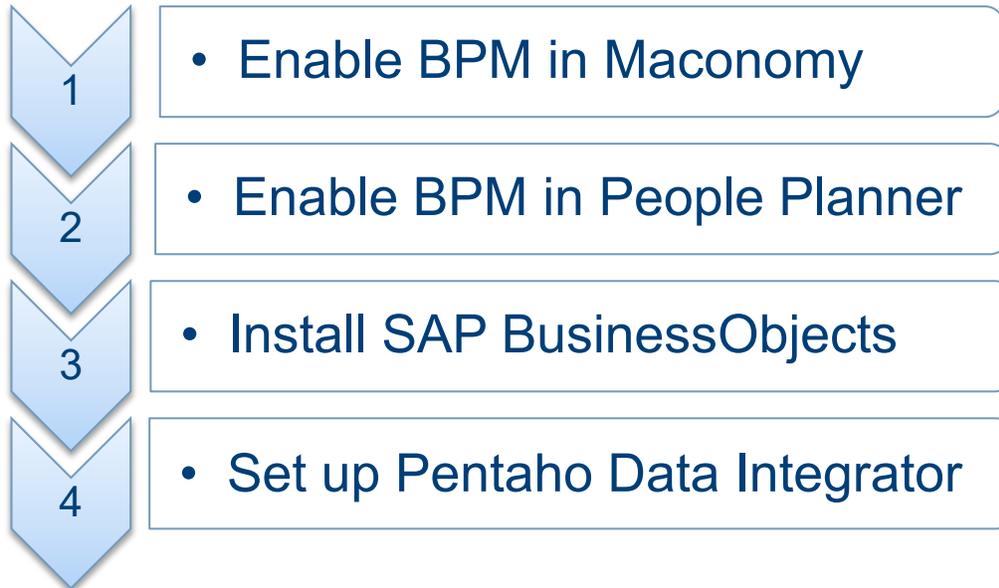
Enable BPM in Maconomy

Enabling BPM in MConfig is required for *BPM Reporting* and *BPM Analysis*.

To prepare the Maconomy database for BPM, enabling is needed. The enabling will install the necessary database views, triggers, and indexes that BPM uses, on the Maconomy database.

1. Start MConfig and then select the relevant Maconomy application.
2. On the Application Configuration dialog box, select the relevant shortname, and click **Edit**.
3. On the Shortname Configuration dialog box, enable:
   - **Reporting** to prepare for installing BPM Reporting.
   - **Business Intelligence** to prepare for installing BPM Analysis.

     Enabling **Reporting** will install EX-views and access control views for BPM Reporting. It will also install the necessary performance views and for Oracle databases a flushing mechanism that automatically cleans up in the shared pool.

     Enabling **Business Intelligence** will install EX-views, apply transaction time stamps, and trigger managing the marking of deleted records in Maconomy for certain tables.
4. Set a time for when the flushing mechanism is to run.

> **Note:** If **BPM Reporting** or **BPM Analysis** is already enabled, but there are problems with the views, disable it, complete the installation change, and then complete the above steps again to reinstall the reporting views, transaction time stamps, and delete triggers.

Enable BPM in People Planner

Enabling BPM in People planner is required for *BPM People Planner*.

You have to create a task in PeoplePlanner preparing data for ETL. Follow these steps:

1. Run CreateExternalViewsMSSQL.sql. This will create the required Views.

2. Select **New Scheduled Task** from the ScheduledTasks node in the Views:
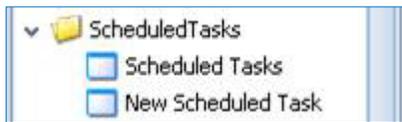
3. Create a scheduled task with type Update Reporting Tables:
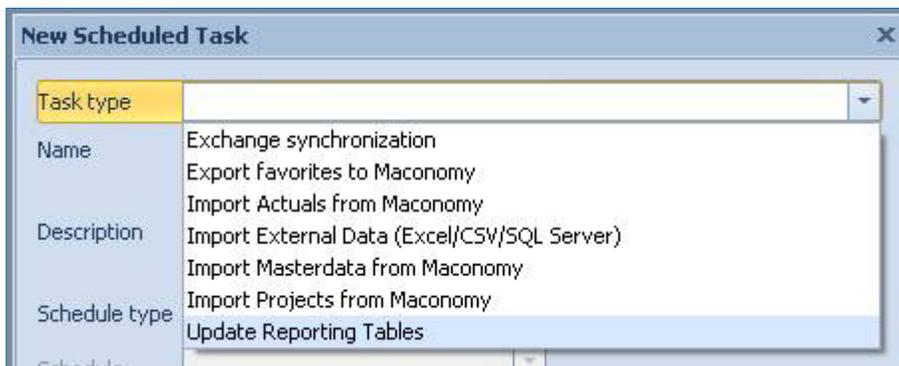
4. Set schedule for the Task.

5. Install the PeoplePlannerService (if it is not installed already.)

To refresh the data immediately, you can open the Scheduled Tasks window. There you can see your tasks and you can select the task and press the **Execute** button.

Install SAP BusinessObjects

SAP BusinessObjects is required for *BPM Reporting*, *BPM Analysis*, and *BPM People Planner*.

Complete the steps of installing SAP BusinessObjects on the reporting server.

> **Note:** We are now using "One-Installer" starting from BO 4.2 SP8 Patch 1. You can use this installer for Upgrading or Full Installation. For example, you can install BO 4.2 SP8 Patch 1 directly without the need for BO 4.2 SP8 base.

To install SAP BusinessObjects:

1. It is strongly recommended that all Windows programs be closed before running the installation. Then, run the installer.

2. Check prerequisites. Failed critical dependencies must be addressed before the installation can continue, while failed optional dependencies might result in some components not installing.

3. Enter the SAP BusinessObjects BI platform 4.2 product key.

4. Select **Full** for the type of installation.

5. Select **Configure and install a Sybase SQL Anywhere database**. This will serve as the storage of data for the SAP BusinessObjects BI platform.

6. For the Java Web Application Server solution, select **Install the default Tomcat Java Web Application Server and automatically deploy web applications**.

7. For the version management, select **Do not configure a version control system at this time**.

8. The succeeding steps will be asking for port numbers but the standard for those are already preset in the setup.

9. Set a name and port number for the SIA node. The node name is usually the name of the machine.

10. Set a port number for the CMS.

11. Set the administrator account password and cluster key for the CMS. The cluster key is used to connect multiple CMS servers together.

12. If Sybase SQL Anywhere was set to be installed, set the server port and administrator password for the Sybase SQL Anywhere database.

13. Set port information for Tomcat.

14. Set the HTTP listening port for connecting to WACS and RESTful Web Services.

15. For the server profiling and diagnostics, select **Do not configure connectivity to SMD Agent**.

16. Select **Do not configure connectivity to Introscope Enterprise Manager**.

17. After the setup, the program is ready to begin the installation of SAP BusinessObjects BI platform 4.2 SP8 Patch 1.
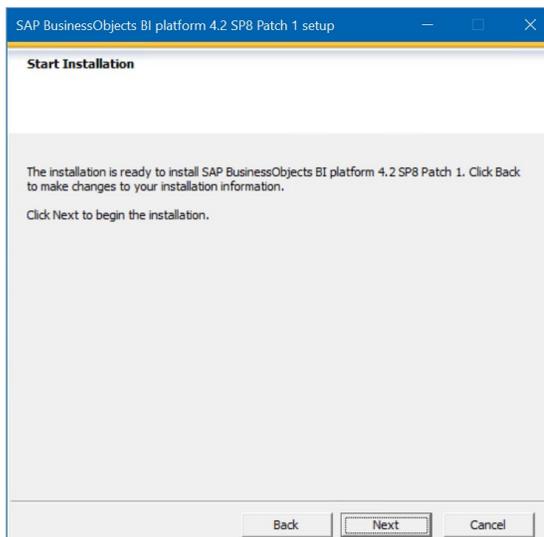
> **Tip:** See the material at the link below for SAP Business Object tutorials.
>
> http://scn.sap.com/docs/DOC-7725

> **Tip:** If you have installed SAP BusinessObjects Business Intelligence 4.1 Support Pack 7 or BI 4.2 onward, it is recommended to disable the Enable Memory Analysis feature to improve performance.
>
> If you are using an earlier version on the Windows Platform, enable this feature to prevent a possible crash of the Web Intelligence Processing Server.
>
> **Note:** Ensure that you have enough RAM and monitor the memory usage of your server. If your server runs out of RAM, it will start to use swap, which may cause performance issues.
>
> If you have limited RAM and you find your server is using swap, keep this feature enabled to ensure maximum performance.

When you upgrade to SAP BusinessObjects 4.2 SP8 Patch 1, you must enable JavaScript content in Web Intelligence to ensure links to the workspace client work.

Enable JavaScript Content

To enable JavaScript content in Web Intelligence, follow these steps:

1. Log in to the Central Management Console (CMC).
2. Go to **Applications » Web Intelligence**.
3. Select the **Enable javascript content into Web Intelligence documents** check box.
4. Click **Save**.

Set up BPM Data Warehouse

Pentaho Data Integrator is required for *BPM Analysis* and *BPM People Planner*. Where names for these two products differ, it is stated.

To set up Pentaho Data Integrator, you must:

- Install Pentaho Data Integration tools
- Set up kettle.properties
- Prepare the Pentaho repository
- Install the ETL
- Create the Data Warehouse

Install Pentaho Data Integration Tools

Pentaho is the third-party tool that is used for data integration. You can download Pentaho Data Integration (PDI) from the Deltek download server.

To install Pentaho data integration tools, you must:

- Do all items listed in Before You Begin

- Perform Pentaho Installation
    - Server: MS SQL Server or Oracle Server
    - Client
- Activate the PDI license

Before You Begin

Before you begin the installation of Pentaho, you must do the following:

- Uninstall previous versions of Pentaho.
- Delete the folders .kettle and .Pentaho, typically located on **C:\Users\<username>.**
- Ensure that J2SE Runtime Environment is installed and that the PATH environment variable references the Java SE folder. For 64-bit operating systems, you need to have JRE 32-bit installed.

Perform Pentaho Installation

There are various ways to install Pentaho: Evaluation, Archive, Manual, or Client tools only. This guide includes instructions to install the Pentaho server via archive installation, and the client via manual installation.

> **Note:** More information about Pentaho installation is available in the Pentaho documentation. Steps to upgrade Pentaho can be found here.

Prepare the environment

This section describes how to prepare your Windows system for an archive installation of Pentaho.

> **Note:** More information is available here.

1. Create the folder <Installation_directory>/Pentaho/server.
2. Download and install a supported JRE or JDK version.
    a. Set the PENTAHO_JAVA_HOME environment variable to the JRE or JDK folder path.
        i. Go to the Windows **Settings** » **System** » **About** » **Advanced system properties**.
        ii. Click **Environment Variables** and add or edit the path, for example:

        ```
        PENTAHO_JAVA_HOME=C:\Program Files\Java\jdk11.x.x.x
        ```

    b. Verify the settings are in place:

        ```
        ECHO %PENTAHO_JAVA_HOME%
        ```

3. Install the Pentaho Server Database. This guide includes instructions for MS SQL and Oracle.
    a. Download and extract the server installation files:

- pentaho-server-ee-9.3.0-dist.zip — <Installation_directory>\Pentaho\server
- pdd-plugin-ee-9.3.0-dist.zip — <Installation_directory>\Pentaho\server\pentaho-server\pentaho-solutions\system
- pir-plugin-ee-9.3.0-dist.zip — <Installation_directory>\Pentaho\server\pentaho-server\pentaho-solutions\system
- paz-plugin-ee-9.3.0-dist.zip — <Installation_directory>\Pentaho\server\pentaho-server\pentaho-solutions\system
- pentaho-operations-mart-9.3.0-dist.zip — <Installation_directory>\Pentaho\server\pentaho-server\data

b. Verify that you have the following folders:

- pentaho\server\pentaho-server\pentaho-solutions\system\analyzer
- pentaho\server\pentaho-server\pentaho-solutions\system\dashboards
- pentaho\server\pentaho-server\pentaho-solutions\system\pentaho-interactive-reporting
- pentaho\server\pentaho-server
- pentaho\server\pentaho-server\data\<database name>

Using MS SQL Server as Pentaho Server Database

This section describes how to change the backend database of Pentaho Server from the default PostgreSQL installation to MS SQL Server. This is **not** required when using MS SQL Server as data source for the ETL.

> **Note:** More information is available here.

Prepare the environment

1. Extract pentaho-server-ee-9.3.0.0-428-dist.zip.
2. Run **installer.bat** as Administrator.
3. Accept the license term agreement, then click **Next**.
4. Change the installation path to **<Installation_directory>/Pentaho/server**.
5. A warning message that the directory already exists appears. Click **Yes**.
   Any existing files in the directory will be retained.

Configuring MS SQL Pentaho Repository Database

> **Note:** More information is available here.

1. Adjust MS SQL Server configuration settings:

   a. **Use mixed authentication** – on MS SQL Server Management Studio object explorer, right-click the server then go to **Properties** » **Security**, and select **SQL Server** and **Windows Authentication** mode.

b. **Enable TCP/IP for MS SQL Server** – on **SQL Server Configuration Manager** » **SQL Server Network Configuration** » **Protocols** for MSSQLSERVER, right-click **TCP/IP** and click **Enable**.

c. **Configure MS SQL Server to listen on an external IP and not localhost** – on **SQL Server Configuration Manager** » **SQL Server Network Configuration** » **Protocols** for MSSQLSERVER, right-click **TCP/IP** » **Properties** » **IP Addresses** and update the IP Addresses.

2. Navigate to <INSTALL_DIR>/Pentaho/server/pentaho-server/data/sqlserver to find the following SQL scripts:

   ▪ create_jcr_sqlServer.sql

   ▪ create_repository_ sqlServer.sql

   ▪ create_quartz_ sqlServer.sql

3. To change the passwords, go to the <INSTALL_DIR>/Pentaho/server/pentaho-server/data/sqlserver directory and use any text editor to change the passwords in the SQL scripts from step 2.

4. Run SQL Scripts using Microsoft SQL Server Management Studio in the following order:

   a. **File** » **Open** » **File…**.

   b. Select all the scripts.

   c. Execute each script.

   Alternatively, you can run the scripts using sqlcmd (Requires admin privileges):

   ▪ -i <filepath to DDL>/create_quartz_sqlServer.sql

   ▪ -i <filepath to DDL>/create_repository_sqlServer.sql

   ▪ -i <filepath to DDL>/create_jcr_sqlServer.sql

5. Verify that databases and user roles have been created:

   a. Open MS SQL Server Management Studio.

   b. In the Object Explorer section of the window, make sure that the Quartz, Jackrabbit (JCR), Hibernate, and Pentaho Operations Mart databases are present.

   c. Navigate to **Security** » **Logins** and make sure that the appropriate users have been created.

   d. Exit the MS SQL Server Management Studio tool.

Set Up Quartz

1. Open the <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/quartz/quartz.properties file in any text editor.

2. Locate the #_replace_jobstore_properties section and set the org.quartz.jobStore.driverDelegateClass as shown here.

```
org.quartz.jobStore.driverDelegateClass =
org.quartz.impl.jdbcjobstore.oracle.MSSQLDelegate
```

3. Locate the # Configure Datasources section and set the org.quartz.dataSource.myDS.jndiURL equal to Quartz.

```
org.quartz.dataSource.myDS.jndiURL = Quartz
```

4. Save the file and close the text editor.

Set Hibernate Settings

1. Open <INSTALL_DIR>/Pentaho/server/pentaho-server/pentaho-solutions/system/hibernate/hibernate-settings.xml file in any text editor.

2. Change the <config-file>:
   From:

```
<config-file>system/hibernate/postgresql.hibernate.cfg.xml</config-file>
```

   To:

```
<config-file>system/hibernate/sqlserver.hibernate.cfg.xml</config-file>
```

3. Save the file and close the text editor.

Set Audit Log file with SQL Server version

1. Copy the file <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/dialects/sqlserver/audit_sql.xml.

2. Paste and replace the existing audit_sql.xml in <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system.

Set Jackrabbit Repository Information

1. Open <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/jackrabbit/repository.xml file in any text editor

2. Comment out (<!—code -->) the code section for PostgreSQL and uncomment the MS SQL Server counterpart of the following and then update the parameters:

Code Section
Repository
``` <FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem"> <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/> <param name="url" value="jdbc:sqlserver://<Server>:<Port>;DatabaseName=jackrabbit"/> <param name="user" value="jcr_user"/> <param name="password" value="password"/> <param name="schemaObjectPrefix" value="fs_repos_"/> </FileSystem> ```
DataStore

```xml
<DataStore class="org.apache.jackrabbit.core.data.db.DbDataStore">
 <param name="url"
value="jdbc:sqlserver://<Server>:<Port>;DatabaseName=jackrabbit"/>
 <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="databaseType" value="mssql"/>
 <param name="minRecordLength" value="1024"/>
 <param name="maxConnections" value="3"/>
 <param name="copyWhenReading" value="true"/>
 <param name="tablePrefix" value=""/>
 <param name="schemaObjectPrefix" value="ds_repos_"/>
</DataStore>
```

Workspaces

```xml
<FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem">
 <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
 <param name="url"
value="jdbc:sqlserver://<Server>:<Port>;DatabaseName=jackrabbit"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schema" value="mssql"/>
 <param name="schemaObjectPrefix" value="fs_ws_"/>
</FileSystem>
```

Persistence Manager (Part 1)

```xml
<PersistenceManager
class="org.apache.jackrabbit.core.persistence.bundle.MSSqlPersistenceManager"
>
 <param name="url"
value="jdbc:sqlserver://<Server>:<Port>;DatabaseName=jackrabbit "/>
 <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver "/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schema" value="mssql"/>
 <param name="schemaObjectPrefix" value="${wsp.name}_pm_ws_"/>
</PersistenceManager>
```

Versioning

```xml
<FileSystem class="org.apache.jackrabbit.core.fs.db.MSSqlFileSystem">
 <param name="url"
value="jdbc:sqlserver://<Server>:<Port>;DatabaseName=jackrabbit"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schema" value="mssql"/>
<param name="schemaObjectPrefix" value="fs_ver_"/>
</FileSystem>
```

Persistence Manager (Part 2)

```xml
<PersistenceManager
class="org.apache.jackrabbit.core.persistence.bundle.MSSqlPersistenceManager"
>
 <param name="url"
value="jdbc:sqlserver://<Server>:<Port>;DatabaseName=jackrabbit"/>
 <param name="driver" value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schema" value="mssql"/>
 <param name="schemaObjectPrefix" value="pm_ver_"/>
</PersistenceManager>
```

Note: The password for jcr_user is the one defined in create_jcr_ora.sql.

3. Comment out the DatabaseJournal code section:

```
<!--
<Cluster id="node1">
<Journal class="org.apache.jackrabbit.core.journal.MSSqlDatabaseJournal">
<param name="revision" value="${rep.home}/revision.log" />
<param name="url" value="java:comp/env/jdbc/jackrabbit"/>
<param name="driver" value="javax.naming.InitialContext"/>
<param name="schema" value="mssql"/>
<param name="schemaObjectPrefix" value="cl_j_"/>
<param name="janitorEnabled" value="true"/>
<param name="janitorSleep" value="86400"/>
<param name="janitorFirstRunHourOfDay" value="3"/>
</Journal></Cluster>
-->
```

4. Apply JDBC Drivers to the Pentaho Server.

   a. Download the Microsoft JDBC Driver for SQL Server.

   b. Copy the jar file to <INSTALL_DIR>/pentaho/server/pentaho-server/tomcat/lib.

Update Tomcat Context XML File with JDBC Connection

1. Open <INSTALL_DIR>/pentaho/server/pentaho-server/tomcat/webapps/pentaho/META-INF/context.xml in any text editor.

2. Comment out the existing <Resources> in the file and add the following lines:

```
<Resource
validationQuery="select 1"
url="jdbc:sqlserver:// <Server>:<Port>;DatabaseName=hibernate"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
password="password" username="hibuser"
initialSize="0" maxActive="20" maxIdle="10" maxWait="10000"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory"
type="javax.sql.DataSource" auth="Container" name="jdbc/Hibernate"/>

<Resource
validationQuery="select 1"
url="jdbc:sqlserver:// <Server>:<Port>;DatabaseName=hibernate"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
password="password" username="hibuser"
initialSize="0" maxActive="20" maxIdle="10" maxWait="10000"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory
type="javax.sql.DataSource" auth="Container" name="jdbc/Audit"/>

<Resource
validationQuery="select 1"
url="jdbc:sqlserver:// <Server>:<Port>;DatabaseName=quartz"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
password="password" username="pentaho_user"
initialSize="0" maxActive="20" maxIdle="10" maxWait="10000"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory
type="javax.sql.DataSource" auth="Container" name="jdbc/Quartz"/>
```

```
<Resource
validationQuery="select 1"
url="jdbc:sqlserver:// <Server>:<Port>;DatabaseName=pentaho_operations_mart"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
password="password" username="pentaho_operations_mart"
initialSize="0" maxActive="20" maxIdle="10" maxWait="10000"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory
type="javax.sql.DataSource" auth="Container"
name="jdbc/pentaho_operations_mart"/>

<Resource
validationQuery="select 1"
url="jdbc:sqlserver:// <Server>:<Port>;DatabaseName=pentaho_operations_mart"
driverClassName="com.microsoft.sqlserver.jdbc.SQLServerDriver"
password="password" username="pentaho_operations_mart"
initialSize="0" maxActive="20" maxIdle="10" maxWait="10000"
factory="org.apache.tomcat.jdbc.pool.DataSourceFactory
type="javax.sql.DataSource" auth="Container"
name=" jdbc/PDI_Operations_Mart "/>
```

> **Note**: **jdbc/pentaho_operations_mart** and **jdbc/PDI_Operations_Mart** resources needs to be included even if it the pentaho_mart_oracle.sql was not run.

Clean Pentaho Server Repository Cache

The evaluation installation created samples that are still referencing the previous PostgreSQL database. This need to be removed for Pentaho Repository to recreate the cache with the right database references. Follow these simple instructions:

1. Stop Pentaho Server.

    a. Open a command prompt.

    b. Run stop-pentaho.bat in <INSTALL_DIR>/server/pentaho-server/.

2. Navigate to <INSTALL_DIR>/server/pentaho-server/pentaho-solutions/system/jackrabbit/.

3. Rename or delete the repository folder.

Start Pentaho Server

1. Open a command prompt.

2. Run **start-pentaho.bat** as Administrator in <INSTALL_DIR>/server/pentaho-server/.

3. Verify that the server is running by opening Pentaho User Console. In any browser open http:<hostname>:<port>/pentaho.

Using Oracle Server as Pentaho Server Database

This section will describe how to change the backend database of Pentaho Server from the default PostgreSQL installed by the evaluation installation to Oracle database. This is **not** required when using Oracle database as a data source for the ETL.

> **Note:** More information is available here.

Prepare the environment

1. Extract pentaho-server-ee-9.3.0.0-428-dist.zip.

2. Run installer.bat as Administrator.

3. Accept the license term agreement, then click **Next**.

4. Change the installation path to <Installation_directory>/Pentaho/server.

5. A warning message that the directory already exists appears. Click **Yes**. Any existing files in the directory will be retained.

Configuring Oracle Pentaho Repository Database

> **Note:** More information is available [here](#).

Initialize Values

1. Navigate to <INSTALL_DIR>\server\pentaho-server\data\oracle10g to find the following SQL scripts:

   - create_jcr_ora.sql
   - create_repository_oral.sql
   - create_quartz_ora.sql

2. Edit the above files and uncomment the following lines:

   -- conn admin/password@pentaho

3. Update the files with the proper database credentials

   Example when connecting to a remote host:

   ```
   conn admin/password@(DESCRIPTION=(ADDRESS = (PROTOCOL = TCP)(HOST =
   XXXXXXX)(PORT = 1521))(CONNECT_DATA=(SERVER = DEDICATED)(SERVICE_NAME =
   XXXXXXX)))
   ```

4. Update the default passwords in the SQL scripts.

   ```
   create user jcr_user identified by "password" ...
   ```

5. Run SQL Scripts using SQL*Plus (or any Oracle SQL development tool) in the following order:

   - start <INSTALL_DIR>/server/pentaho-server/data/oracle10g/create_jcr_ora.sql
   - start <INSTALL_DIR>/server/pentaho-server/data/oracle10g/create_repository_ora.sql
   - start <INSTALL_DIR>/server/pentaho-server/data/oracle10g/create_quartz_ora.sql
   - start <INSTALL_DIR>/server/pentaho-server/data/oracle10g/pentaho_mart_ora.sql (Optional)

6. Verify that databases and user roles have been created by running the following in SQL*Plus (or any Oracle SQL development tool):

   ```
   SELECT USERNAME FROM DBA_USERS;
   ```

Set Up Quartz

1. Open the <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/quartz/quartz.properties file in any text editor.

2. Locate the #_replace_jobstore_properties section and set the org.quartz.jobStore.driverDelegateClass as shown here.

