

MFR83 Blueprint

Installation Guide

IMPORTANT :

This document was last updated Summer 2008

Please check the Proven Practice website
(<http://support.cognos.com/supported/tti/public/index.html>)
for any updates/later versions of this document

(RJC)

• BluePrints	3
○ System Requirements	4
• Installation guide	5
○ Step1 – Publish	6
○ Step2 – Release	11
○ Step3 – Enhance	15
• Enhance – Data Manager	16
• Enhance – SQL 2000 DTS	18
• Enhance – SQL Server 2005 SSIS	21
▪ Cognos Transformer	23
▪ Transformer Automation	26
▪ MFR Trigger	28
• MFR Trigger – Data Manager	28
• MFR Trigger – SQL 2000 DTS	34
• MFR Trigger – SQL 2005 SSIS	37
• Appendix 1	41
• Appendix 2	43
• Appendix 3	45
• Appendix 4	46