```
org.quartz.jobStore.driverDelegateClass =
org.quartz.impl.jdbcjobstore.oracle.OracleDelegate
```

3. Locate the # Configure Datasources section and set the org.quartz.dataSource.myDS.jndiURL equal to Quartz, like this.

```
org.quartz.dataSource.myDS.jndiURL = Quartz
```

4. Save the file and close the text editor.

Set Up Hibernate Settings

1. Open <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/hibernate/hibernate-settings.xml file in any text editor and then change the <config-file>.

From:

```
<config-file>system/hibernate/postgresql.hibernate.cfg.xml</config-file>
```

To:

```
<config-file>system/hibernate/oracle10g.hibernate.cfg.xml</config-file>
```

2. Open <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/hibernate/oracle10g.hibernate.cfg.xml and update the following properties:

```
<property name="connection.driver_class">oracle.jdbc.driver.OracleDriver</property>
<property name="connection.url">jdbc:oracle:thin:@<host>:<port>:<SID></property>
<property name="dialect">org.hibernate.dialect.Oracle10gDialect</property>
<property name="connection.username">hibuser</property>
<property name="connection.password"><password></property>
<property name="connection.pool_size">10</property>
<property name="show_sql">false</property>
<property name="hibernate.jdbc.use_streams_for_binary">true</property>
```

> **Note:** Connection.username and connection.password is from create_repository_ora.sql.

Set audit log file with Oracle version

1. Copy the file <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/oracle10g/audit_sql.xml.

2. Paste and replace the existing audit_sql.xml in <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/.

Set Jackrabbit Repository Information

1. Open <INSTALL_DIR>/pentaho/server/pentaho-server/pentaho-solutions/system/jackrabbit/repository.xml file in any text editor

2. Comment out (<!—code -->) the code section for PostgreSQL and uncomment the Oracle counterpart of the following and then update the parameters:

Code Section
Repository

```xml
<FileSystem class="org.apache.jackrabbit.core.fs.db.OracleFileSystem">
 <param name="url" value="jdbc:oracle:thin:@<host>:<port>/<SID>"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schemaObjectPrefix" value="fs_repos_"/>
 <param name="tablespace" value="pentaho_tablespace"/>
</FileSystem>
```

DataStore

```xml
<DataStore class="org.apache.jackrabbit.core.data.db.DbDataStore">
 <param name="url" value=" jdbc:oracle:thin:@<host>:<port>/<SID>"/>
 <param name="driver" value="oracle.jdbc.OracleDriver"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="databaseType" value="oracle"/>
 <param name="minRecordLength" value="1024"/>
 <param name="maxConnections" value="3"/>
 <param name="copyWhenReading" value="true"/>
 <param name="tablePrefix" value=""/>
 <param name="schemaObjectPrefix" value="ds_repos_"/>
</DataStore>
```

Workspaces

```xml
<FileSystem class="org.apache.jackrabbit.core.fs.db.OracleFileSystem">
 <param name="url" value="jdbc:oracle:thin:@<host>:<port>/<SID>"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schemaObjectPrefix" value="fs_ws_"/>
 <param name="tablespace" value="pentaho_tablespace"/>
</FileSystem>
```

Persistence Manager (Part 1)

```xml
<PersistenceManager
class="org.apache.jackrabbit.core.persistence.bundle.OraclePersistenceManager">
 <param name="url" value="jdbc:oracle:thin:@<host>:<port>/<SID>"/>
 <param name="driver" value="oracle.jdbc.OracleDriver"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schema" value="oracle"/>
 <param name="schemaObjectPrefix" value="${wsp.name}_pm_ws_"/>
 <param name="tablespace" value="pentaho_tablespace"/>
</PersistenceManager>
```

Versioning

Code Section

```
<FileSystem class="org.apache.jackrabbit.core.fs.db.OracleFileSystem">
 <param name="url" value="jdbc:oracle:thin:@<host>:<port>/<SID>"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schemaObjectPrefix" value="fs_ver_"/>
 <param name="tablespace" value="pentaho_tablespace"/>
</FileSystem>
```

Persistence Manager (Part 2)

```
<PersistenceManager
class="org.apache.jackrabbit.core.persistence.bundle.OraclePersistenceManager
">
 <param name="url" value="jdbc:oracle:thin:@<host>:<port>/<SID>"/>
 <param name="driver" value="oracle.jdbc.OracleDriver"/>
 <param name="user" value="jcr_user"/>
 <param name="password" value="password"/>
 <param name="schema" value="oracle"/>
 <param name="schemaObjectPrefix" value="pm_ver_"/>
 <param name="tablespace" value="pentaho_tablespace"/>
</PersistenceManager>
```

> **Note:** The password for jcr_user is the one defined in create_jcr_ora.sql.

3. Comment out the DatabaseJournal code section:

```
<!--
<Cluster id="node1">
 <Journal class="org.apache.jackrabbit.core.journal.DatabaseJournal">
   <param name="revision" value="${rep.home}/revision.log" />
   <param name="url" value="jdbc:postgresql://localhost:5432/jackrabbit"/>
   <param name="driver" value="org.postgresql.Driver"/>
   <param name="user" value="jcr_user"/>
   <param name="password" value="password"/>
   <param name="schema" value="postgresql"/>
   <param name="schemaObjectPrefix" value="cl_j_"/>
   <param name="janitorEnabled" value="true"/>
   <param name="janitorSleep" value="86400"/>
   <param name="janitorFirstRunHourOfDay" value="3"/>
 </Journal>
</Cluster>
-->
```

Apply JDBC Drivers to the Pentaho Server

1. Download the ojdbc8.jar oracle driver.
2. Copy the jar file to <INSTALL_DIR>/pentaho/server/tomcat/lib.

Update Tomcat Context XML File with JDBC Connection

1. Open <INSTALL_DIR>/pentaho/server/tomcat/webapps/pentaho/META-INF/context.xml in any text editor.

2. Comment out the existing <Resources> in the file and add the following lines:

```
<Resource
validationQuery="select 1 from dual"
url="jdbc:oracle:thin:@<host>:<port>/<SID>"
driverClassName="oracle.jdbc.OracleDriver"
password="password" username="hibuser"
maxWaitMillis="10000" maxIdle="5" maxTotal="20"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
type="javax.sql.DataSource" auth="Container" name="jdbc/Hibernate"/>

<Resource
validationQuery="select 1 from dual"
url="jdbc:oracle:thin:@<host>:<port>/<SID>"
driverClassName="oracle.jdbc.OracleDriver"
password="password" username="hibuser"
maxWaitMillis="10000" maxIdle="5" maxTotal="20"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
type="javax.sql.DataSource" auth="Container" name="jdbc/Audit"/>

<Resource
validationQuery="select 1 from dual"
url="jdbc:oracle:thin:@<host>:<port>/<SID>"
driverClassName="oracle.jdbc.OracleDriver"
password="password" username="quartz"
maxWaitMillis="10000" maxIdle="5" maxTotal="20"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
type="javax.sql.DataSource" auth="Container" name="jdbc/Quartz"/>

<Resource
validationQuery="select 1 from dual"
url="jdbc:oracle:thin:@<host>:<port>/<SID>"
driverClassName="oracle.jdbc.OracleDriver"
password="password" username="hibuser"
maxWaitMillis="10000" maxIdle="5" maxTotal="20"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
type="javax.sql.DataSource" auth="Container"
name="jdbc/pentaho_operations_mart"/>

<Resource
validationQuery="select 1 from dual"
url="jdbc:oracle:thin:@<host>:<port>/<SID>"
driverClassName="oracle.jdbc.OracleDriver"
password="pentaho_operations_mart" username="pentaho_operations_mart"
maxWaitMillis="10000" maxIdle="5" maxTotal="20"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
type="javax.sql.DataSource"
auth="Container" name="jdbc/PDI_Operations_Mart"/>
```

Note: Jdbc/pentaho_operations_mart and **jdbc/PDI_Operations_Mart** resources needs to be included even if it the pentaho_mart_oracle.sql was not run.

Clean Pentaho Server Repository Cache

The evaluation installation created samples that are still referencing the previous PostgreSQL database. This need to be removed for Pentaho Repository to recreate the cache with the right database references.

Follow these simple instructions:

1. Stop Pentaho Server.

    a. Open a command prompt.

    b. Run stop-pentaho.bat in <INSTALL_DIR>/server/pentaho-server/.

2. Navigate to <INSTALL_DIR>/server/pentaho-server/pentaho-solutions/system/jackrabbit/.

3. Rename or delete the repository folder.

Start Pentaho Server

1. Open a command prompt.

2. Run start-pentaho.bat as Administrator in <INSTALL_DIR>/server/pentaho-server/.

3. Verify that the server is running by opening Pentaho User Console. In any browser open http:<hostname>:<port>/pentaho.

Install the PDI Client

This section includes steps to install the Pentaho Data Integration client manually.

To install the PDI client:

1. Extract pdi-ee-client-9.3.0.0-428-dist.zip.

2. Run **installer.bat** as Administrator.

3. Accept the license term agreement, then click **Next**.

4. Change the installation path to your preferred **<Installation_directory>** and click **Next**.

5. After the installation is complete, click **Quit**.

6. Create a folder for tools and utilities, for example <INSTALL DIR»/pentaho/design-tools, and copy extracted files here.

Activate PDI License

To activate the PDI License

1. Log in to the Pentaho User Console.

2. Click the drop down menu on the upper left side of the screen and select **Administration**.

3. Select the **Licenses** tab.

4. In the upper right corner, click the **+** icon.

5. Select the following files:

    ▪ Pentaho BI Platform Enterprise Edition.lic

    ▪ Pentaho PDI Enterprise Edition.lic

Alternatively, run **Install_license.bat** from **<Installation directory>\Pentaho\license-installer** in a command prompt or shell. Input the directory of the license files in the command, for example:

Set Up Kettle.Properties

The ETL for Business Performance Management Analysis utilizes the kettle.properties file to provide default values to variables used when running Jobs and/or Transformations. This section will describe the parameters that need to be added and the values they should contain.

The kettle.properties file is located at the following directories by default:

- For Windows: C:\Users\<username>\.kettle\
- For Unix/Linux: .kettle folder is placed in the $HOME folder.

If multiple users are to share the properties, we recommend you create a copy of the file to another commonly accessible folder. Create a new system environment variable KETTLE_HOME and assign the path of the file to that variable. After this is done and a Pentaho application is run, the folder .kettle is created automatically in the path designated by KETTLE_HOME. Make sure to copy the .kettle.properties file down into that folder.

To set up kettle.properties, you must:

- Set up parameters for BPM.
- Test the value is set correctly.

Set Up Parameters for BPM

To set up parameters for BPM, follow these steps:

1. Open kettle.properties in any text editor.
2. Remove all contents of the kettle.properties, if it is automatically generated. The evaluation installer includes pre-defined log schema in the kettle.properties file which conflicts with the logging configuration of the ETL provided by Deltek.
3. If the Maconomy database runs Oracle, add the parameter:

   ```
   MAC_CONNECTION_TYPE=ORACLE
   ```

   If the Maconomy database runs MS SQL Server, add the parameter:

   ```
   MAC_CONNECTION_TYPE=SQL
   ```

4. If the Data Warehouse runs Oracle, add the parameters:

   ```
   DWH_CONNECTION_TYPE=ORACLE
   DWH_DEFAULT_SCHEMA=
   ```

   > **Note:** The parameter "DWH_DEFAULT_SCHEMA=" has no value after the equal sign.

   If the Data Warehouse runs MS SQL Server, add the parameters and values:

   ```
   DWH_CONNECTION_TYPE=SQL
   DWH_DEFAULT_SCHEMA=dbo
   ```

5. Add the parameter stating whether to run a *full load* or an *incremental load*.

   ```
   FULL_LOAD=YES
   ```

The FULL_LOAD parameter is not used in the pure standard flows of creating a data warehouse or performing a full or incremental load. These Pentaho jobs set the parameter themselves. But if other jobs are run the parameter may be read.

> **Note:** Pentaho tools read these variables when starting. Whenever a Pentaho tool is open at the moment of resetting the variables, you must restart the tool to use new variable settings.

Test the Value is Set Correctly

To test that the value has been set correctly, run the Test FULL_LOAD variable transformation. The job is part of the BPM Analysis ETL and will be available after you have installed that in the Install the ETL section.

Prepare the Database

The Pentaho Data Integration tool requires the creation of the actual data warehouse database. To prepare the Pentaho repository, you must:

- Create database users in either Oracle or SQL Server.
- Create storage for BPM People Planner.

Create Database Users

To create database users in Oracle:

1. From the command prompt, run the following:

   ```
   sqlplus system/manager
   ```

   It may be necessary to specifically connect to the database to be used.

2. In sqlplus, enter the following:

   ```
   create user MACONOMY_DWH identified by MACONOMY_DWH;
   grant dba to MACONOMY_DWH;
   ```

3. If you do not want to use default database settings for the user's default table space and temporary table space, you can change them with statements such as the following in sqlplus:

   ```
   alter user MACONOMY_DWH default table space <x1> temporary table space <y1>;
   ```

   where $<x1>$, $<x2>$, $<y1>$, $<y2>$ should be replaced by relevant values.

4. To confirm that the settings work, enter the following:

   ```
   connect MACONOMY_DWH / MACONOMY_DWH;
   ```

6. Exit sqlplus.

To create database users in MS SQL Server:

1. Log in to Microsoft SQL Server Management Studio with the user name **sa**.
2. Right-click **Databases** and create the database **MACONOMY_DWH**.
3. In the Options tab, set collation to **SQL_Latin1_General_CP1_CS_AS**.

   > **Note:** This makes the SQL data warehouse case-sensitive to match with Maconomy database for an accurate comparison of data for the ETL.

4. Expand **Security**, right-click **Logins**, and select **New Login**.

5. Enter the name **MACONOMY_DWH** for the login.

6. Set **Authentication** to **SQL Server Authentication**.

7. Enter a password.

8. On the left-hand side of the window, click **User Mappings**.

9. On the User Mappings dialog box, map the database to the corresponding user.

10. Specify the role **db_owner**.

11. To check that the databases work, log out and then log in again with the data warehouse database credentials.

> **Note:** The Oracle database for the BPM data warehouse requires a table space called macindex_dwh, where indexes for the DWH are created. This table space must be created in addition to the normal table space of MACONOMY_DWH.
>
> For more information, see Build the BPM Data Warehouse.

Create Storage for BPM People Planner

Use the steps above to create storage for BPM People Planner. Replace (what) with the user PEOPLEPLANNER_DWH.

For BPM People Planner other parameters in the kettle.properties file are needed:

-   #DWH_CONNECTION_TYPE = ORACLE
-   DWH_CONNECTION_TYPE = SQL
-   #PP_CONNECTION_TYPE = ORACLE
-   PP_CONNECTION_TYPE = SQL
-   #PP_STAND_ALONE = YES
-   PP_STAND_ALONE = NO
-   # How many years old AvailableHours data should be loaded into DWH
-   PP_ACTUAL_YEAR_RANGE_MIN = 2
-   # How many years in future since current date
-   # should be loaded for AvailableHours into DWH
-   PP_ACTUAL_YEAR_RANGE_MIN = 2

The "#" symbol means that all strings following it are comments. By changing its position, you can switch which RDBMS system to use to access the DWH database and which one to use for PeoplePlanner.

Parameter PP_STAND_ALONE defines how PeoplePlanner is used, meaning, is it used in integration with Maconomy system or as a standalone system.

> **Note:** Existing Business Performance Management PeoplePlanner Reporting assumes that PeoplePlanner is used in integration with Maconomy – so, the reports do not work for a stand-alone solution (in this case you will be able to use the DWH only).

The last two parameters define how many years before and after the current date should be loaded in the DWH.

> **Warning:** Remember that for every additional year, a significant number of additional data is loaded. Therefore, use these parameters with care and only include the number of years required to cover the user's business needs.

Install the ETL

The ETL (Extract-Transform-Load) is the mechanism which reads (extracts) data from Maconomy or People Planner, transforms it, and stores it (loads it) in the data warehouse database.

The steps below take origin in a data warehouse on Oracle. If the data warehouse is on MS SQL Server, change the connection type from Oracle to MS SQL Server accordingly.

Furthermore, the steps assume the setup of a data warehouse for BPM Analysis (Maconomy). If a data warehouse is set up for BPM People Planner, import the DataIntegrationRepository.xml from the peopleplanner folder in the RPU.

To install the ETL, you must:

- Connect to a Pentaho Server

- Import the BPM ETL. The file name is DataIntegrationRepository.xml.
    - Standard ETL is under etl\reporting
    - PeoplePlanner ETL is under etl\peopleplanner

Connecting to a Pentaho Server

> **Note:** When running ETL on a server that has its time zone set to Singapore, an issue occurs with the Gregorian Calendar when running Create Tables and Views. This affects Pentaho 8.2 and 9.3. The workaround is to change the time zone of the Pentaho server to a different UTC+08:00 time zone, for example, Beijing.
>
> The issue is caused by the time zone itself and affects other applications worldwide.

To connect to a Pentaho Server, follow these steps:

1. Go to **Start > Pentaho Enterprise Edition > Data Integration**

2. Click the **Connect** button on the upper-right corner of the window.

   a. For the first connection, it will open the Repository Manager automatically.

   b. For subsequent connections, select **Repository Manager** in the drop-down list, then click **Add**.

3. Click **Get Started** on the Repository Manager window.

4. Fill in the details:

   a. **Display Name** – name to identify the connection.

   b. **URL** – URL to open Pentaho User console. Make sure to enter the correct port no.

   c. **Description** – additional information to help identify the connection

5. Click **Finish.**

6. A test will be automatically performed, if successful click **Connect Now**. Otherwise, review the connection details.

7. Log in with username **admin** and password **password**.

Import the BPM ETL

To import the BPM ETL, follow these steps:

1. Select **Tools > Repository > Import Repository...**

2. Browse to the location of the *BPM Analysis* or *BPM People Planner* ETL XML file.
   The file is named DataIntegrationRepository.xml.

3. At the prompt (Do you want to apply a set of rules to the import?) select **No**.

4. Select the root as directory (default). Click **OK**.

5. Click **Yes** to create the suggested folders. The transformations and jobs are now imported.

6. Click **Close**.

Create the Data Warehouse

This section describes how to create and load the data warehouse. To prepare for creating the data warehouse, you must set up the connections to the Maconomy/People Planner database and the data warehouse database. Then you must run the ETL job that creates the data warehouse itself. All this is done in Spoon.

To create the data warehouse, you must:

- Set up a connection to Maconomy / People Planner
- Set up the Table Space
- Create the date warehouse

Set Up a Connection to Maconomy / People Planner

To set up a connection to Maconomy / People Planner, follow these steps:

1. Select **Tools > Repository > Explore...**

2. Switch to the **Connections** tab.

3. Select the **Maconomy** connection and click .

   Fill in the following fields:

   - **Host Name** – Enter the name of the Maconomy / People Planner server.
   - **Database Name** – Enter the name of the Oracle SID or MS SQL Server database name.
   - **Tablespace for Data** – Enter the name of the tablespace if the default is not used.
   - **Tablespace for Indices** – Enter the name of the tablespace for indices if the default is not used.
   - **Port Number** – Enter the port of Maconomy, typically 1521.
   - **User Name** – Enter the name of the Maconomy database user (short name).
   - **Password** – Enter the password for accessing the database user.

4. Click **Test** to test the connection. Click **OK.**

5. Select the **DWH** connection and click .

6. Fill in the following fields:

   - **Host Name** – Enter the name of the Data warehouse server.
   - **Database Name** – Enter the name of the Oracle SID or MS SQL Server database name.
   - **Tablespace for Data** – Enter the name of the tablespace if the default is not used.
   - **Tablespace for Indices** – Enter the name of the tablespace for indices if the default is not used.
   - **Port Number** – Enter the port of Maconomy, typically 1521.
   - **User Name** – Enter MACONOMY_DWH.
   - **Password** – Enter MACONOMY_DWH.

7. Click **Test** to test the connection. Click **OK.**

8. Click **Close**.

Set Up the Table Space

For BPM People Planner, set up the right table space.

To set up the table space, follow these steps:

1. In this transformation you should edit either do the **Oracle DWH Variables** step or the **Oracle DWH Variables** step (depending upon your DWH connection type) and set the right tablespace for the constant **DWH_INDEX_TABLESPACE**:

2. Click **OK** and **Save**.

Create the Data Warehouse

Before You Begin

Before you begin this section, note the following:

- Because the **Create** job starts by dropping the full data warehouse, it should **not** be used for upgrading a data warehouse.

- Drop indexes on facts could fail the first time you run this job. If the enhanced reporting views are not in place, the job fails. This may take a long time to finish, depending on the size of the Maconomy database.

- Only run the **Drop** or **DropTablesAndViews** jobs if you really want to remove the full data warehouse completely. Performing a full load does not require running DropTablesAndViews first.

To create the data warehouse, follow these steps:

1. Select **Tools > Repository > Explore.**

2. Double-click the job **Create.**

> **Note:** For *BPM People Planner* the jobs are **PP_CreateTablesAndViews** and **PP_LoadData**.

3. Run this job.

> **Note:** For *BPM People Planner* first run **PP_CreateTablesAndViews** and then **PP_LoadData**. The **PP_DropTablesAndViews** removes the People Planner data warehouse.

The above **Create** job drops the full data warehouse (in case one should already exist), then creates tables and views and performs a full load. This could take some time depending on the size of the Maconomy / People Planner database.

The job initiates by dropping the data warehouse. If this is the first time this is done, errors may occur because the job tries to drop something that does not exist. Ignore these errors.

> **Note:** To verify, open the command prompt, type **sqlplus maconomy_dwh/maconomy_dwh**, and then select **table_name** from user_tables.

> **Note:** The People Planner reports and universe use data from Maconomy data warehouse and therefore cannot be used when Business Performance Management BI for PP is installed in standalone. The universe for standalone mode has not been implemented at the moment.

Install Universes and Documents

Universes and documents are required for all three products: *BPM Reporting*, *BPM Analysis,* and *BPM People Planner*. The universes and documents for each product are contained in individual LCMBIAR files of the RPU.

The following procedure is the same for the LCMBIARs of *BPM Reporting*, *BPM Analysis*, and *BPM People Planner*.

To install universes and documents, you must:

- Import LCMBIAR files
- Set Universe Connection

Importing LCMBIAR File with Promotion Management

All BPM Analysis universes, reports, and dashboards are enclosed in the LCMBIAR files, which are imported using Promotion Management.

This section shows how Promotion Management can be used to deploy BPM reports on client systems. Promotion Management is a tool that is integrated with the Central Management Console to promote (or move) contents such as reports, universes, and connections to other systems. The source system can be either be another system or a LCMBIAR file. It can be especially useful for structures in which a client uses separate DEV, QA, and Production systems.

> **Note:** There is a similar tool for transferring content called Upgrade Management tool, but it is only available on the server, and its main function is for making LCMBIARs created from previous versions usable on BO 4.1. It must be modified to keep it from checking versions; otherwise, it throws a version check failed (UMT 20012) error.

Features

Server-to-Server

Promotion Management can be used to transfer contents from one server to another. It is similar to Import Wizard of BO XI 3.1, but it is web-based and integrated with the CMC.

LCMBIAR-to-Server (and Vice Versa)

LCMBIAR files can be imported to the CMS repository via Promotion Management, and exported from the CMS. You can do this via the command line.

Overrides

Overrides enable the automatic update of connection when contents are transferred from another server or LCMBIAR.

Rollback

You can restore a part of or the whole promoted content on the repository to its previous state. Note that this is only applicable if the source is another system and NOT a LCMBIAR.

Scheduling

You can schedule when content will move to another system.

LCM CLI

This command-line interface can be used to run Promotion Management features without the need to access CMC. This section shows the steps for how to import and export an LCMBIAR using the LCM CLI tool.

Implementation

To do a LCMBIAR-to-server promotion:

1. Log in to the Central Management Console at https://<server>:<port>/BOE/CMC.
2. On the CMC homepage, find the Manage island at the right side and click **Promotion Management**.
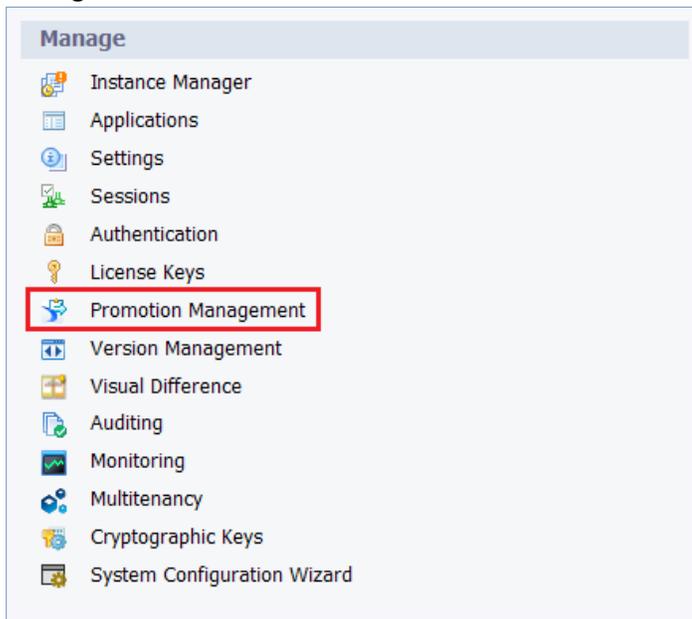
3. Click the Import drop-down and select **Import file**.
   The Import from file window appears.
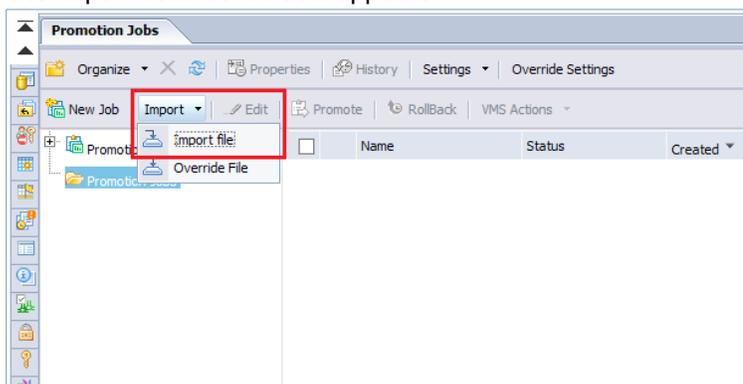
4. Select **File System**, then click **Browse**.

---

5. Select the LCMBIAR, then click **Open**.

6. The text beside the **Browse…** button is updated with the location and file name of the LCMBIAR. Click **OK** to create the new job.

> **Note:** A password can be set for LCMBIAR. Importing a password-protected LCMBIAR displays another window that allows a user to input the password.

7. A new job with the Source set as **From File** is automatically created. Update the **Name** and **Destination** of the new job and click **Create**.

Fields marked with an asterisk (*) are mandatory fields

Name*:	BIAR promotion
Description:	
Keywords:	
Save Job in*:	Promotion Jobs
Source :	From File ▾
Destination :	------------------------------- ▾

Create   Cancel

8. All objects from the LCMBIAR are pre-selected, including all dependencies. Click **Promote** to display a new window that contains the Summary, Security Settings, and Test Promote.

  ▪ **Select Security Settings** — Specify whether there are any rights that must be included for the objects to be promoted.

  ▪ **Select Test Promote** — Select this option to check whether there are any errors in the promotion before you click **Promote** again in Step 8 and commit to the changes.

9. Click **Promote** to start the transfer, or **Schedule** to have the promotion run at a later date (or a recurring schedule, depending on the need).

Setting Universe Connection

After you import the LCMBIAR file, universes and connections are stored in the CMS. To set database connection on the universes, you must create a project and a relational connection, retrieve universe files, assign a connection, and publish the universes.

This is all done in **Information Design Tool**. In **Information Design Tool,** a session to the CMS is required for certain operations. For information about sessions, as well as for more information and guides about creating projects, connections, and handling universe files, see documentation and tutorials on **Information Design Tool**.

> **Tip:** See SAP for details about Information Design Tool:
>
> http://scn.sap.com/docs/DOC-8461

To set a universe connection, you must:

- Create a project
- Create a relational connection
- Retrieve the universe files
- Set universe connections and publish

Create a Project

To create a project (skip this if a project is available already):

1. Select **File » New » Project**.

2. Enter a name for the project.

3. Select a **Project Location**.

   This is the folder where universe and connection files are stored in the project. By default, a folder under the user's profile is chosen. If other universe developers are to access the files, it is recommended that you select another folder (not under the user profile).

Create a Relational Connection

To create a relational connection (skip this if a connection is available already):

1. Select **File » New » Relational Connection**.

2. In the Resource Name window, enter a name for the connection. Click **Next**.

3. In the Database Middleware Driver Selection window, choose a database type.

   - BPM supports Oracle and MS SQL Server connections.
   - For BPM Reporting, you must choose the database type of the Maconomy database.
   - For BPM Analysis and BPM People Planner, you must choose the database type of the BPM Data Warehouse or People Planner Data Warehouse database, respectively.

4. In the Parameters window, enter the name of the database user, password, reference to TNS, service/database, port number, and SID/database. The parameters required depend on whether an Oracle or MS SQL Server connection is chosen, as well as whether a Client, ODBC, or JDBC connection is selected. Click **Next**.

5. Choose default settings for the next Parameters screen.

6. Set up the ALL_ROWS hint to speed up performance. Enter the following value for the custom property Hint:

   /*+ ALL_ROWS */

7. Enable Cursor Sharing, if needed. Enter the following value for the custom property ConnectInit.

- On Oracle connections:

  ALTER SESSION SET CURSOR_SHARING = FORCE

- On MS SQL Server connections:

  FORCE PARAMETERIZATION

> On many installations, performance is improved by applying Cursor Sharing. However, it can reduce performance on some systems.
>
> If the system has performance issues while Cursor Sharing is disabled, enable it to help improve performance. If the system has performance issues while Cursor Sharing enabled, disabling it may improve performance.

8. Click **Finish**. A new connection file (type cnx) is added to the project.

   A cnx connection is a local connection and can be used for publishing universes locally on disk. However, it is recommended that you publish universes on the CMS. This requires a secure connection.

9. To create a secure connection of the new connection file, right-click the cnx file and choose Publish Connection to a Repository.

10. In the window that appears, browse for the CMS location in which the connection is to be stored. For example, select the default Connections folder of the CMS. Click **Finish**.

11. A question window appears. Click **Yes** to retrieve a shortcut file to the secure connection. A secure connection file (type cns) is added to the project.

> **Note:**
>
> **ALL_ROWS Information**
>
> By default, the Maconomy database is set to retrieve a small number of records at a time for each query. This is convenient when entering and working in Maconomy workspaces or dialogs where you typically only see 10, 50, or 100 records at a time. However, when reporting, it is beneficial to retrieve all records at once, because a report document or dashboard is not refreshed until all records are retrieved.
>
> **Cursor Sharing Information**
>
> Often the reports that are run have the same query outline. Oracle and SQL Server can use this by saving a template of the queries that are executed. In many cases, this means that even if restrictions are changed (and the queries thereby differ), the database can skip the analysis of how to execute the queries, such as the creation of the explain plan. For many reports, the creation of the explain plan takes most of the time, so there is significant value in being able to reuse query templates.

Retrieve the Universe Files

To retrieve the universe files, follow these steps:

1. Make sure the project is selected in the Local Projects view.

2. Select **File » Retrieve a Published Universe » From a Repository...**

3. In the window that appears, browse to the location of the universes.
   If the universe is an old BO 3.1 universe, it has no extension; otherwise, it has a **unx** extension.
   Retrieving a universe of type **unv** migrates the universe to the new BI 4 **unx** format.
   If a BO 3.1 universe should be kept in the BO 3.1 **unv** format, set connection with the Universe Designer tool instead.

4. Select **Save for All Users**. Click **OK**.
   The Universe files (**blx** and **dfx**) are retrieved to a **timestamp** named folder under the project folder selected.

> **Note:** If needed, the universe files can be moved to another folder (for example, out of the **timestamp** named folder). However, this may break the local dependency that the **blx** file has to the **dfx** file and the local dependency that the **dfx** file has to the **cnx/cns** file. If this happens, open the **dfx** file and choose the connection again and possibly make some other minor change to force Information Design Tool to mark the **dfx** file as changed. Then save the file. Then open the **blx** file and choose **Change Data Foundation**. Here select the **dfx** file and possibly make some other small change to force Information Design Tool to mark the **blx** file as changed. Then save the file.

Set Universe Connections and Publish

To set universe connections and publish, follow these steps:

1. Open the **dfx** file of a universe.

2. Click the **Connections** tab.

3. Click **Change Connection...** .

4. Select the connection file (for example, the newly created **cns** connection shortcut file). Click **Finish**.

5. Save the **dfx** file.

6. Right-click the **blx** file and choose **Publish » To a Repository**.

7. Optionally, mark areas for checking integrity. Click **Next**.

8. Browse to the CMS folder in which the universe is to be published. Click **Finish**.
   The universe has now been published with the selected database connection.

9. Continue this process for all universes needed.

UPGRADE

Upgrade Workflow

This process of upgrading is divided into three main steps, described in the following.

1 • Upgrade ETL

2 • Upgrade Universes and Documents

3 • Upgrade Customizations

Upgrade ETL

Upgrading the ETL is required for upgrading *BPM Analysis* and *BPM People Planner*.

To run Create Job:

1. Import ETL.
2. After successful ETL import, open the repository and go to the /public/Business Performance Management folder.
3. Click **Create Job**.

The following occurs:

1. Drop all the DWH tables (including the deprecated tables in previous versions, namely INDEX_MANAGEMENT, TABLE_MANAGEMENT,UPGRADE_LOG).
2. The views, dimension, and fact tables are then recreated for the current version while INDEX_MANAGEMENT, TABLE_MANAGEMENT, and UPGRADE_LOG are longer available.
3. The job then proceeds to load the FULL data from Maconomy to the DWH.

To upgrade the ETL you must:

- Upgrade the ETL and BPM Data Warehouse
- Upgrade the ETL and People Planner Data Warehouse

Before You Begin

Before upgrading, it is recommended that you make a backup of the ETL and a backup of the data warehouse database.

> **Note:** Since there are many methods to back up a data warehouse database, and methods differ for Oracle and MS SQL, consult your Database Administrators for this task.

Back Up the ETL

To back up the ETL, follow these steps:

1. Run **Spoon.bat**.
2. Select **Tools » Repository » Connect...**
3. In the **Repository Connection** window, select the repository that contains the ETL to back up.
4. Select **Tools » Repository » Export Repository…**
5. Choose a location and a file name for the XML file to be created.
6. Click **Save**.

Upgrade BPM ETL and BPM Data Warehouse

Run Create Job from the ETL to drop and re-create the data warehouse during upgrades.

To run Create Job:

1. Import ETL.
2. After successful ETL import, open the repository and go to the /public/Business Performance Management folder.
3. Click **Create Job**.

The following occurs:

- Drop all the DWH tables (including the deprecated tables in previous versions, namely INDEX_MANAGEMENT, TABLE_MANAGEMENT,UPGRADE_LOG).
- The views, dimension, and fact tables are then recreated for the current version while INDEX_MANAGEMENT, TABLE_MANAGEMENT, and UPGRADE_LOG are longer available.
- The job then proceeds to load the FULL data from Maconomy to the DWH.

Upgrade People Planner ETL and Data Warehouse

To upgrade the ETL and People Planner Data Warehouse, install the ETL of the new RPU.

Upgrade Universes and Documents

Upgrading universes and documents is required for upgrading *BPM Reporting*, *BPM Analysis*, and *BPM People Planner*.

Upgrading is done by installing the respective LCMBIARs from the new RPU. Before installing, make a backup of the CMS, for example, a shape of a LCMBIAR export using Promotion Management.

Upgrade Customizations

Customizations are the responsibilities of the consultant who applies them. There is no automatic resolving mechanism that safely upgrades BPM without the risk of overwriting customizations or that automatically makes customized BPM components adjust to upgraded BPM components.

However, there are several precautions that you can take when customizing BPM that help to protect the customizations from being overwritten when upgrading.

The following outline the procedures for upgrading customizations, which include:

- Upgrading custom ETL
- Upgrading custom universes
- Upgrading Custom Documents on Standard Universes
- Upgrading Custom Documents on Custom Universes

Upgrading Custom ETL

Customized ETL is typically create as separate ETL pieces—that is, separate jobs and tasks that are called from the standard ETL. When a new standard ETL is installed, any modifications to it are overwritten, including the places that called the custom ETL. This link to the custom ETL then must be reestablished. The custom ETL itself should have facilities for upgrading to new database schemas of tables, and so on.

Upgrading Custom Universes

Customized universes are typically created in one of two ways:

- **Copied** — A copy of a standard universe. In this case, upgrading is not needed except for running refresh structure on the custom universe to update it to fit the present Maconomy database.
- **Linked** — A universe linking to a standard universe. In this case, there should be no need for doing modifications to the custom universe.

There are exceptions, namely if deprecated elements have been removed from the standard universe or that universe has undergone substantial changes, as happens on occasion.

Also, the custom universe may have introduced tables, objects, and so on, which are also introduced in the new standard universe. Therefore, it is always a good idea to inspect the link from the custom universe to the standard universe and make sure it is intact.

Upgrade Custom Documents on Standard Universes

Custom documents running on standard universes typically work without any changes.

If the principles for customization have been followed, installing a new standard package does not overwrite custom documents.

However, as standard universes from time to time may undergo some changes or fixes, or because deprecated elements may be removed from universes, it is always recommended to test run custom documents and inspect for any issues.

Upgrade Custom Documents on Custom Universes

If customizations have been performed following the recommended principles, upgrading to a new BPM standard package should not ruin any custom documents or custom universes.

It is always a good idea to inspect custom documents, and especially universes, to ensure that they are still functioning correctly after an upgrade; the Maconomy database could have been changed and made custom universe elements invalid or require a refresh of the structure.

SETUP AND CONFIGURATION

Set Up Maconomy

You must set up Maconomy before you can take full advantage of BPM Reporting's features.

Calendar Day

Universes and reports in BPM Reporting use dates that refer to the Calendar Day table in Maconomy. To use this table, you must specify a system parameter that identifies the starting date of the calendar and the number of years for which it should run.

To set up the calendar day:

1. Open the System Parameters workspace.

2. Click **Use Calendar Day Table**.

3. Specify the number of years and the start date in the following format: **MM/DD/YYYY**.
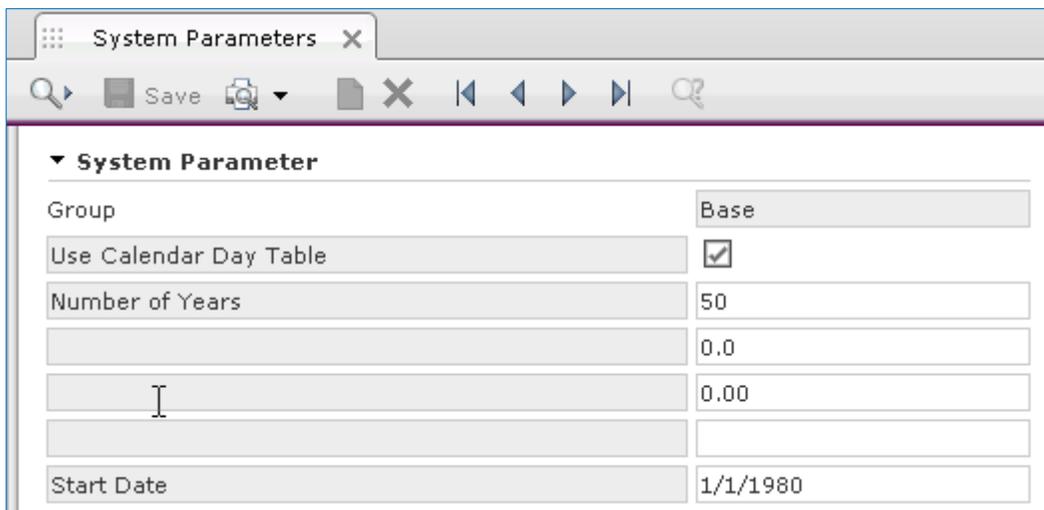
This creates an entry in the CalendarDay table for each date starting with 1/1/1980 and 50 years beyond.

Split Week Time Sheet

You need to set up the split week time sheets to use BPM Reporting. Split week time sheets are mandatory when using monthly time reporting. For more information, see the *Deltek Maconomy Online Help* and *Deltek Maconomy Online Concepts Guide.*

To enable the split week time sheet:

1. Go to **Setup » System Setup » Parameters and Numbers » System Parameters.**

2. Double click the **Split Week Time Sheet** system parameter**.**

3. On the System Parameter sub-tab, select the **Split Week Time Sheet** check box**.**

4. Click **Save.**

5. Restart the Workspace Client.

Reporting Structures

The Finance universe of BPM Reporting and the General Ledger universe of BPM Analysis use reporting structures to structure and filter accounts.

Reporting Structure for Finance Measures

To set up dimension grouping:

1. Open the Option List window.

2. In the **Option List No**. field, enter **Finance.**

3. In the **Name** field, enter **Standard**.

4. Open the Reporting Structures window in the workspace Reporting.

5. Create a reporting structure with the following settings:

    ▪ **Type** — Account

    ▪ **Option List** — Finance

    ▪ **Selected Value** — Standard

6. Insert group headers, sub-group headers to provide the desired account structure.

7. Add accounts to the individual groups of the reporting structure.

Each group states the header labels for the finance report using the reporting structure. For example, account number **1110** is displayed in the group with the top-level header RESULT, then RESULT BEFORE EXTR. ENTRIES, REVENUE, and so on.

> **Note:** If using Local Charts of Accounts, reporting structures should be setup for these as well. The **Type** for these is Local Account.

Reporting Structures for Cashflow

> **Note:** If Maconomy is already set up for BPM Reporting, this reporting structure may already exist. If it does, you do not need to perform the following procedure.

To select the accounts for which postings are considered as contributing to cash flow (cash flow statement):

1. Open the **Option Lists** window.

2. Create an option list named **Finance**.

3. Enter a value with the name **Cashflow Statement**.

   If Business Performance Management Reporting has already been set up in Maconomy, the option list might already exist; if so, add only the value Cashflow Statement.

4. Open the Reporting Structure window in the workspace Reporting.

5. Create a new reporting structure with the following settings:

   - **Type** — Account
   - **Option List** — Finance
   - **Selected Value** — Cashflow Statement

6. Create grouping headers and add accounts similar as done above. Cashflow usually only include a small subset of accounts. Thereby, the reporting structure works as a filter on accounts in addition to providing a hierarchical structure.

7. Create another reporting structure with the option list value **Cashflow Forecasting**.

Reporting Structure for Cashflow Statement Indirect Method

To create a reporting structure for Cashflow Statement Indirect Method:

1. Go to **Setup » System Setup » Parameters and Numbers » Option Lists.**

2. Click **+ New Option List.**

3. In the **Option List No**. field, enter **Finance.**

4. Click **Create.**

5. On the Options sub-tab, click **Add Option** or press **Ctrl + M.**

6. In the **Name** and **Description** fields, enter **Cashflow Statement**.

7. Click **Save Option (Enter),** then click **Save Option List (Enter).**

8. Go to **Setup » Reporting » Reporting Structures.**

9. Click **+ New Reporting Structure.**

10. In the Create Reporting Structure dialog, enter the following setting:

    - **Name** — Cashflow Statement
    - **Description** — Cashflow Statement Reporting Structure
    - **Type** — Account

- **Option List** — Finance
- **Selected Value** — Cashflow Statement

If Business Performance Management Reporting has already been set up in Maconomy, the option list might already exist; if so, add only the value Cashflow Statement and description.

11. Click **Create.**

12. Insert group headers, sub-group headers to provide the desired account structure, similar as done above. Cashflow usually only include a small subset of accounts. Thereby, the reporting structure works as a filter on accounts in addition to providing a hierarchical structure.

    a. Add two top-level groups for **Net Cashflow** and **Cash Reconciliation**.

    b. Under Net Cashflow, add three sub-level groups for **Operating**, **Investing**, and **Financing Activities**.

    c. Then, you can configure the next levels based on your preference.

Reporting Structure for Dashboard Components

The dashboard components and dashboards in BPM Reporting also require the same kind of reporting structure as is needed for the dashboards in BPM Analysis. The following replicates the setup steps. However, it is not necessary to define all dimension categories, only Revenue on level 5 and Gross Profit on level 4. Deltek recommends that you set up a meaningful structure that includes all of the groups.

To classify accounts that are used to calculate finance key metrics:

1. Open the **Option Lists** window.

2. Create an option list with the number Finance and a value with the name Key Metrics.

   If BPM Reporting has already been set up in Maconomy, this option list might already exist. If so, add only the Key Metrics value.

3. Open the Reporting Structure window in the workspace Reporting.

4. Create a reporting structure with the following settings:

   - **Type** — Account
   - **Option List** — Finance
   - **Selected Value** — Key Metrics

5. The reporting structure needs to have specific grouping headers and a certain structure. This is described below:

6. Place each account in the group where it belongs.

Assign	To the Value	For All Accounts in which
Grouping 1	NET PROFIT	Posted amounts should contribute to the calculation of Net Profit.
Grouping 1	ASSET	Posted amounts should contribute to the calculation of Assets.
Grouping 1	LIABILITY	Posted amounts should contribute to the calculation of Liability.

Assign	To the Value	For All Accounts in which
Grouping 1	EQUITY	Posted amounts should contribute to the calculation of Equity.
Grouping 2	EBIT	Posted amounts should contribute to the calculation of EBIT.
Grouping 3	EBITDA	Posted amounts should contribute to the calculation of EBITDA.
Grouping 4	GROSS PROFIT	Posted amounts should contribute to the calculation of Gross Profit.
Grouping 5	REVENUE	Posted amounts should contribute to the calculation of Revenue.
Grouping 5	COST	Posted amounts should contribute to the calculation of Cost.
Grouping 5	DEPRECIATION	Posted amounts should contribute to the calculation of P&L depreciation.
Grouping 5	INTEREST	Posted amounts should contribute to the calculation of P&L interest.
Grouping 5	TAX	Posted amounts should contribute to the calculation of P&L TAX.

Reporting Structure for NSAFT

The NSAFT report uses a reporting structure to group finance data.

To create a reporting structure for NSAFT:

1. Go to **Setup » System Setup » Parameters and Numbers » Option Lists.**

2. Click **+ New Option List.**

3. In the **Option List No**. field, enter **Finance.**

4. Click **Create.**

5. On the Options sub-tab, click **Add Option** or press **Ctrl + M.**

6. In the **Name** and **Description** fields, enter **NSAFT**.

7. Click **Save Option (Enter),** then click **Save Option List (Enter).**

8. Go to **Setup » Reporting » Reporting Structures.**

9. Click **+ New Reporting Structure.**

10. In the Create Reporting Structure dialog, enter the following setting:

    - **Type** — Accounts or Local Accounts

    - **Option List** — Finance

    - **Selected Value** — NSAFT

11. Click **Create.**

12. Insert group headers, sub-group headers to provide the desired account structure.

13. Add accounts to the individual groups of the reporting structure.

Grouping Headers

The grouping headers specified in the reporting structure is mapped to the generated XML file following the structure below:

Tags (attributes indented)	Mapping
<MasterFiles>	
<GeneralLedgerAccounts>	
<Account>	
<AccountID>	[Account No.]
<AccountDescription>	[Account Description]
<StandardAccountID>	[Grouping Level 1]
<GroupingCategory>	[Grouping Level 2]
<GroupingCode>	[Grouping Level 3]
<Account>	
<GeneralLedgerAccounts>	
<MasterFiles>	

Reporting Structure for Statement of Changes in Equity

The Statement of Changes in Equity report uses a reporting structure to group data.

> **Notes:**
> - Only Grouping Level 1 is used for this report. No other grouping levels need to be set for this report as it does not utilize them.
> - The reporting structure does not need to have Option List and Option Value set as the report prompts the user to select a reporting structure instead.

To set up a reporting structure for the Statement of Changes in Equity report:

1. Go to **Setup » Reporting » Reporting Structures.**

2. Click **+ New Reporting Structure.**

3. Add a group for each of the columns to be displayed.

   Example:

   Column 1: "Share Capital"

Column 2: "Statutory Reserve"

Column 3: "Other Reserve"

Column 4: "Retained Earnings"

> **Note:** The naming groups and naming of them can be adjusted to your business needs. However, it is usually standard to have one group named "Retained Earnings" or something similar. This group should include the Year-end-Result Account and all the profit and loss accounts.
>
> Each of the other groups should include the respective Equity accounts for which you want the postings to appear for in the respective columns.

4. Click **Create.**

For more information on the column structure and how it is utilized in the report, see the **Deltek Maconomy BPM Report Description Guide**.

Aging Principles Setup

Beginning with version 2.1 (internally 16.0) you must set up aging principles in Maconomy to be able to run the various aging reports (WIP Aging, AR Aging, AP Aging, and variants). A number of BPM Reporting reports also require this setup, even though they are not—as such—aging reports. This includes the various transaction sub-reports to the aging reports and, in general, job cost, AR, and AP reports that show balances.

> **Note:** The functionality is only used and available in BPM Reporting.

What is an Aging Principle?

An aging principle is a named collection of aging periods. Each aging period marks the start and end of a period. Periods can either be backward- or forward-looking. You use backward-looking periods to display aged figures that have passed the statement date; you use forward-looking periods to display figures that are after the statement date.

Periods can also be open intervals such that it is possible to report on aged figures that are more than 90 days old. For each period you can provide a title that appears as the header for the aging column in the aging reports.

Each aging principle states whether the periods are measured in months or in days. In addition, it states whether or not aging is based on the entry date.

> **Tip:** For information about how to set up aging principles, see the Maconomy Reference Manual or single dialog online help.

Changes to the Standard Reports

From Maconomy version 2.1.5 and 2.2.1, all standard reports that use an aging principle no longer prompt users for an aging principle. Maconomy assumes that certain aging principles have been set up.

Customized Reports

If you are upgrading to Maconomy 2.2, you should note that customized reports typically use the aging objects that were available before Maconomy 2.2. Those aging periods are hard-coded, and the reports prompt users to indicate whether they should use the entry date. The hard-coded aging objects are deprecated as of Maconomy 2.2. They are located in the DEPRECATED universe class and have "DEPRECATED" as a suffix to their titles.

> **Note:** These aging objects are no longer maintained and are not included in the universes after the release of version 2.2. It is therefore important to migrate customers' reports to use aging principles instead.

Mandatory Aging Principle Setup

Version 2.1 of BPM includes a collection of dashboard components that you can combine in different ways to create performance dashboards on company and customer levels. These dashboards are also embedded in the Workspace Client for the solutions.

Some of the components assume a specific kind of aging principle. The following procedure shows you how to create that kind of aging principle.

To set up that aging principle:

1. Open the Option List workspace.

2. In the **Option List No**. field, enter **Aging Principles.**

3. Enter **Dashboards** in the **Name** field.
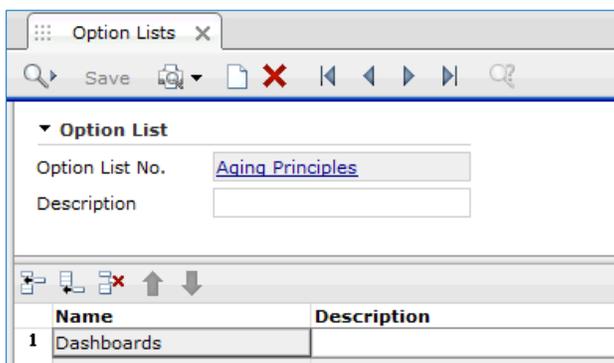
4. Open the Aging Principles workspace.

5. Create an aging principle; use any appropriate name.

6. Enter **Aging Principles** in the **Option List** field.

7. Enter **Dashboards** in the **Selected Value** field.

8. In the table, create four periods. Enter an appropriate title and choose appropriate settings.

Beginning with version 2.1.5, standard reports do not prompt you to select an aging principle. Reports that require an aging principle assume that you have set one up, and that you can indicate it via the option list and selected value, as explained in the preceding steps.

These aging principles assume the same **Aging Principles** option list as described in the preceding procedure:

- **WIP Aging** reports — Designated by having the value **WIP Aging** in the **Selected List** field.

- **AR Aging** reports — Identified by having the value **AR Aging** in the **Selected Value** field.

- **Aging reports** — Identified by having the value AP Aging in the Selected Value field.Utilization-Specific Setup

To take full advantage of the utilization facilities in Business Performance Management Reporting (including the Utilization and Realization reports and the Utilization universe), you must set up activities and employees to comply with the assumptions that are made.

Employee Utilization Pop-Up Literals

Some utilization figures are calculated by using the employee utilization pop-up that is defined on activities. BPM Reporting complies with the PSO setup, and you should set up Pop-Up Literals accordingly.

To set up employee utilization pop-up literals:

1. Open the Popup Fields window.

2. From the drop-down list for **Popup field**, select **Employee Utilization Levels**.

3. Insert the lines that are shown in the following figure.

4. For each activity, assign the appropriate pop-up value for **Employee Utilization**, as shown in the following figure.

If other employee utilization literals are needed, you must insert them in the Popup Fields window (see the figure that follows step 2 in the preceding procedure). You can then customize the objects in the Custom Utilization Figures class to use the literals.

Fixed Hours Setup for Employees

Fixed hours are calculated from the week calendars, from the employee revisions, or as the minimum of fixed hours that are defined on both. Therefore, you must assign a week calendar to each employee and set up fixed hours for both week calendars and employees.

To set up fixed hours for employees:

1. Open the Week Calendars window.

2. For each week calendar to be referenced by employees, ensure that all of the necessary days within the weeks show the appropriate fixed working hours.

3. Open the Employees window.

4. For each employee, assign a week calendar number to the current employee revision.

5. Assign employee-specific fixed working hours.

Derived Dimensions for Employees

In utilization and realization reports, you can drill down on employees' dimensions. This is different from other reports, where you drill down on the dimensions of transactional entries that to which standard dimensions are assigned. For drilling down to be meaningful, you must assign values for all ten standard dimensions for each employee.

To set up derived dimensions for employees:

1. Open the Derived Dimensions window.

2. For each employee that is displayed in the upper pane, assign values to all ten standard dimensions in the lower pane.

Note: Previously, you were required to enable split weeks on time sheets. BPM version 2.1 and newer can handle split weeks being enabled or **not** enabled.

Set Up BPM Reporting Images

To set up BPM Reporting Images, you must copy the files from the images folder of the Reporting Packing Unit (RPU) and paste them in the destination folder for graphical images in SAP BusinessObjects.

To copy and paste the image:

1. Go to images folder of the Reporting Packing Unit (RPU) and copy the image files.

2. Go to the destination folder for graphical images in SAP BusinessObjects, and paste the image files in this folder.

   For example:

   C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0\images

Set Up BPM Reporting Dashboard Components

This setup is required for BPM Reporting Dashboard Components only.

BPM Dashboard Components allows you to view different dashboard components in a window tab.

To properly view the Company Performance, Customer Performance, and Job Performance workspaces in the Workspace Client, copy the **BPMDashboards.war** file from the download server to the local machine.

To copy and prepare the Dashboard Component iFrames, follow these steps:

1. Copy the **BPMDashboards.war** file from the download server and place in an appropriate location on the local machine.

2. Navigate to C:\Business Objects\Tomcat7\conf and update ***tomcat-users.xml***

3. Insert the following lines between the above `</tomcat-users>` **end tag**. This creates the user **admin** that has the role for **manager-gui**:

   a. `<role rolename="manager-gui"/>`

   b. `<user username="admin" password="admin" roles="manage-gui"/>`

4. Navigate to the http://<servername>:8080/ Tomcat interface.

5. Click on the **manager webapp** link and use the credentials that you created for **admin**.

6. In the **Deploy directory or WAR file located on the server** section, deploy the BPMDashboards.war and complete the following:

   - Context Path (required) — (Starts with a backslash (/) and is concatenated with the war file name, for example, /BPMDashboards).

   - WAR or Directory URL — (Complete URL of the WAR file to be deployed, including the name of the war file, for example, C:\temp\BPMDashboards.war).

7. Click **Deploy**.

8. Once the webapp is deployed, check whether the web app is started. If not, start the web app.

CUSTOMIZATIONS

Workflow

This process of customization is divided into four main sections, described in the following.

1	• **Customizing ETL**
2	• **Customizing Universes**
3	• **Customizing Report Documents**
4	• **Customizing BPM Dashboards**
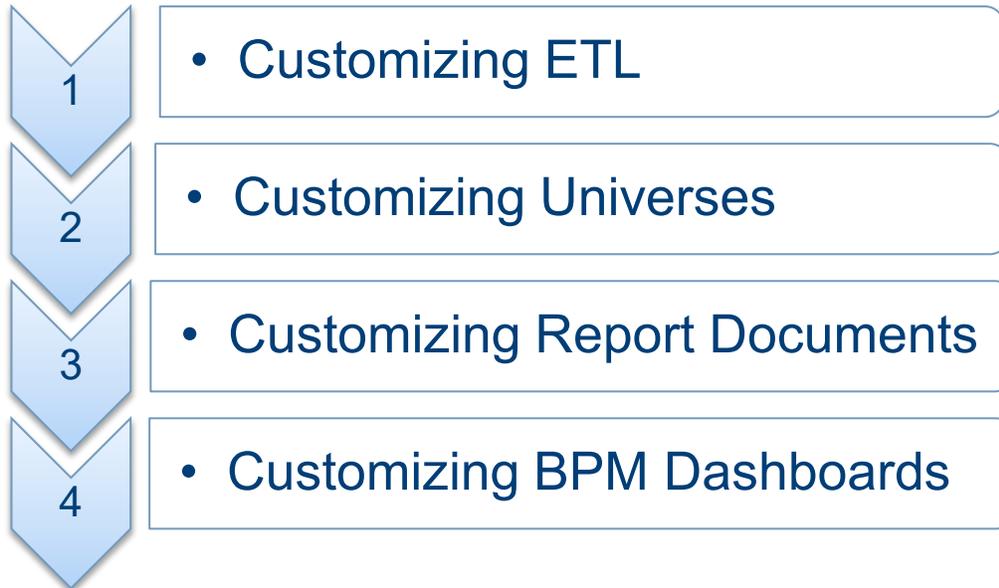
All parts of BPM can in principle be customized in the sense that ETL, Universes, and Report Documents can all be modified to fit the needs of customers.

However, such modifications may be overwritten when upgrading to a new version or service pack of BPM. Therefore it is recommended as a general rule that all customizations are done on copies of the instances. In the following we shall explain further what that means.

Customizing ETL

General Rules

- All customizations to ETL (that is, jobs or transformations) should be done by new jobs and transformations dedicated to handle a certain customization.

- These custom jobs and transformations must be placed in a folder called "custom" or "Custom" in the repository.

- No standard jobs or transformations should be changed.

- The folder structure (aside from that in the "custom" folder) should remain unchanged.

- New jobs/transformations for loading custom-defined data should be scheduled or manually run after the standard LoadData or a custom version of LoadData can be made.

- When upgrading, customizations may need to be reconsidered:

  - An upgrade may introduce standard functionality that previously has been implemented by customizations. It is recommended to include the standard functionality in favor of the customizations, if they fully satisfy the needs of the customer.

- An upgrade may change field names or semantics of fields. Consequently, customizations need to be adjusted carefully.

- An upgrade may declare fields or tables as deprecated (for example, if they are replaced by others or not included in Maconomy anymore). Customizations that rely on such fields and tables should be considered changed. In some situations it may be convenient for the customer to change the customizations such that they still rely on old functionality, for example, deprecated fields. Deprecated fields or tables that are removed by the upgrading process can then be reintroduced in custom tables. Note that a dump of the content of the deprecated table then must be done as part of the upgrade process.

- Carefully document the steps performed in the customization such that these can easily be reconsidered and possibly redone after an upgrade. Primarily, document the following:

  - Which standard tables the custom tables are copies of.

  - Which fields have been added to the copies of the standard tables.

  - The links that have been made.

Transaction Time Stamps for Customized ETL

If the data warehouse ETL has been customized, additional tables may need to have transaction time stamps enabled. For example, if additional tables are maintained by the customized ETL, and incremental load is defined for these tables, the corresponding tables in Maconomy must have transaction time stamp triggers enabled.

Check **Enable Transaction Time Stamps** in MConfig for the database short name. This action enables transaction time stamp triggers for all Maconomy tables, and thus also the tables used in the ETL customization.
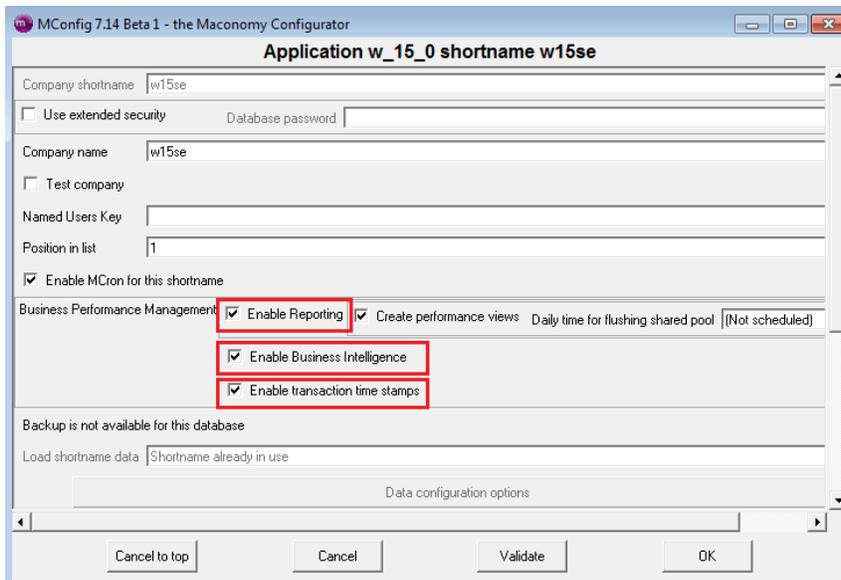
Customizing Universes

To ensure that custom universes are not ruined when installing a new BPM package, it is best practice to always introduce a custom universe as a copy of a standard universe or a whole new universe.

To customize a standard universe, complete the following steps:

1. In Information Design Tool, retrieve the desired universe to universe files (see Retrieve the Universe Files).

2. Choose another project or folder than the one used for the universe files of the standard universes.

3. The universe files can be edited freely to fit the customer's needs.

4. Publish the universe. You can choose whether to overwrite the existing universe or publish it with another name or to another folder. If you overwrite the existing (Standard) universe, make sure that all documents that use that universe (for example, Standard documents) indeed should run on the custom universe; otherwise, choose another name or folder for the custom universe.

When upgrading, custom universes that are stored with another name or in another folder are not overwritten. However, it is still recommended that you keep the universe files in case you need to publish the custom universes again.

Customizing Report Documents

This section describes how to complete report documents and universe customizations.

You can create the following customizations:

- Customize a single report
- Customize multiple reports
- Customize document styles

Customizing a Single Report

To customize a single standard report, complete the following steps:

1. In Web Intelligence, locate the standard report to customize (in this example, AR Aging).
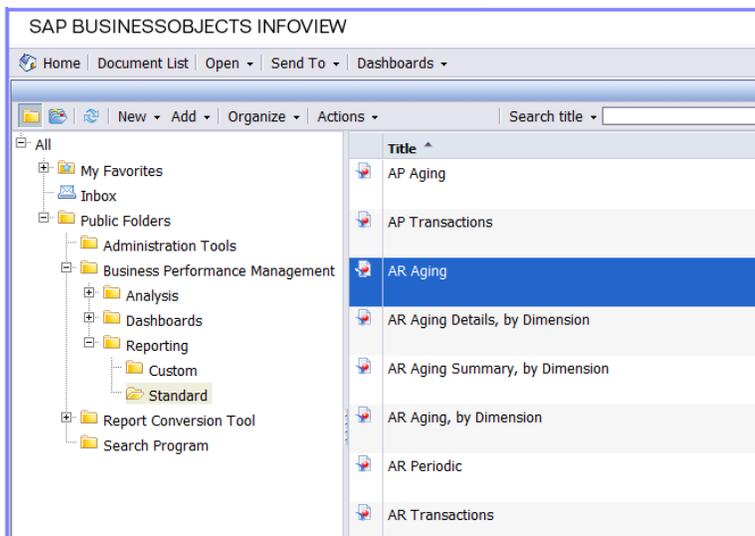
2. Open the selected standard report and click **Document » Save As**, then select a custom folder. Click **OK**.

> **Note:** Choose the **Refresh on open** option on the Save As dialog to refresh the report each time that you open it.

3. Make changes to the report and save the document.

Customizing Multiple Reports

To copy all or selected standard reports to a custom location, complete the following steps:

1. In the Central Management Console, click **Business Performance Management » Reporting » Standard** to open the standard reports.
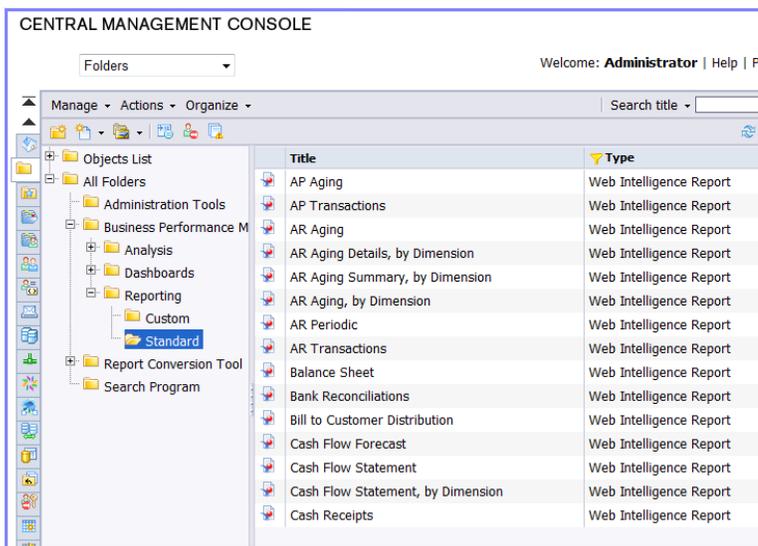
2. Select the report documents that you need, then right-click and select **Organize » Copy To**.

3. On the Copy To screen, select a Custom folder and click **Copy**.

    The selected standard reports are now available as copies in the Custom folder.

Customizing Document Styles

The standard report documents are prepared for customization concerning document style. Most of the layout elements use a default setting, which means that their style is taken from a cascading style sheet. An example is that font family, color, and border style in data tables follow the settings of the cascading style sheet.

The cascading style sheet is built into each report document and is not read dynamically.

If you want to use another style, you can import a new style sheet into each document. For convenience the cascading style sheet of the standard report documents is included in the RPU. This cascading style sheet is a good starting point for creating a custom one.

You can do the following with custom cascading style sheets:

-   Modify
-   Apply the style
-   Export

Modify the Style

To modify a cascading style sheet, follow these steps:

1. Copy the cascading style sheet from the RPU to the local machine from which WebIntelligence is run.

2. Open the style sheet in a text editor.

3. Modify the style sheet to the style that you want.

4. Save the style sheet.

   Now you can apply the style sheet to your documents.

Apply the Style

To apply the style of a custom cascading style sheet, follow these steps:

1. Open the WebIntelligence document to be modified.

2. Click **Properties**.

3. Click **Document**.

4. In the window that appears, click **Change Default Style...**

5. Click **Import Style**.

6. Browse to and select the modified style sheet file. Click **Close**, **OK**, and **Save**.

   The styles of the cascading style sheet are now reflected in the document.

Export the Style

To export the cascading style sheet of a document, follow these steps:

1. Open the WebIntelligence document that contains the style sheet that you want to export.

2. Click **Properties**.

3. Click **Document**.

4. In the window that appears, click **Change Default Style...**

5. Click **Export Style**.

6. Browse to the location of the style sheet file that you want to export.

7. Click **Save**.

   A cascading style sheet file that reflects the styles of the document is saved to the file system. You can use this as a foundation for customizing the styles, similar to using the cascading style sheet delivered in the RPU.

Customizing BPM Dashboards

You can do the following to customize BPM dashboards:

- Edit the JSP files

Editing the JSP Files

Each of the three BPM Dashboards has a JSP page that contains four iFrames that reference dashboard components (WebI Documents) in InfoViewApp.
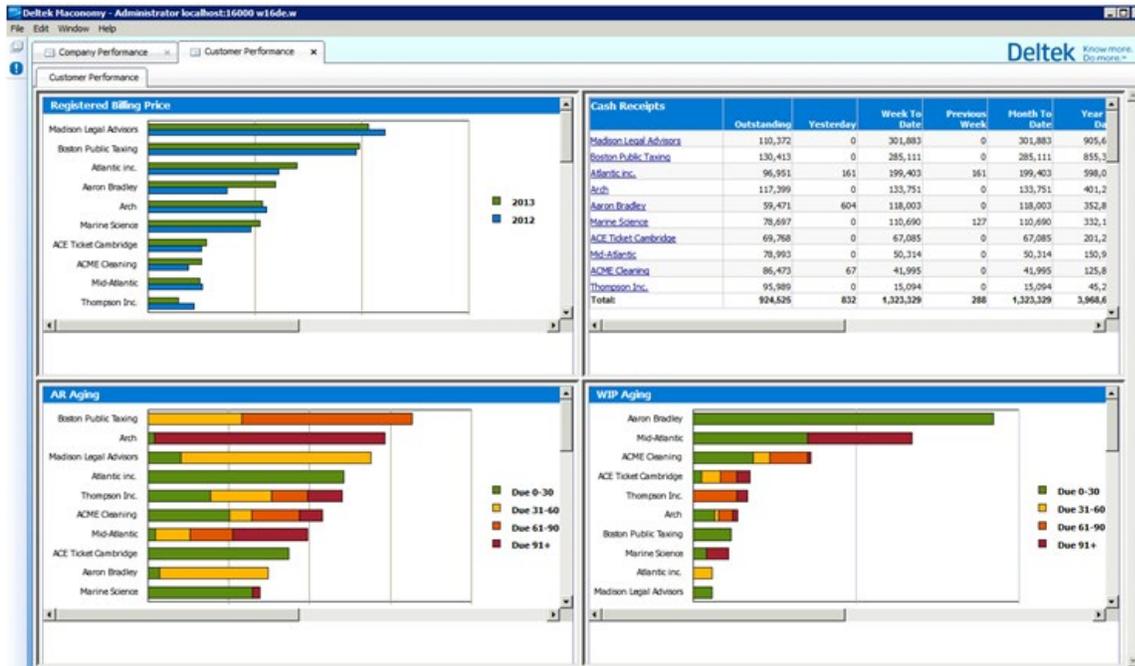
The three JSP pages that are found in the RPU are as follows:

For Standard Dashboards:

/dashboards/reporting/

- CompanyPerformance.jsp
- CustomerPerformance.jsp
- JobPerformance.jsp

For CPA Dashboards:

/dashboards/reporting/cpa/

- CustomerPerformance.jsp
- JobPerformance.jsp

> **Note:** CompanyPerformance.jsp is also used in CPA solution dashboards as the document names used are the same in Standard.

You can see the layout of the iFrames in this part of the code in each JSP:

```
out.println("<table>");

out.println("<tr>");

out.println("<td>");

out.println("<iframe src='" + ProductionRevenue + "' height='100%' width='100%'>"
+"</iframe>");

out.println("</td>");

out.println("<td>");

out.println("<iframe src='" + CashReceipts + "' height='100%' width='100%'>"
+"</iframe>");

out.println("</td>");

out.println("</tr>");

out.println("<tr>");

out.println("<td>");

out.println("<iframe src='" + ArAging + "' height='100%' width='100%'>" +"</iframe>");

out.println("</td>");

out.println("<td>");

out.println("<iframe src='" + WIPAging + "' height='100%' width='100%'>"
+"</iframe>");

out.println("</td>");

out.println("</tr>");

out.println("</table>");
```

The **src** attribute contains the path to the dashboard component.

The following figure shows a sample variable that contains the source path to the dashboard components.

```
//OpenDocument URLs for each of the Job Dashboards Components


String ProductionRevenue="http://" + cmsName +
":8080/OpenDocument/opendoc/openDocument.jsp?sType=wid&sPath=[Busines
s+Performance+Management],[Reporting],[Dashboard+Components],[Job]&sD
ocName=Registered+Billing+Price,+Top+10+Jobs+%28Bar+Chart%29&token="
+ wscToken + …
```

The **sPath** parameter indicates the location of the report/dashboard component.

The **sDocName** indicates the name of the report. Note that the name of the report must be properly URL-encoded with spaces converted to "**+**" instead of "**%20**".

You can also pass prompt values to the dashboard components via the URL by using the URL-encoded name of the prompt. Thus, if you want to pass a value to the prompt "Account Manager No:" add the following part to the URL:

```
&lsSAccount+Manager+No%3A=" +  accountMgrVal
```

> **Note:** The name must be preceded by **lsS**.

An example of how to change the dashboard is shown in the following:

1. Open the file to be changed.

2. Scroll down to the section where the four dashboard components are referenced.

3. Create a copy of the line to be changed and comment one out.

4. Change the name of the referenced component.

The following shows the result.
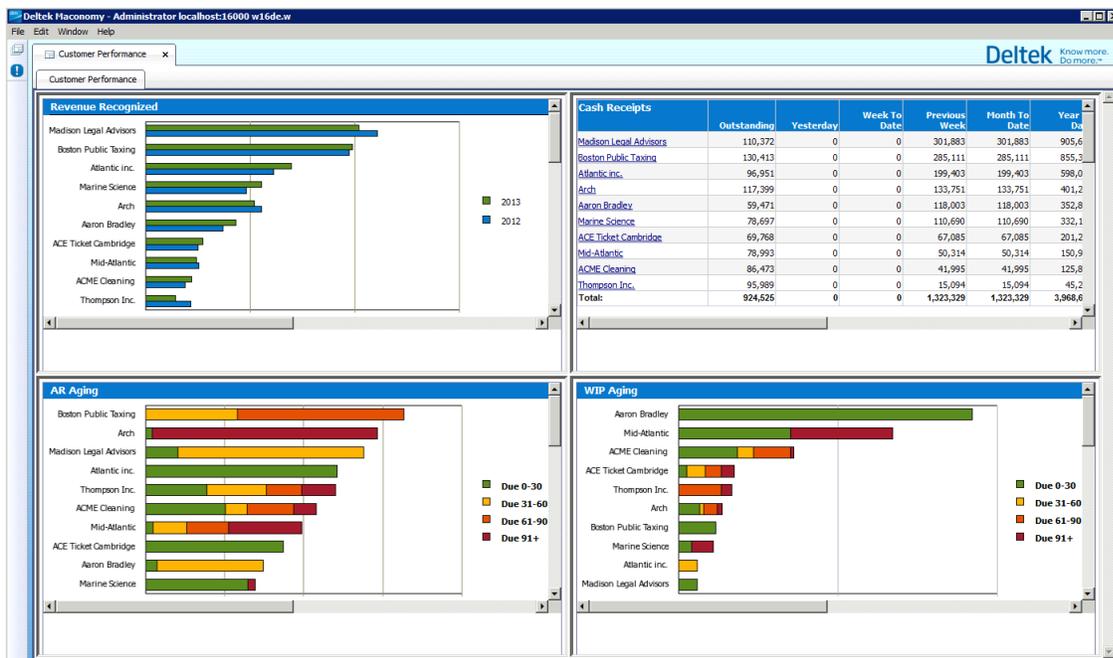
Localization

In BPM, you perform tasks for localization using the BusinessObjects Translation Manager tool, which is part of BOE. The Translation Manager uses the standard XLIFF XML-schema that defines the word or term translation between all standard languages. You cannot create languages, but you can maintain several versions that represent different dialects. For example, for the language "English," there are different kinds (dialects) such as US English and UK English.

You can localize both universes and reports.

This section includes the following material and procedures:

- BPM translator tool overview
- Perform a translation

BPM Translator Tool Overview

Maconomy uses simple text files that are called dictionaries that contain word-to-word translations. These dictionaries are translated to XLIFF format using BPM's Translator tool.

The Translator tool takes two input parameters:

- A Maconomy dictionary
- Initial XLIFF file

The Maconomy dictionary is a text file that contains word combinations (terms) of items such as universe object names and descriptions. The dictionary associates these terms with corresponding terms in another language. The terms in the two languages are tab-separated. The Translator tool recognizes space symbols as part of the terms so that, for example, "good night" can be translated into "buenas noches."

The Translator tool takes the Maconomy dictionary and an initial XLIFF-file and generates another XLIFF-file by substituting dictionary terms that occur before the tab symbol with dictionary terms that occur after the tab symbol. The initial XLIFF-file is generated from the universe and reports and thus contains all of the terms that are necessary for the translation.

Perform a Translation

This section describes how to perform a translation. To perform a translation, you must:

- Generate the initial XLIFF file
- Modify the initial XLIFF file
- Perform the translation

Generate the Initial XLIF File

To generate the initial XLIF file, follow these steps:

1. Run the BusinessObjects Translation Manager.
2. Click **File** » **Import from CMS**.
3. Browse to and select the universe to be localized, and then click **Add**.

4. Click **Import**.

    The following example shows selections for steps 3 and 4.
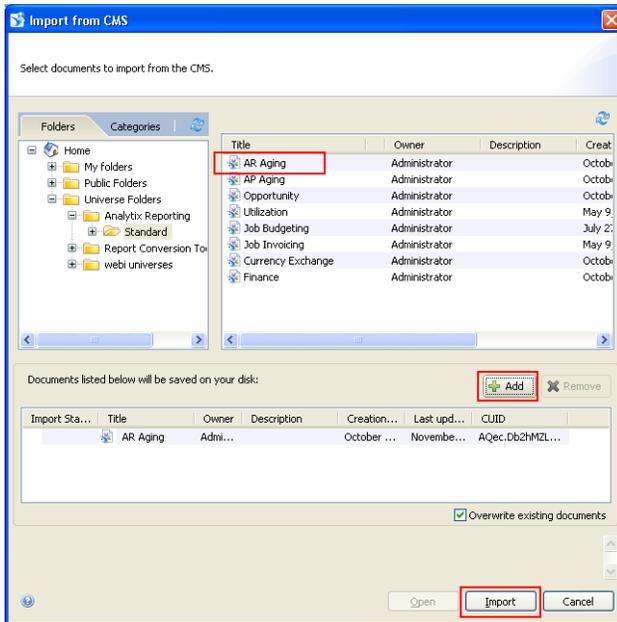
5. Click **Add.**

6. On the Language Management tab, select the **From** and **To** languages.

    The original language is already selected. You must select two additional languages.

    - **Original language** — Refers to the language of the universe (Maconomy W language). This is selected from the start.

    - **From-language** — The language to which the Maconomy W language refers in XLIF. This is usually English (United Kingdom), and terms are provided in UK English by default.

    - **To-language** — The language into which you want to translate, such as Spanish.
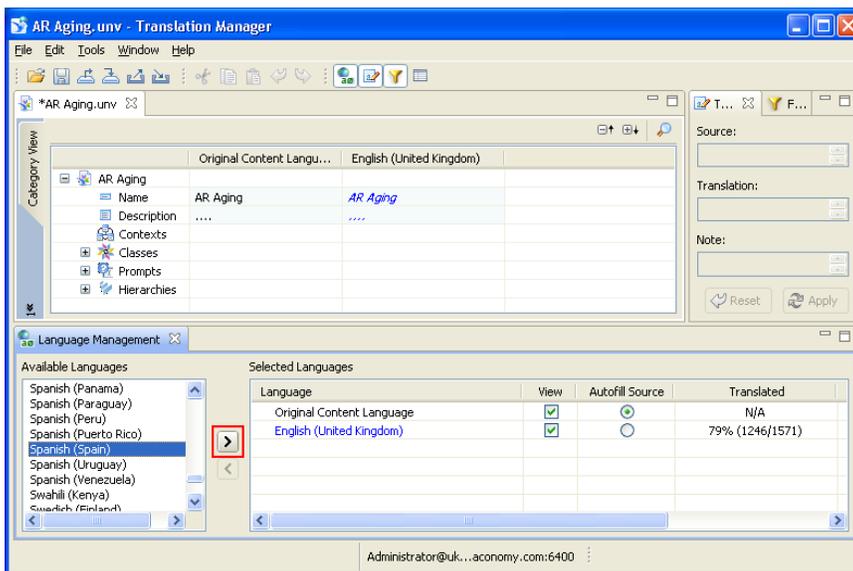
After you have designated the languages, the following is displayed.
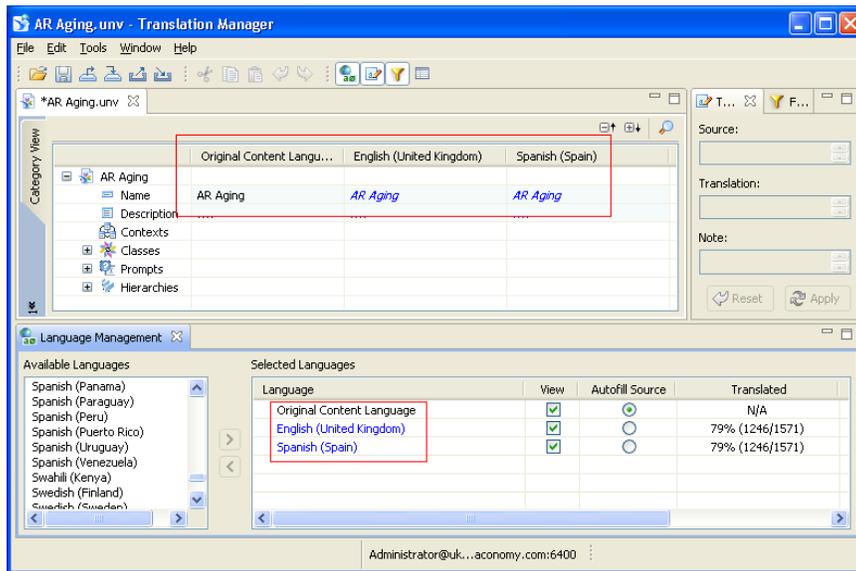
7. Click **File** » **Export to XLIFF**.
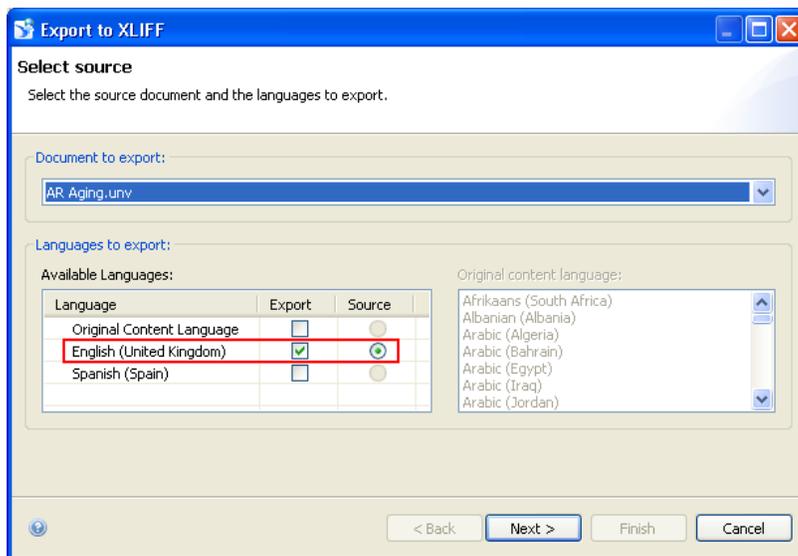
8. Select **Source** to designate one of the languages as the language in which the universe is provided. In the preceding example, it is English (United Kingdom). The source should be the same as the language from the first **Dictionary-file** column.

9. Click **Next**.

10. From the **Available Languages** list, select the **From** language for the XLIFF file.
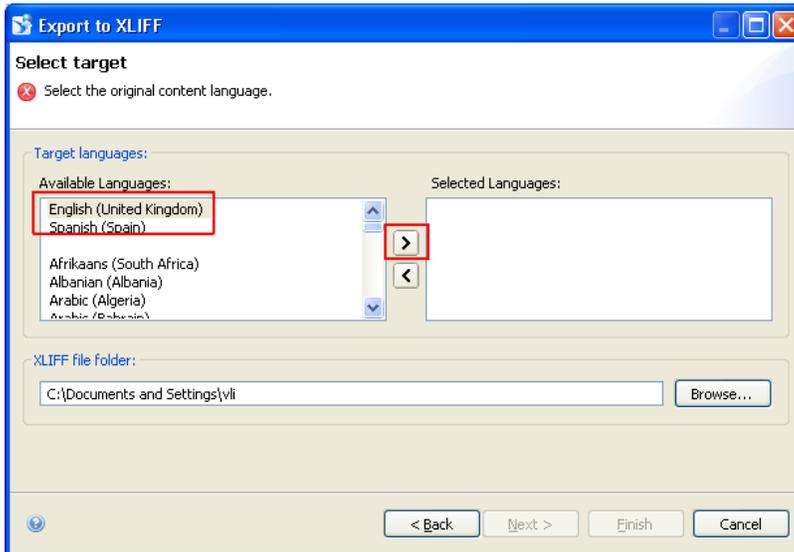
11. Click the arrow to move the languages to the **Selected Languages** list.

These are the **To** languages for translation.
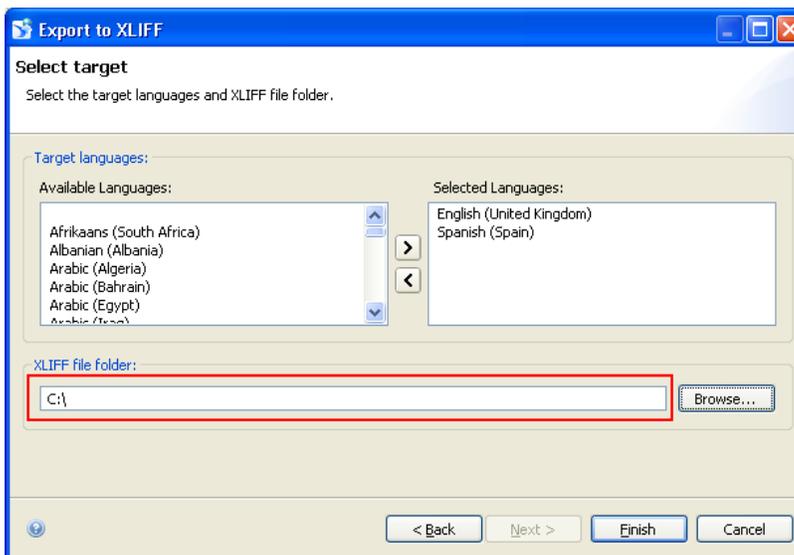
12. Click **Finish**.

Two files are available in the XLIFF file folder:

- AR Aging.unv_en_GB_en_GB.xlf

- AR Aging.unv_en_GB_es_ES.xlf

The file that has different languages in its name should be the initial XLIFF file that is used as one of the tool's input files.

Modify the Initial XLIFF File

The initial XLIFF file contains all of the terms of the universe; however, you must incorporate the translation principles of the Maconomy dictionary. Therefore, you should apply this tool to modify the initial XLIFF file before continuing.

> **Tip:** Refer to the **readme.txt** file that is provided with the tool for translating the Maconomy dictionary file into XLiff for information about how to run that tool.

Perform the Translation

To perform the translation, follow these steps:

1. Click **File » Import from CMS** and import the universe to the Translation Manager.
2. Click **File » Import from XLIFF** and import the output file created in the previous steps.
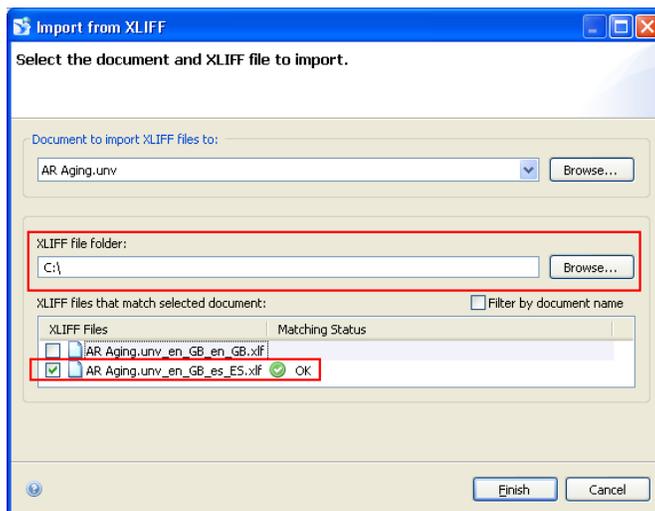
3. After you import the XLIFF file, select the second language in the list as **Ready for use**.
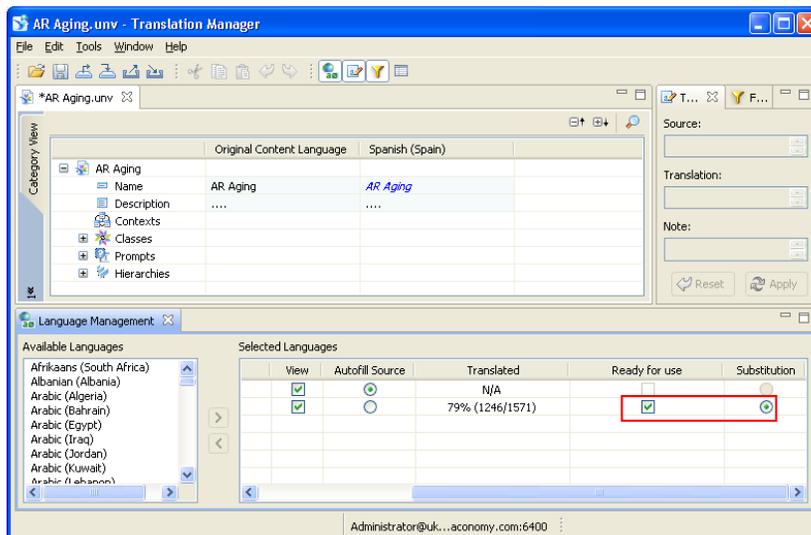
4. Export the universe to the CMS and run WebI to validate that the translation has been applied.

> **Tip:** It is recommended that you export a copy to another location, so that future upgrades of the standard delivery do not overwrite the changes.

SYSTEM MAINTENANCE and REGULAR USE

Data Warehouse Maintenance

Data warehouse maintenance includes:

- Creating ETL jobs
- Scheduling loads
- Processing manual loads

Top-Level Jobs for Maintaining the Data Warehouse

The following are top-level jobs for maintaining the data warehouse. The first two jobs are for loading data. The third is for upgrading. Once a data warehouse is successfully created, you want to load data regularly, such as once a day or once a week, using Incremental Load. Once a month or so, run Full Load to get any missing dependencies corrected. Run Upgrade when you install service packs or new versions.

Use the following jobs for maintaining the data warehouse once it has been created:

Full Load	This job resets the fact tables and loads all data in again.
	Over time the data in incrementally loaded fact tables may become out of sync with some data in their dimension tables for certain scenarios. For example, a record in a dimension table may change without resulting in corresponding fact records to be adjusted.
	Therefore, it is recommended to do a full load, for example, once a month.
Incremental Load	This job loads data that have changed, been inserted or deleted, such as once the data warehouse has been created, new or modified data are retrieved by running this job.
	It is recommended to schedule this job to run regularly.
Upgrade	This job upgrades the data warehouse.

Scheduling Loads

For periodic updates of the BPM DWH, you must use incremental loads. The Kitchen tool, which is part of Pentaho Data Integration, allows you to launch jobs, such as **Incremental Load**, from the command line.

> **Tip:** For more information, see documentation about Pentaho Data Integrator.

The call may be displayed as follows:

```
Kitchen.bat /rep: Pentaho repository /job: "Incremental Load" /dir:/ /user:admin
/pass:admin /level:Basic
```

> **Note:** Periodic updates are accomplished by writing a batch script that launches Kitchen and then schedules the launch of the script.

Start a text editor, such as Notepad, paste the following text, and then save it as updateDWH.bat.

```
@echo off
rem remove '-' from the date
for /F tokens=1-4 delims=-  %%i in ('date/t') do set filedate=%%i%%j%%k
rem remove ':' from the date
for /F tokens=1-4 delims=:  %%i in ('time/t') do set filetime=%%i%%j%%k
rem set the log filename with date and time
set logfilename=C:\Kettle-2[1].3.0\MyLogFiles\Log_%filedate%_%filetime%.log
call Kitchen.bat /rep: Pentaho repository /job: "Incremental Load" /dir:/
/user:admin /pass:admin /level:Basic > %logfilename%
echo Load finished. See the logfile %logfilename%
```

To schedule loading of data through a Scheduled Task, follow these steps:

1. Create a folder named **MyLogFiles** in the Kettle folder.

2. Open a command prompt and make the Kettle folder the current directory.

3. To test the script, write **updateDWH.bat**. This may take some time, since the script loads the data warehouse.

4. Open the control panel and then open **Scheduled Tasks**.

5. Click **Add Scheduled Task**, browse to select the script **updateDWH.bat**, and specify how often to perform the task (for example, daily or weekly).

6. Go through the rest of the wizard. You must enter the password of the user for whom the task is scheduled.

Process Manual Loads

To process manual updates of data warehouse, follow these steps:

1. On the repository, start Spoon.

2. Log in as **admin/admin**.

3. Open the **Incremental Load** job.

4. Run the job (**F9**).

> **Note:** The tab **Log view** should display **Success** in the result column. If you cannot open the repository, it may be because a firewall is blocking the access. Deactivate the firewall or consult your network administrator.

Other Setup

In this section...

This section includes these additional procedures:

- Install the Oracle Instant Client Drivers (Windows)
- Set Up an ODBC Connection for SQL Server
- Enable Unicode
- Increase the Number of WebI Connections
- Configure Characterset Settings if Using Oracle
- User Setup and Single Sign-On
- Configure Logon Token on BusinessObjects
- Localization
- Handling BusinessObjects Services
- Use a Non-Standard Web Server Port
- Set Templates for Web Intelligence
- Burst Publishing
- Materialized Views on Oracle

Install the Oracle Instant Client Drivers (Windows)

If your Maconomy system uses Oracle DBMS, and the BPM server is different from the one that has the Maconomy database, you must install Oracle drivers to allow BusinessObjects access to the database and to prevent the following warning.

To install the Oracle Instant Client Drivers, follow these steps:

1. Navigate to: http://www.oracle.com/technology/software/tech/oci/instantclient/index.html.
2. Select your operating system.
3. Review and accept the license agreement and click **Next**.
4. Download either **Instant Client Package-Basic** or **Instant Client Package-Basic Lite**.

5. Unpack the drivers to a folder (for example, C:\Program Files\Oracle).
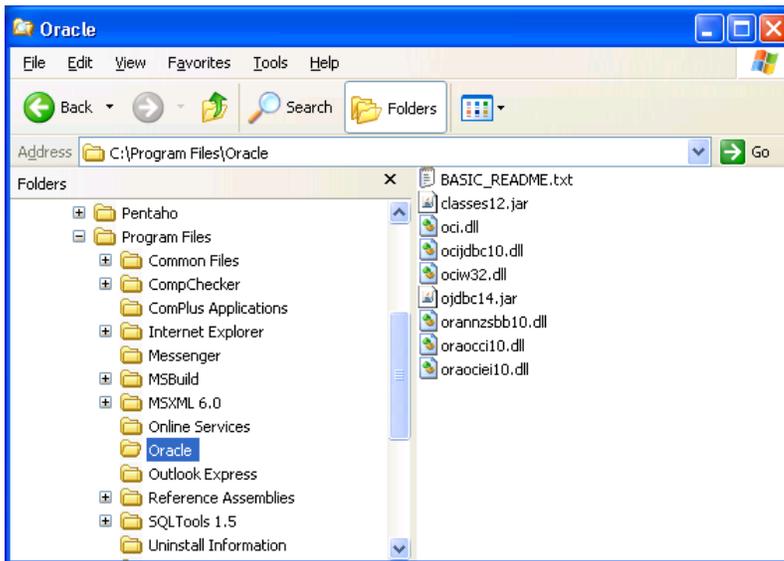
6. Add the path to the drivers to the PATH variable.
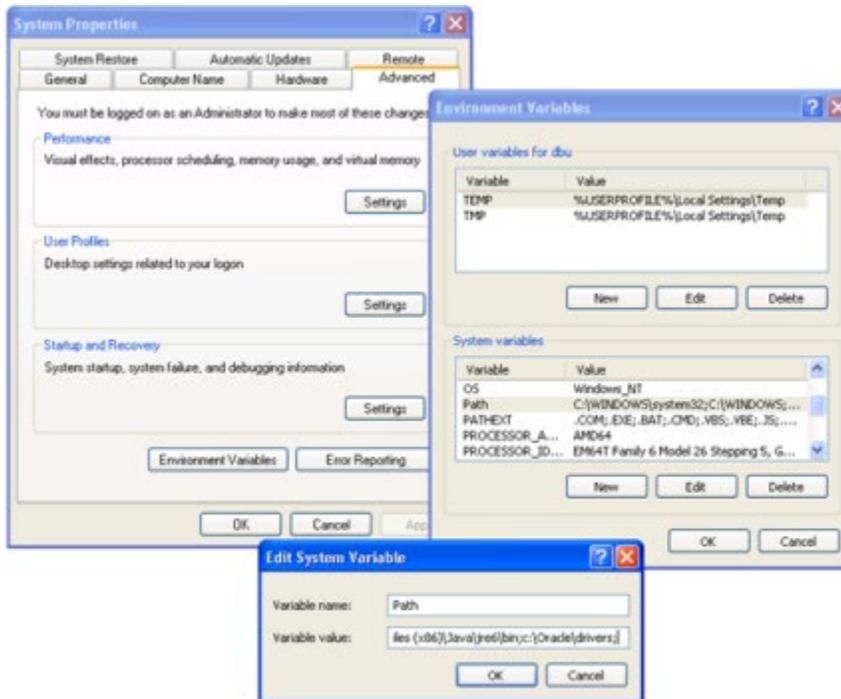
7. Create the variable **TNS_ADMIN** with the path to the same folder.

8. Create **tnsnames.ora** in the folder, for example:

```
maconomy =
    (DESCRIPTION =
        (ADDRESS_LIST =
            (ADDRESS = (PROTOCOL = TCP)(HOST = MACONOMYSERVER)(PORT = 1521))
```

```
        )
        (CONNECT_DATA =
            (SERVICE_NAME = maconomy)
        )
    )
```

In this example, MACONOMYSERVER is the server that hosts the Maconomy database, and maconomy is the SID of that database.

Set Up an ODBC Connection for SQL Server

To set up the ODBC connection, follow these steps:

1. Run Data Sources (ODBC) using the appropriate path:

    ▪ For 32-bit Windows — Click **Control Panel » Administrative Tools » Data Sources (ODBC)**.

    ▪ For 64-bit Windows — Click **All Programs » BusinessObjects XI 3.1 » BusinessObjects Enterprise » 32-bit Data Source (ODBC)**.

2. Click **Add**.
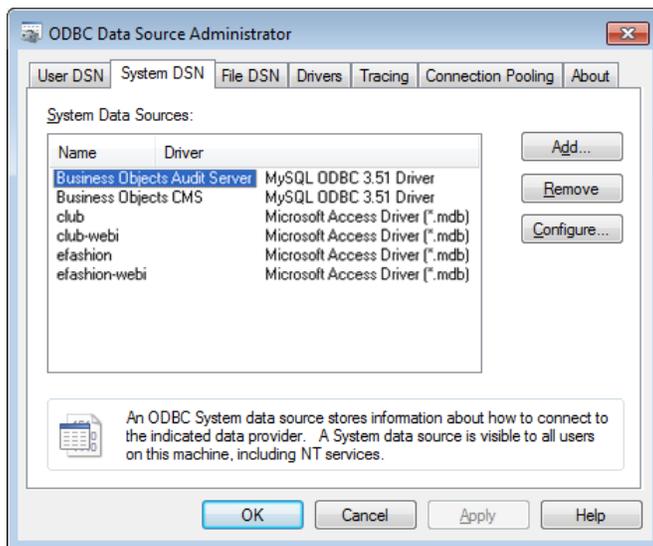
3. Select **SQL Server** and click **Finish**.
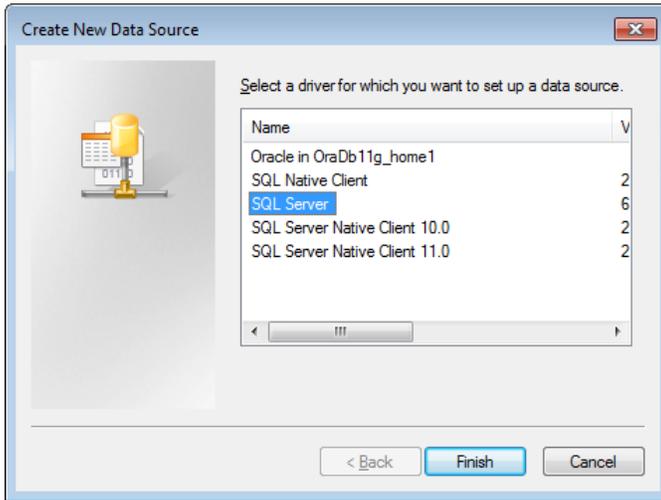
The Create a New Data Source to SQL Server wizard is displayed.

4. Enter the **Name**, **Description**, and **Server** information, and click **Next**.
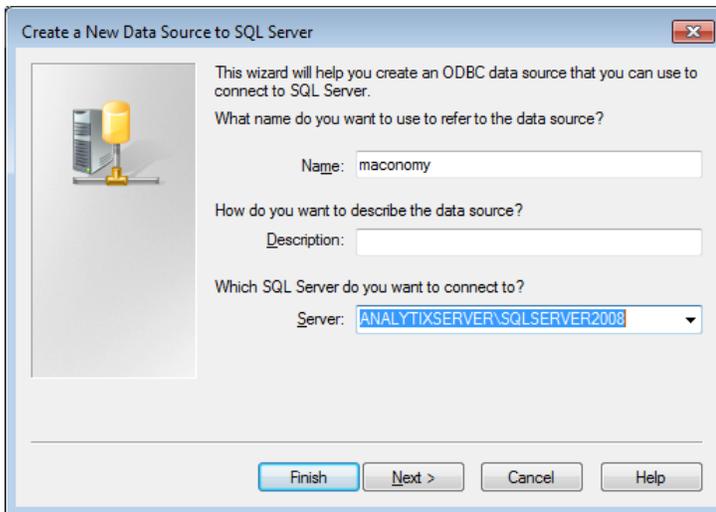
5. Select the method to use for authentication and click **Next**.
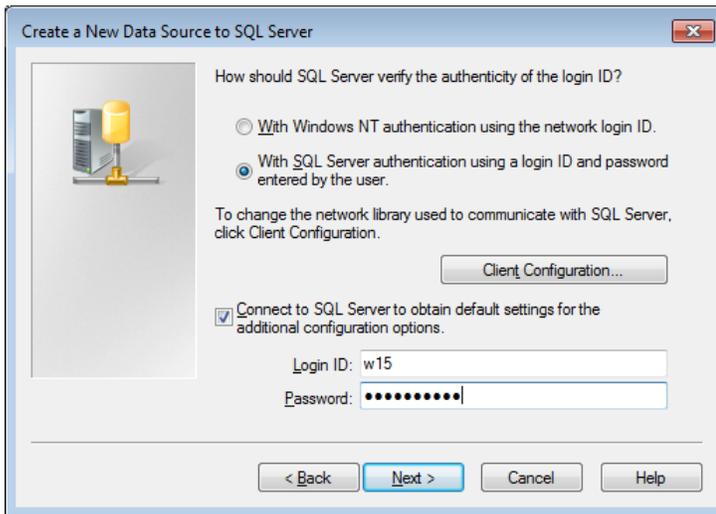
6. Enter the default database and ANSI information and click **Next**.

The following dialog box is displayed.
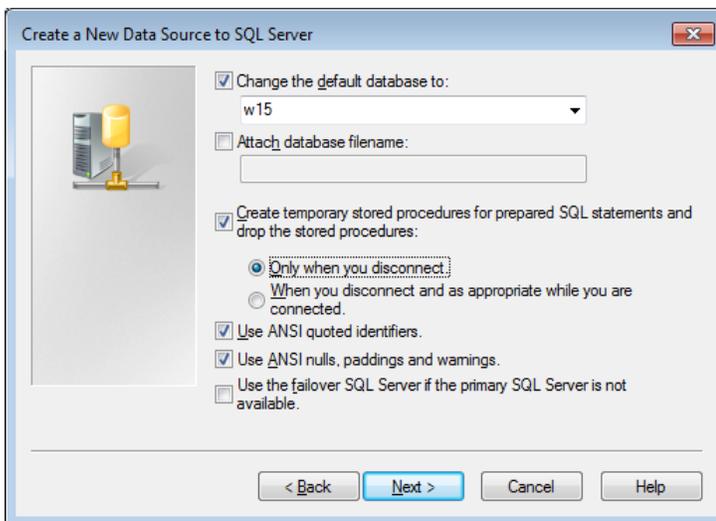
7. Click **Finish** to complete the process.
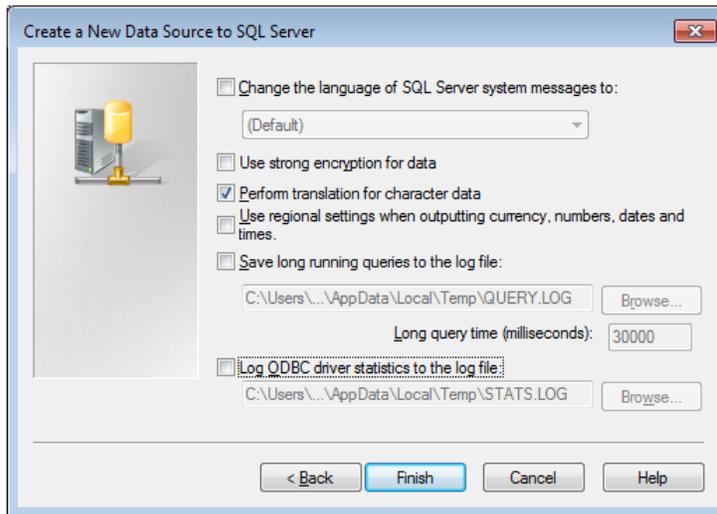
Enable Unicode

Beginning with version 16.0 (2.1), the Maconomy database and BPM must be Unicode-enabled. You do this as part of the installation or upgrade.

To set up BPM for supporting Unicode, follow these steps:

1. After you enable the Maconomy database for Unicode, follow the same steps for enabling the BPM Datawarehouse database for Unicode. See the documentation for information about how to enable Unicode on the Maconomy database.

2. If you are upgrading BPM Analysis, perform the upgrade steps as described in the Upgrade section.

3. If this is a new installation, perform the installation steps that are described in the BPM Analysis Installation Guide. Note that the database for the data warehouse must be Unicode-enabled.

4. If you have BusinessObjects installed on Windows, perform the following steps:

   a. In the Windows registry set the following key:

      HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb11g_home1\NLS_LANG

      to

      AMERICAN_AMERICA.AL32UTF8

   b. Create a system variable NLS_LANG and assign it the value:

      AMERICAN_AMERICA.AL32UTF8

5. If you have BusinessObjects installed on Linux or AIX, perform the following steps:

   a. Set up an environment variable NLS_LANG with the value AMERICAN_AMERICA.AL32UTF8.

   b. Restart the BusinessObjects services.

Ensure that universes have the parameter UNICODE_STRINGS set to YES (this is the default value).

Increase the Number of WebI Connections

In the solutions, the Workspace Client embeds various BPM Reporting reports. To support this, you must increase the default number of parallel connections.

To increase the number of parallel connections, follow these steps:

1. Log in to BusinessObjects Enterprise Central Management Console (CMC).

2. Click **Servers**.

3. Expand **Server Categories** and click **Web Intelligence**.

4. Under **Server name**, double-click **WebIntelligenceProcessingServer**.

   The **Properties** window appears.

5. In the **Properties** window, under **Web Intelligence Processing Service**, enter the appropriate number of parallel connections.

   The default value is 50. It is recommended that you increase the value to at least 100.

Alternatively, add more servers. Adding one more server with the default of 50 parallel connections yields the same number of connections (100) as the preceding steps.

On large production systems, it is recommended that you increase the number of servers and then lower the number of parallel connections on each of those servers. However, note that each user who runs the Workspace Client occupies a number of WebI connections.

> **Note:** It is recommended that you **not** have more than one Web Intelligence Report server per available CPU core.

Configure Characterset Settings if Using Oracle

BusinessObjects needs to know the characterset of Oracle to correctly display characters. You should set the environment variable NLS_LANG on the BO server.

To configure characterset settings, follow these steps:

1. Determine the NLS_LANGUAGE and NLS_CHARACTERSET using the following statement:

   ```
   SELECT * FROM NLS_DATABASE_PARAMETERS WHERE PARAMETER IN
   ('NLS_LANGUAGE','NLS_CHARACTERSET')
   ```

   > **Tip:** For the default setup, the result is AMERICAN, WE8ISO8859P1.

2. Set the environment variable NLS_LANG to the concatenation of these two strings separated by a dot, for example, AMERICAN.WE8ISO8859P1.
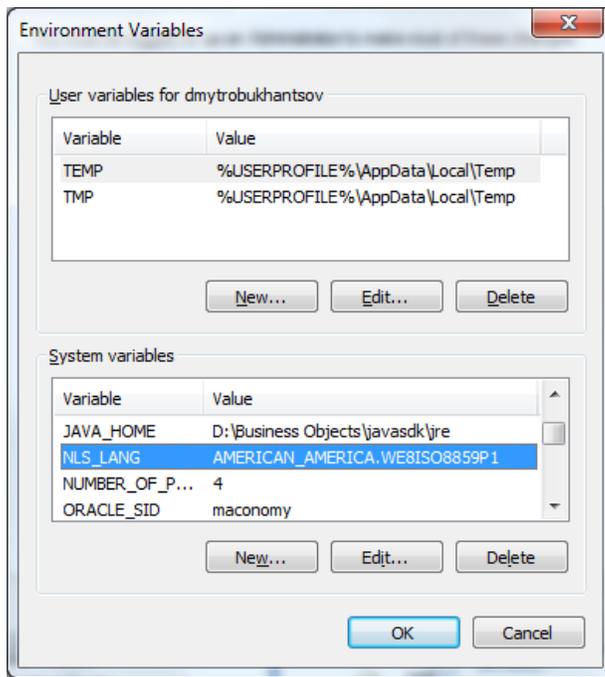
User Setup and Single Sign-On

This section contains the following topics:

- User setup and single sign-on overview
- Active desktop and Maconomy configurations
- USync
- Configure a tenant in a multitenancy environment
- Control which users to create
- Error handling
- Assign report and universe rights

In Maconomy, access to data is controlled by the user name (**NameOfUser**). If single sign-on (SSO) is not defined and used, a user must log in to BusinessObjects using the same user name to be able to access data.

However, most often SSO is defined both in Maconomy and BusinessObjects.

- In Maconomy, a network user name is defined for each user. This network user name is equal to the active directory (AD) (Windows) user name. When you log in to the network, log-in is automatically performed in Maconomy using the network user name. Access to data is controlled by the user name with which the network user name is associated.

- In BusinessObjects, the AD (Windows) user name is the normal user name and thus equal to the network user name. To have access to Maconomy data, you create an alias for that user name. This alias is equal to the user name (**NameOfUser**) in Maconomy. When you log in to the network, log-in is also automatically performed in BusinessObjects using the user name. Access to the Maconomy data is via the alias that matches the user name (**NameOfUser**) in Maconomy.

The following figure illustrates this.
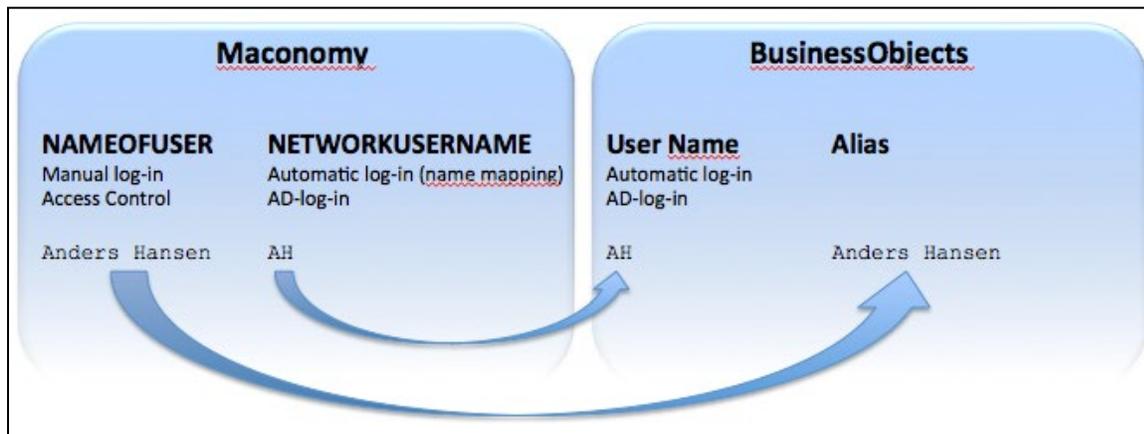
Example

- In Maconomy, Anders Hansen uses **AH** as the network user name (and thus also AD log-in).
- Anders Hansen logs in to the network and automatically logs in to Maconomy as **AH**.
- This gives Anders Hansen access to all of the Maconomy data that his user name is defined to have access to.

- When opening WebIntelligence in BusinessObjects, Anders Hansen is automatically logged in as **AH**.

- The BusinessObjects user **AH** has the alias Anders Hansen. Therefore, the user has access to all of the Maconomy data that Anders Hansen is defined to have access to.

> **Note:** With multi-user roles, Maconomy passes the login name as the user name.

-

Active Desktop and Maconomy Configurations

Users must exist as AD (Windows) users on the domain. The mapping in Maconomy must be set up properly so that each Maconomy user has his or her AD (Windows domain) user name specified in the **Network Username** field in the Users workspace in Maconomy.

For example, the person **Anders Hansen** might have the user name **AHA** on the domain. The Network user name for the Maconomy user **Anders Hansen** should therefore be filled in as **AHA.**

USync

You can use the USync tool to create users, user groups, and user aliases in BusinessObjects. USync creates BusinessObjects users and groups that correspond to the users in the Maconomy system. You can configure USync to create only users for which certain criteria are met.

USync can also remove BusinessObjects users or groups. This is convenient if users and groups have been removed from Maconomy.

Use the USync tool to complete the following tasks:

- Create users and groups in BusinessObjects that correspond to the ones that are defined in Maconomy. This allows users to access Maconomy data when running reports.

- Create aliases for each user so that, when logged in using another name by means of SSO, users can still access data.

- Synchronize users and groups for a tenant in a multitenant environment.

It is assumed that both Maconomy and BusinessObjects are set up for SSO.

Install USync

To install and prepare the USync tool:

1. Log in to the Maconomy server system.

2. Start MConfig and select the appropriate Maconomy application.

3. Click **Web products** and select the web server.

4. Click **OK** and apply the change with MConfig.

5. On the Web Products screen, enable **Web Services**.

6. Download the file **USync.zip** from the download server:

   \\dl\applications\Released\BPM\Tools\USync\BO 4.2\

7. Extract the zip file to a folder on the server where BusinessObjects is installed. Make sure that both the USync.jar file and the "external" folder are within the same folder.

Create Users and Groups

To transfer users and groups from Maconomy with USync:

1. Log in to the server on which BusinessObjects is installed.

2. Open a command prompt.

3. Run the USync java tool and enter the following parameters in the exact sequence as listed:

   - **<Maconomy administrator username>** – Enter the username of the Maconomy administrator to access the Web Service.

   - **<Maconomy administrator password>** – Enter the password of the Maconomy administrator to access the Web Service.

   - **<URL to MaconomyWS executable on web server>** – Enter the Maconomy web service URL from which USync derives the list of users to create in BusinessObjects.

   - **<BO username>** – Enter the username of the BusinessObjects user that has access to view, create, modify, and delete users in the CMS.

   - **<BO password>** – Enter the password of the BusinessObjects user that have the rights to view, create, modify and delete users in the CMS.

   - **<BO Server name>** - Enter the BusinessObjects server where users are created.

   - **<BO Authentication Type>** – Enter the authentication type that USync uses to log in the provided BO user above. It is recommended to use **secEnterprise**.

   - **<Log level>** – Set the value to **100** to show the minimum amount of information in the logs or to **500** to show a detailed trace of information.

   - **<Path and file name of log file>** – If there is no log file, the tool creates a new log file. If a log file already exists, the tool appends the logs to it. Note that USync does not create folders.

   - **<Delete-user-data (true/false)>** – This determines whether to allow USync to automatically delete a user in BusinessObjects that no longer matches with a user in Maconomy.

   - **<Create named BO users (true/false)>** – This determines the license type used to create the users. Set this to **true** to create named users (one license required per user) or **false** to create concurrent users (number of license pertains to the number of users that can login at the same time, no limit on the number of users that can exist in the CMS)

   - **<Create AD aliases in BO (true/false)>** – This determines whether BusinessObjects requires the configuration of the Windows AD plugin. Set this to **true** to automatically assign an alias to users that exists in Windows Active Directory.

   - **<Domain name used for AD alias creation>** – Enter the combined prefix and extension of the domain name used for AD alias creation (for example, trifolium.com). This is only necessary if the "Create AD aliases in BO" parameter is set to **true**.

   - **<Default password for new BO users>** – Enter a default password that all new created BusinessObjects users can use. This is only used for Enterprise authentication.

   - **<Passwords never expire (true/false)>** – To determine the expiration of passwords, enter **true** to ensure the user's password will never expire. If you enter **false**, the user's password expires according to the configured user restriction in **CMC » Authentication » Enterprise**. By default, this parameter's value is set to **false**.

   - **<Must change password (true/false)>** – To determine whether the user must change the default password when first logging in, enter **true** to force users to change password on first

log in. Note that you cannot set this parameter to **true** if the "Can't change password" parameter is also set to **true**.

- **<Can't change password (true/false)>** – To determine whether who can set passwords, enter **true** to allow only the system administrators to set passwords. Note that you cannot set this parameter to **true** if the "Must change password" parameter is also set to **true**.

- **<Whitelisted group>** – Use this parameter to determine which users should not be deleted by USync when the "delete-user-data" parameter is set to **true**. Leave this blank if no group needs to be whitelisted.

- **<Allow update of existing BO users>** – This parameter controls whether the properties of an existing user in BusinessObjects is updated by USync. By default, this parameter's value is set to **false**.

For example, if you enter:

```
java -jar USync.jar "Administrator" "123456" "http://BPMServer:20001/cgi-bin/Maconomy/MaconomyWS.w15p2mc.US_MCS.exe/" "Administrator" "ppu" "macsrv.trifolium.com" "secEnterprise" "100" "C:\temp\usync.log" "false" "false" "true" "trifolium.com" "1234-Pass" "true" "false" "false" "" "true"
```

This does the following:

- Creates users and groups that correspond to users and groups that are set up in Maconomy. If only certain users are wanted, you can define criteria for them.

  > **Note:** See Configuring a Tenant in a Multitenant Environment (Optional Installation).

- Creates aliases for these users to handle access control.

- Does not remove existing users that are defined in BusinessObjects. The users that are created are in addition to the users that are already there.

- Sets the default password of the new users to **1234-Pass**.

- Uses the web server executable (a 15.0 sp 2 MCS system in the preceding example) to determine the Maconomy system that is used.

- Stores the log output from USync in the file **C:\temp\usync.log**.

4. Open the Central Management Console and click **Groups** or **Users** to confirm that the synchronization succeeded.

For each user, you can see the group memberships on the Member of tab. For each group, you can see the members of the group on the Users tab. If you chose to create AD aliases, you can see them under each user in the lower part of the window.

Configuring a Tenant in a Multitenant Environment (Optional Installation)

Multitenancy enables you to host multiple clients on one server where each client has their own access sites and properties. These additional steps are optional and not part of the Usync default installation.

The setup of multitenancy includes the following steps:

- Tenant configuration in the Maconomy Workspace client.
- Set up corresponding tenant in BusinessObjects server.

- Synchronize users using Usync.

Set Up a Tenant Name in Maconomy

To set up a tenant name in Maconomy:

1. Open the **System Information** single dialog workspace.

2. Scroll down to the Tenant Name island and enter a unique name in the **Tenant Name** field.
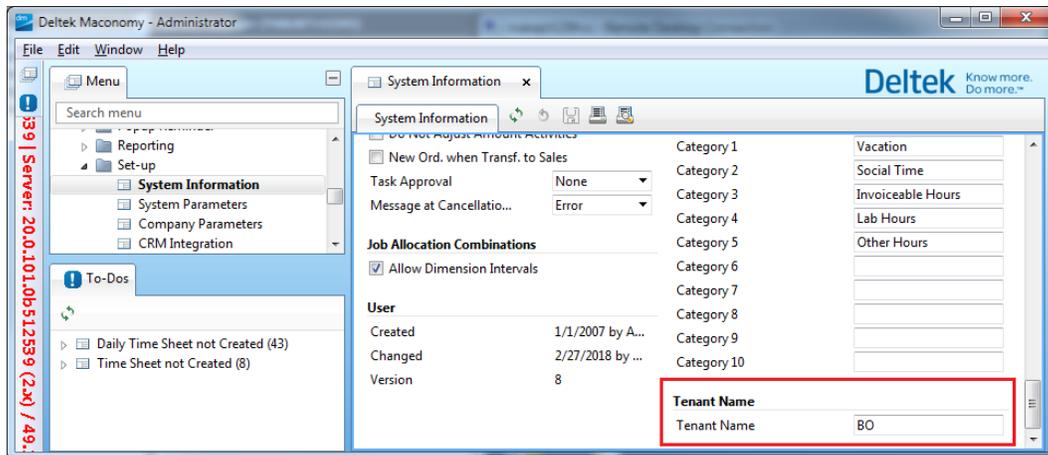
Note: To check if a tenant name is already in use, go to BusinessObjects CMC » Multitenancy.

3. Click **Save**.

Validate the Tenant Name in Maconomy:

To validate the tenant name in Maconomy:

1. Log in to the Maconomy Workspace Client with the necessary rights to access system setup, preferably as Administrator.

2. Open the **Users** workspace (Setup » Users) and expand the **User Information** tab.

3. Add the **Tenant Name** column to the Roles tab.

   a. Right-click any column on the Roles tab and click **Customize columns**.

   b. In the Available Columns section, select **Tenant Login Name**.

   c. Click **Add** and move the column up or down, depending on the desired location.

   d. Click **OK**.

4. In the Tenant Login Name column, verify that the tenant name is prefixed in the login name with the format **tenantName#LoginName**.
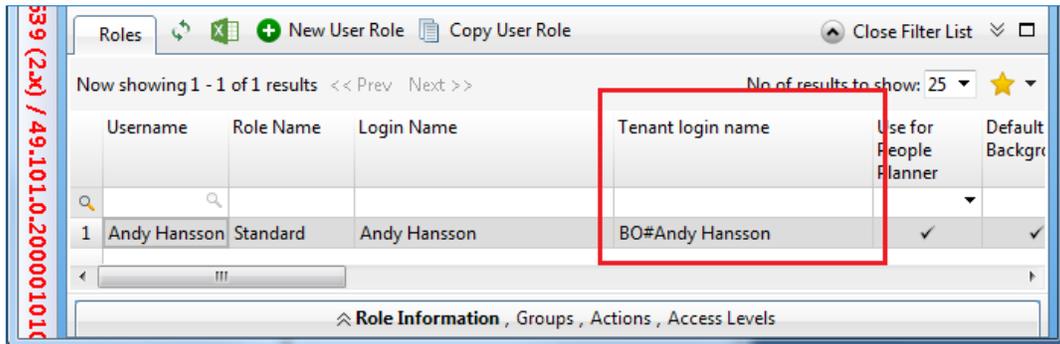
Register Tenant with BusinessObjects 4.2 Multitenancy Management Tool (Windows)

BusinessObjects 4.2 Business Intelligence Platform comes pre-installed with the Multitenancy Management Tool (MTM). This tool handles the registration of a tenant and other configurations controlled through a template definition file.

> **Tip:** Refer to the SAP Multitenancy Guide for more information.

To define MTM properties:

1. Open the **tenant_template_def.properties** file in a text editing application. The file is located in the same directory as the BusinessObjects 4.2 Multitenancy Management Tool.

   Note: The default location and file name is <InstallDir>\SAP BusinessObjects Enterprise XI 4.0\java\apps\multitenancyManager\jars\multitenancymanager.jar.

2. In the **tenant_template_def.properties** file:

   a. Add the desired tenant name to the tenantName parameter. For example, tenantName=XXX.

   b. Add the mandatory login information:

      - cms

      - auth

      - user

      - pwd (Optional. If left blank, user is prompted for password to proceed.)

3. Save the file.

Run the MTM tool

To run the MTM tool:

1. Open a command prompt. It is recommended that you run this as Administrator to avoid permission conflicts.

2. Change the directory to the location of the multitenancymanager.jar file (<InstallDir>\SAP BusinessObjects Enterprise XI 4.0\java\apps\multitenancyManager\jars)

3. Run the following command:

```
java -jar multitenancymanager.jar -configFile
```

Verify the Tenant is Created Successfully

To verify the tenant is created successfully:

1. Log in to the **Central Management Console**.

2. On the Multitenancy tab, verify the tenant name is listed in the **Name** column.

Synchronize Users with USync

To synchronize users with USync:

1. Download the **USync.zip** from the dl server at \\dl\applications\Released\BPM\Tools\USync\BO 4.2 Multitenancy\.
2. Extract the zip files.
3. Open a command prompt (run as Administrator).
4. Change the directory to the location of the extracted zip file, **USync.jar**.
5. Run the command: **java -jar USync.jar** with the following parameters:

PARAMETER	VALUE
Maconomy Username	Maconomy Admin username
Maconomy Password	Maconomy Admin password
Web Service URL	MaconomyWS URL found in ..\maconomy\index.html
BO Username	BO Admin username
BO Password	BO Admin password
CMS	BO Server
BO Authentication	secEnterprise, secWinAD, etc.
Log Level	0 = no message, 500 = debug mode
Log File Name	File for the log file including directory
Delete User Data	delete users that does not exist in Maconomy
Named Users	Boolean, true=named user or false=concurrent user
Create AD Aliases	Boolean, create AD Alias for SSO
AD	Domain used for AD Alias Creation
Default Password	Password assigned to created BO users
Passwords Never Expire	Boolean, user attribute
Must Change Password	Boolean, user attribute
Can't Change Password	Boolean, user attribute

PARAMETER	VALUE
White List Groups	Groups ignored by delete user data parameter
Allow Update	Allows USync to update existing users

Example:

```
C:\USync>java -jar USync.jar "Administrator" "MaconomyPassword" "http://localhost:20003/cgi-bin/Macon
omy/MaconomyWS.shortname.en_US.exe/" "Administrator" "BOPassword" "BOServer" "secEnterprise" "500"
"C:/USync.log" "false" "true" "false" "" "DefaultPassword" "true" "false" "false" "SysAdmins" "false"
```

6. When completed, verify that the users and groups are created in BusinessObjects CMC.
   a. Log in to **Central Management Console**.
   b. On the **Users and Groups** tab, verify both users and groups are prefixed with the tenant name.

Control which Users to Create

To create only specific users, or users that satisfy certain conditions, configure USync.

To configure USync, follow these steps:

1. In the Maconomy web server folder **\MaconomyWS\Services\Standard** locate the **USyncSQL.txt** file.
2. Edit the file so that it contains an SQL-like expression that selects the users.

Examples

Example 1: Select Users by Means of Popup 1

You want to control which user to import by using Popup 1 on the employee who is associated with the user. If you set up this pop-up with one value, BusinessObjects, so that it either has this value or is blank on each employee, you can insert the following additional condition in the WHERE clause for the import of user data and role membership data.

```
0 = (select EmployeePopup1 from Employee
    where EmployeeNumber = UserInformation.EmployeeNumber)
```

If there is no employee on the user, the selected value is **null**. If Popup 1 is blank on the employee, the selected value is **-1**.

Example 2: Select Users Based on Roles

You want to import only those users who have certain roles. In this case, use the following additional condition:

```
EXISTS (SELECT 'OK' FROM EXUserDialogGroup
    WHERE NameOfUser = UserInformation.NameOfUser
        AND GroupName IN ('FinancialManagement', 'LineManagement'))
```

In an MAS solution, this imports 19 users, including Andy Polansky, Lisa Welsh, and Sue French. In the MCS solution you could change the restriction on the group name to **GroupName IN** (**FinancialManagement, DepartmentManagement, AccountManagement**), and then you would get 17 users, including Edward Powers and Micho Spring.

Error Handling

The output message may contain errors.

- If the output message contains:

  - "This user does not exist" — You did not enter a valid Maconomy user name.

  - "Password is not correct" — You did not enter a correct password for the Maconomy user.

  - "Enterprise authentication could not log you on" — There is a problem with the BusinessObjects username or password.

  - If you cannot open the **http://localhost/index.html** page, there may be an error in the global parameters in M-Config. Open M-Config, click the **Global settings** button and make sure that the **Export HTML index file to** field has the correct value (for example: C:\Apache\Apache2\htdocs).

(404) Not Found

- If the output message contains "(404) Not Found," there is a problem with the URL in the **Web service URL** field. This is also the case if the output message refers to SAXParseException, ConnectException, UnknownHostException, or "unknown protocol."

  One possible reason for the 404 error could be that the Webserver URL in M-Config is incorrect.

To fix the 404 error, follow these steps:

1. Open M-Config.

2. Select the relevant application.

3. Click **Web products**.

4. Select the appropriate web server.

5. Click **Web server parameters** and make sure that the **Webserver URL** (host and port) field has the correct value.

   For example, if the installation uses a virtual web server, the field should be **http://** followed by the name of the server, followed by the virtual web server port number (separated by a colon), with a **/** (slash) at the end. If you make a change in M-Config, you may need to restart the web daemon before the changes take effect.

Assign Report and Universe Rights

Before you perform this procedure, you must consider which users or groups should be given access to which reports and universes.

To manually set up access rights in BusinessObjects, follow these steps:

1. Open the Central Management Console.

2. Click **Folders** and then locate the reports in the Maconomy folder.

3. For each report, set up the access rights by clicking on the report and opening the Rights tab.

4. Make sure that the rights for the group **Everyone** are set to **No Access**, click **Add/Remove** to insert a line (for example, a new group), and then click **OK**.

5. Change the rights to a different setting, for example, **View On Demand**.

Note: You must grant each universe and connection to the specific groups of users. This may take some time if there are many groups and many reports.

It is recommended that you assign rights to groups (containing users), rather than directly to users.

Configure Logon Token in BusinessObjects

With the Dashboard Components iFrames installed, an "obsolete token error" might occur if the token that is used for signing in to BusinessObjects has expired.

There are two ways to avoid this error, both of which involve changing the files that are deployed in BusinessObjects:

- Increase available document instances
- Prevent users from using outdated tokens

Increase Available Document Instances

To increase the number of document instances that are available on the system, modify the parameters in the webi.properties file.

To modify the webi.properties file appropriately, follow these steps:

1. Locate the webi.properties file at C:\Program Files(x86)\Business Objects\Tomcat55\webapps\AnalyticalReporting\WEB-INF\classes\webi.properties.
2. Uncomment the WID_FAILOVER_SIZE=11 and WID_STORAGE_TOKEN_STACK_SIZE=11.
3. Save and close the file.
4. Restart the web application server.

Prevent Users from Using Outdated Tokens

To prevent users from using outdated tokens, set the **logontoken.enabled** parameter in the web.xml file.

To modify the web.xml file appropriately, follow these steps:

1. Locate the web.xml file at C:\Program Files(x86)\Business Objects\Tomcat55\webapps\InfoViewApp\WEB-INF\web.xml.
2. Find the **logontoken** phrase and change the parameter value from true to false.

   ```
   <context-param>

       <param-name>logontoken.enabled</param-name>
       <param-value>false</param-value>

   </context-param>
   ```
3. Save and close the file.
4. Restart the web application server to apply the changes.

Handling BusinessObjects Services

The standard BusinessObjects Enterprise installation has several services that are set up to start automatically. For this reason, the server startup is slower. However, if you do not need all of the services, the following information explains how to change the startup type on some of them, including:

- Changing the start-up type
- Removing Performance Manager

Change Start-Up Type

To change the start-up type, follow these steps:

1. Click the Windows **Start** button, then click **All programs » SAP Business Intelligence » SAP BusinessObjects BI platform 4 » Central Configuration Manager » Manage Servers**.
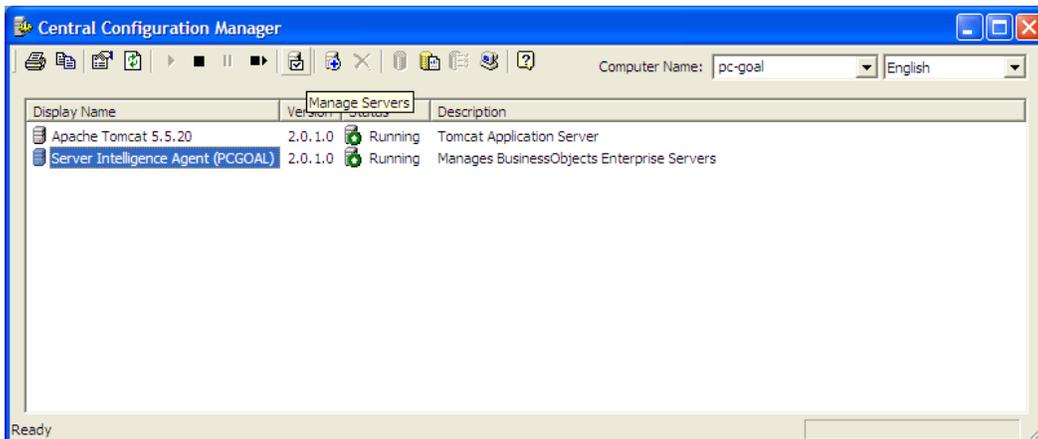
2. Log in to the CMS.

   A list of the BusinessObjects servers is displayed.
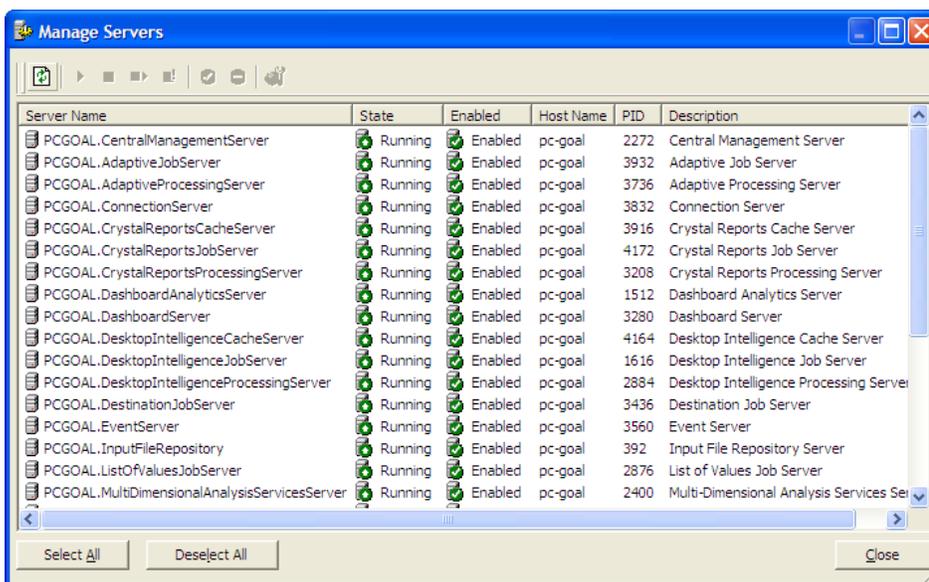
3. You might want to stop the following list of servers and change their startup type to **Manual** in their properties:

- Crystal Reports Cache Server

- Crystal Reports Job Server

- Crystal Reports Processing Server

- Desktop Intelligence Cache Server

- Desktop Intelligence Job Server

- Desktop Intelligence Processing Server

Remove Performance Manager

To restrict access to Performance Manager, follow these steps:

1. Click the Windows **Start** button, then click **All programs » SAP Business Intelligence » SAP BusinessObjects BI platform 4 » SAP » BusinessObjects Enterprise Central Management Console**.

2. Click the Applications tab.
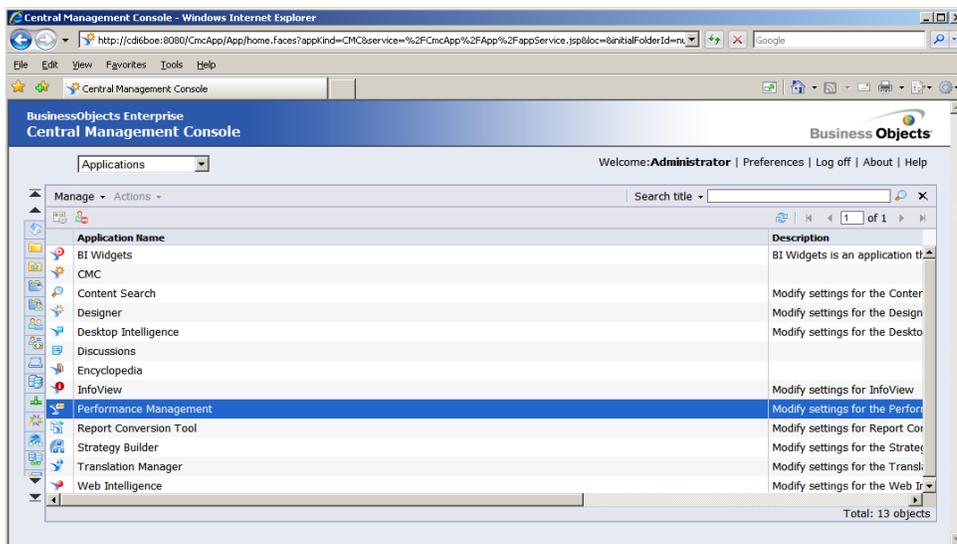
3. Right-click **Performance Management**.
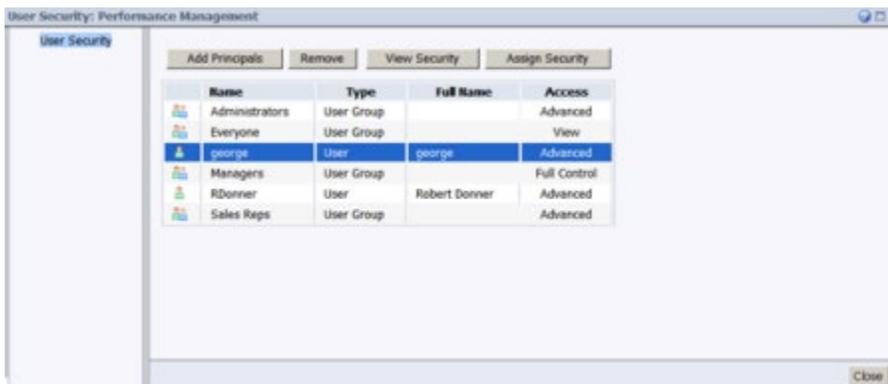
4. Choose a group and click **Assign Security**.

5. Click the Advanced tab, and then click **add/remove rights**.
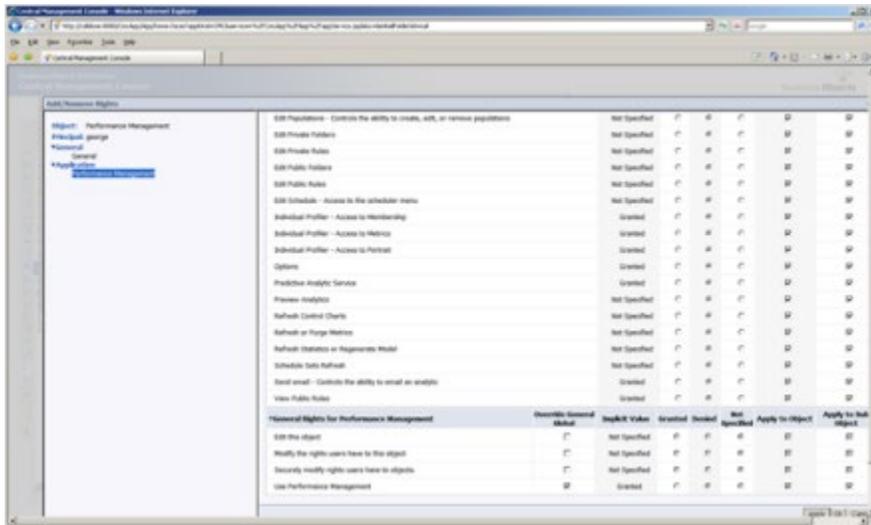
6. Clear all options, including **Use Performance Management**.

7. Click **Apply** and then click **OK** to save your entries.

Use a Non-Standard Web Server Port

If the port specified for the Tomcat server is already in use, you cannot log in to InfoView. You must reconfigure the Tomcat server by changing the port number in **…\Tomcat\conf\server.xml** under the Connector tag and restart the Apache Tomcat service.

To avoid this issue, follow these steps:

1. Open a Command Prompt and enter the following:

   ```
   netstat -a > netstat.txt
   ```

   This displays an **etstat.txt** text file in the current directory that contains, among other information, a list of the port numbers that are currently in use.

2. Open this text file and search for **8080**.

3. If **8080** is already in use, insert a unique number and use that port number, instead of 8080, throughout this section.

Set Templates for Web Intelligence

You must change the template for WebI when there is a new document. This is not an option for existing documents. The Maconomy standards are implemented in the defaultConfig.xml released file.

To set templates for WebI, follow these steps:

1. Make a backup copy of the **AnalyticalReporting.war** file from (default) **C:\Program Files\Business Objects\BusinessObjects Enterprise 12.0\java\applications**.

2. Open the **AnalyticalReporting.war** file and replace the three versions of **defaultConfig.xml** with the changed versions. Note the original paths:

    ▪ webiApplet\AppletConfig\defaultConfig.xml

    ▪ webiApplet\AppletConfig\AppletConfig\defaultConfig.xml

    ▪ WEB-INF\classes\defaultConfig.xml

3. From services, **Stop tomcat**.

4. From a command-line prompt, change the current directory to **C:\Program Files\Business Objects\deployment**.

5. Enter the command **wdeploy tomcat55 predeployall**.

6. If no errors are displayed, enter the command **wdeploy tomcat55 deployall**.

7. From services, **Start tomcat**.

8. Create a WebI document to check the template's performance.

9. Restart the server if necessary.

Enabling Cache and Pre-Load of Cache in ETL

Enabling cache on ETL effectively improves the load speed of both incremental and full loads in the data warehouse. It stores the values of keys that had been looked up at least once and returns these values the next time that these are looked up again. This reduces the need to do a database call every time a lookup step is processed.

> **Note:** Cache uses memory to store the looked-up key, and you risk **running out of memory** when it is enabled on very large tables.

You can enable cache:

▪ Manually

▪ Using an SQL Script

Enable Caching Manually

To enable caching manually, follow these steps:

1. Open the transformations that contain dimensional or database lookup steps.

2. Right-click the lookup step and select **Edit Step** or double click it.

3. Select the **Enable the cache?** check box.
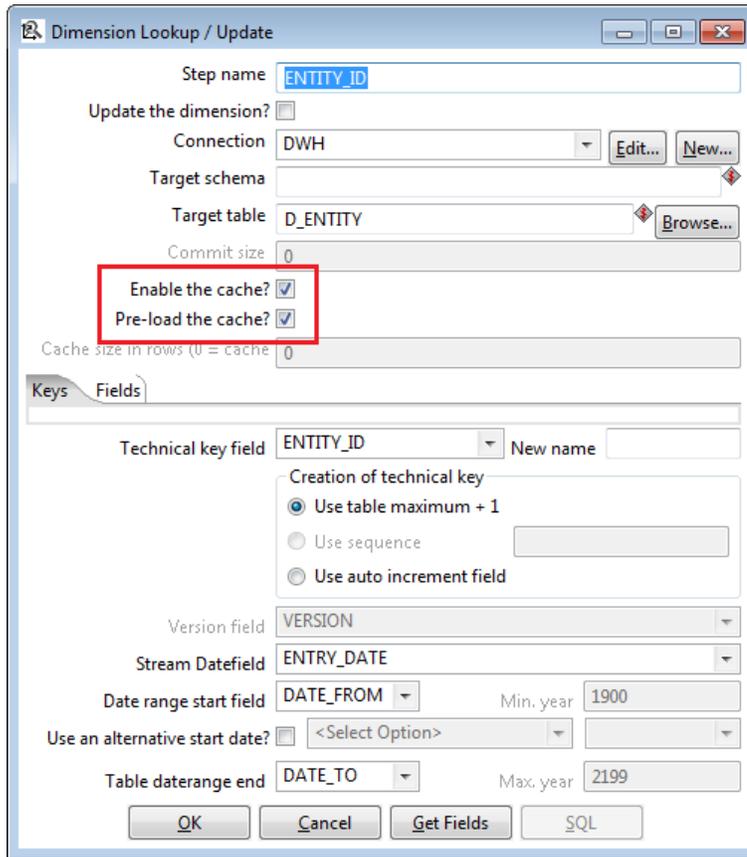
4. If **Pre-load the cache?** is enabled, select it as well. This check box is disabled if the **Update the dimension?** check box is selected.

5. Set **Cache size in rows** to **0** to enable caching of all rows.

6. Click **OK** to apply changes.

7. Apply these changes to all other lookup steps except **Pre-load the cache?** on **TASK_ID** because this is typically a very large table and will result in an out of memory error. Set **Cache size in rows** to **1** for this lookup and then append **ORDER BY TASK_LIST, TASK_NAME** on the SQL statement of the Table Input step. TASK_ID lookup step is present on the following transformations:

   ▪ Job Entry

   ▪ Purchase Order Line

   ▪ Job Invoice Line

   ▪ Budget Entry

   ▪ Finance Entry

   ▪ Job Budget Line

Enable Caching Using an SQL Script

- Another way to enable the cache and pre-load the cache is by using an SQL script.

To enable caching using an SQL script, complete the following steps:

1. Log in to the database repository for your ETL and run the following SQL statements.

```
-- enables cache for Dimensional Lookup steps except TASK_ID
UPDATE R_STEP_ATTRIBUTE
SET VALUE_NUM = 0, VALUE_STR = NULL
WHERE ID_STEP_ATTRIBUTE IN (
SELECT ID_STEP_ATTRIBUTE
FROM R_STEP_ATTRIBUTE
WHERE ID_STEP IN (
SELECT ID_STEP
  FROM R_STEP
  WHERE ID_STEP_TYPE IN
    (SELECT ID_STEP_TYPE
       FROM R_STEP_TYPE
      WHERE CODE IN ('DimensionLookup')) AND
      NAME <> 'TASK_ID') AND
      CODE IN ('cache_size')
);
--enables pre-load cache for Dimensional Lookups except TASK_ID
UPDATE R_STEP_ATTRIBUTE
SET VALUE_STR = 'Y'
WHERE ID_STEP_ATTRIBUTE IN (
SELECT ID_STEP_ATTRIBUTE
FROM R_STEP_ATTRIBUTE
WHERE ID_STEP IN (
SELECT ID_STEP
  FROM R_STEP
  WHERE ID_STEP_TYPE IN
    (SELECT ID_STEP_TYPE
       FROM R_STEP_TYPE
      WHERE CODE IN ('DimensionLookup')) AND
      NAME <> 'TASK_ID') AND
      CODE IN ('preload_cache')
);
--set cache-size to 1 for Dimensional Lookup TASK_ID
UPDATE R_STEP_ATTRIBUTE
SET VALUE_NUM = 1, VALUE_STR = NULL
WHERE ID_STEP_ATTRIBUTE IN (
SELECT ID_STEP_ATTRIBUTE
FROM R_STEP_ATTRIBUTE
WHERE ID_STEP IN (
SELECT ID_STEP
  FROM R_STEP
  WHERE ID_STEP_TYPE IN
    (SELECT ID_STEP_TYPE
       FROM R_STEP_TYPE
      WHERE CODE IN ('DimensionLookup')) AND
      NAME = 'TASK_ID') AND
      CODE IN ('cache_size')
);
--disables pre-load cache for TASK_ID Dimensional Lookup steps
UPDATE R_STEP_ATTRIBUTE
SET VALUE_NUM = 0, VALUE_STR = 'N'
WHERE ID_STEP_ATTRIBUTE IN (
SELECT ID_STEP_ATTRIBUTE
```

```
FROM R_STEP_ATTRIBUTE
WHERE ID_STEP IN (
SELECT ID_STEP
  FROM R_STEP
   WHERE ID_STEP_TYPE IN
    (SELECT ID_STEP_TYPE
       FROM R_STEP_TYPE
        WHERE CODE IN ('DimensionLookup')) AND
        NAME = 'TASK_ID') AND
        CODE IN ('preload_cache')
);
--enables cache for Database Lookup Steps (VALUE_STR = 'Y')
UPDATE R_STEP_ATTRIBUTE
SET VALUE_STR = 'Y'
WHERE ID_STEP_ATTRIBUTE IN (
SELECT ID_STEP_ATTRIBUTE
FROM R_STEP_ATTRIBUTE
WHERE ID_STEP IN (
SELECT ID_STEP
  FROM R_STEP
   WHERE ID_STEP_TYPE IN
    (SELECT ID_STEP_TYPE
       FROM R_STEP_TYPE
        WHERE CODE IN ('DBLookup'))) AND
        CODE IN ('cache')
);
--set cache size to 0 for Database Lookup Steps (VALUE_NUM = 0)
UPDATE R_STEP_ATTRIBUTE
SET VALUE_NUM = 0, VALUE_STR = NULL
WHERE ID_STEP_ATTRIBUTE IN (
SELECT ID_STEP_ATTRIBUTE
FROM R_STEP_ATTRIBUTE
WHERE ID_STEP IN (
SELECT ID_STEP
  FROM R_STEP
   WHERE ID_STEP_TYPE IN
    (SELECT ID_STEP_TYPE
       FROM R_STEP_TYPE
        WHERE CODE IN ('DBLookup'))) AND
        CODE IN ('cache_size')
);

COMMIT;
```

2. After running the provided SQL statement, commit the changes. Then you must manually update transformations that contain TASK_ID (see step 7 on how manually enable cache.)

Servers with Low Memory (RAM)

Pentaho Data Integration 5.4 has a high memory consumption. This is due to Pentaho Data Integration 5.4 installation, as well as installing Tomcat 6 as a necessary component to run the Data Integration Server. However, it consumes a steadily increasing amount of memory, which could be detrimental to systems with low memory.

Data Integration Server is mainly used for scheduling, execution, security, and content management. Since Deltek currently uses a different method of scheduling ETL loads, you can disable it.

> **Note:** This process is optional and is only necessary for servers with low memory (RAM).

To disable Data Integration Server, complete the following steps:

1. Click **Start**, then type *Services.msc*.

2. The Services window is displayed. Locate a service called **Data Integration Server**.
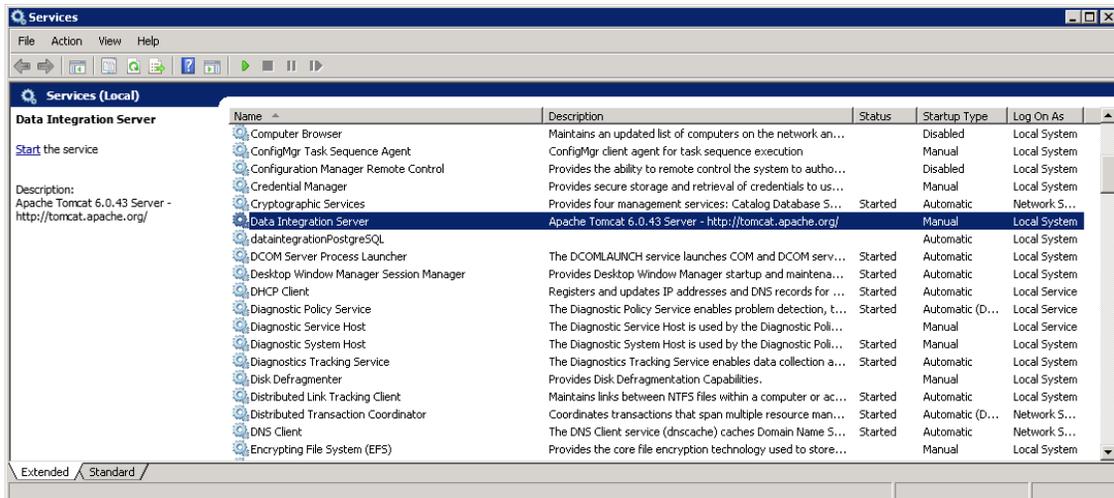
3. Right-click and select **Properties**. The Date Integration Server Properties dialog is displayed.

4. In the **Startup type** field, change the selection to **Manual** so that the service does not run automatically when you restart the system.
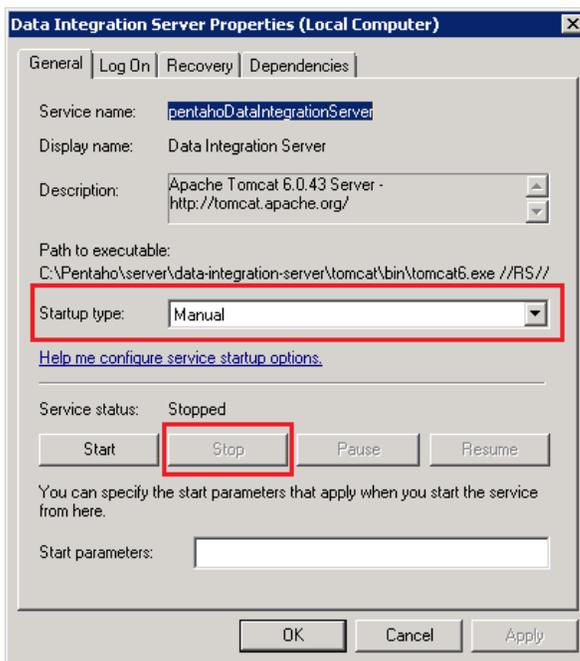
5. Click **Stop** to disable the service, then **OK**.

Burst Publishing

Burst Publishing is the ability to schedule documents to refresh and then automatically distribute them out to receivers. The main benefit is that the recipients (Maconomy users) can get the refreshed documents automatically in their in boxes, and that the documents are only run once.

For information about how to set up burst publishing of documents in SAP BusinessObjects, see
http://scn.sap.com/docs/DOC-56056

Materialized Views on Oracle

A materialized view on Oracle is a database view that is stored physically and provides a view on which indexes and keys can be defined. BPM Reporting uses one materialized view in the Job Information universe because the Maconomy transaction database lacks a table that combines job entries (JobEntry) with corresponding invoice lines (JobInvoiceLine).

This section describes how materialized views on Oracle work and how to manage them.

JobEntryJobInvoiceLinePV

The materialized view that is used in BPM Reporting is called JobEntryJobInvoiceLinePV. It consists of the following elements:

- Two view logs, one on JobEntry and one on JobInvoiceLine
- Materialized view JobEntryJobInvoiceLinePV
- EX view EXJobEntryJobInvoiceLinePV

When records change in the JobEntry table, the changes are recorded in the view log. Similarly, if records are inserted or deleted, changes are recorded in the JobInvoiceLine table.

The Materialized View

The materialized view is set to update on commit. This means that when a transaction changes, inserts, or deletes records in JobEntry or JobInvoiceLine, the changes that are recorded in the view logs are transferred as changes that update the materialized view. The view is designed to perform a fast refresh because only delta changes are considered; it is not the whole view that is updated. In addition, the view does not use the query rewrite mechanism in Oracle.

The EX View

The EX view is a database view that is on top of the materialized view. The EX view provides meaningful values for pop-up numbers (enumerated data types) and date values of data types (in addition to the string format).

For technical reasons, the date definitions are done in the materialized views, and only pop-up value fields are handled in the EX views (in other EX views, date fields are also handled in the EX views).

Explain the View Definition

This section provides the view definition file and explains what each statement does.

Statement/Section	Definition
`ALTER SYSTEM SET "_mv_refresh_use_stats =TRUE"`	This statement tells Oracle to use statistics when refreshing materialized views.
`CREATE MATERIALIZED VIEW LOG ON JOBENTRY`	This statement creates the view log on JobEntry.
`WITH ROWID`	The reference of ROWID is important because it is used for unique identification.

`INCLUDING NEW VALUES;`
 	
`CREATE MATERIALIZED VIEW`
`JOBENTRYJOBINVOICELINEPV`
	This statement creates the actual materialized view.

`REFRESH ON COMMIT AS`	This is the definition of "refresh on commit."
`SELECT`	This statement starts the select statement that defines the view.
`  -- Technical fields` `   JOBENTRY.ROWID JOBENTRYROWID`	Both parts of the union that make the select statement use the ROWID internally for unique identification.
`,NULL JOBINVOICELINEROWID` `    ,1 MARKER`	This field is used to distinguish the upper and lower union parts.
`-- Shared Dimensions` `   ,JOBENTRY.INSTANCEKEY`	This starts the field list.

`CREATE VIEW EXJOBENTRYJOBINVOICELINEPV AS`
`:`
`SELECT`
 	
`       INSTANCEKEY`
 	
`      ,JOBNUMBER`
 	
`      ,ENTRYNUMBER`
	This defines the EX view on the materialized view.

Check the Materialized View

Use the following select statements to check the view.

```
SELECT MVIEW_NAME, REFRESH_MODE, REFRESH_METHOD, LAST_REFRESH_TYPE, LAST_REFRESH_DATE
FROM USER_MVIEWS;
```

This lists a record for the materialized view that is defined. The **last_refresh_date** provides the date when the view was last refreshed, and the other fields indicate how it was refreshed.

```
SELECT LOG_OWNER, MASTER, LOG_TABLE FROM DBA_MVIEW_LOGS;
```

This lists two records, one for each view log.

```
SELECT COUNT(*) FROM MLOG$_JOBENTRY;
SELECT COUNT(*) FROM MLOG$_JOBINVOICELINE;
```

These statements display the number of changes currently recorded in the view logs. Under normal circumstances, they should display zero. If they do not, it is because they currently store changes that are not transferred to the materialized view.

If it appears that the preceding tables are not emptied at any time, it can be because the materialized view is not able to empty them as it should. One reason can be that more than the two view logs have been installed, but without the corresponding materialized views to empty them.

To check that `_mv_refresh_use_stats` system parameter is set up, use the following command:

```
SHOW PARAMETER _mv_refresh_use_stats;
```

It should display the following.

```
NAME                          TYPE      VALUE
_____  _____  _____
_mv_refresh_use_stats boolean TRUE
```

If no data is displayed, the parameter is not set up.

Flush the Shared Pool

Over time, SQL statements for updating the materialized view compile in the shared pool. This is a known issue in Oracle and requires that you empty the shared pool. A stored procedure has been developed for flushing the shared pool every night.

> **Tip:** See *Performance Views* for instructions on how to install the stored procedure.

To check that the flushing procedure runs as expected, execute the following command. FAILED should be zero and BROKEN should be "N."

```
select job, what, last_date, next_date, failures, broken from dba_jobs where what like
'%maconomy_flush_shared_pool%';
```

To check that the flushing procedure cleaned the shared pool, execute the following command. If it is successful, it should return no entries.

```
SELECT sql_text FROM v$sql WHERE sql_text LIKE '/* MV_REFRESH%';
```

Handling Issues

If the Maconomy database performs poorly, it may be because of an incorrect installation of the materialized view.

To resolve this issue, follow these steps:

1. Run the select statements mentioned in Check the Materialized View. Check the results to find out whether the view is updated at all, and whether it is refreshed quickly.

2. Run the select statements mentioned in Flush the Shared Pool to find out whether the stored procedure is not flushing correctly and how many entries there are in the shared pool.

3. Drop the view. Use the drop statements of the view definition file. Start with dropping the actual materialized view and then drop the view logs. There is no need to drop the indexes.

4. Apply the view again and check its installation.

5. Install the flushing mechanism again and check its installation.

TROUBLESHOOTING

Determine BPM Version and Solution

The internal version number, the external version number, and the solution appear in universes and documents of a BPM release. In universes, the version numbers and solution are stated as custom properties of the data foundations (DFX files). In documents, they are stated in variables.

The format of internal version numbers is:

```
<version>.<sub-version>.<service pack>.<fix pack>
```

For example: `17.0.102.0`

Problem with Login to Infoview through the Portal

You cannot log in to BusinessObjects in the Portal using Internet Explorer (ID). The Login screen is displayed, but nothing happens when you try to log in (you stay on the same page). Instead, you can log in directly in **infoview**.

The reports are embedded into the Maconomy Portal using IFrame. IE gives a lower level of trust to IFRAME pages (IE calls this "third-party" content). If the page inside the IFRAME does not have a Privacy Policy, its cookies are blocked. This is indicated by the eye icon in the status bar. When you click this icon, it lists the blocked URLs. The cookies are used to store the BusinessObjects session. Because the session cannot be saved, the user cannot log in.

To resolve the login problem, follow these steps:

1. Create a folder named **w3c** in **[Business Objects]\Tomcat55\webapps**

2. Create a file called **p3p.xml** in the same folder with the following code:

   ```
   <META>
       <POLICY-REFERENCES>
           <POLICY-REF about="/w3c/policy.xml">
               <INCLUDE>/</INCLUDE>
               <COOKIE-INCLUDE/>
           </POLICY-REF>
       </POLICY-REFERENCES>
   </META>
   ```

> **Note:** This procedure does not work when you use an IP instead of a server name in the OpenDocument link.

About Deltek

Better software means better projects. Deltek is the leading global provider of enterprise software and information solutions for project-based businesses. More than 23,000 organizations and millions of users in over 80 countries around the world rely on Deltek for superior levels of project intelligence, management and collaboration. Our industry-focused expertise powers project success by helping firms achieve performance that maximizes productivity and revenue. www.deltek.com

Deltek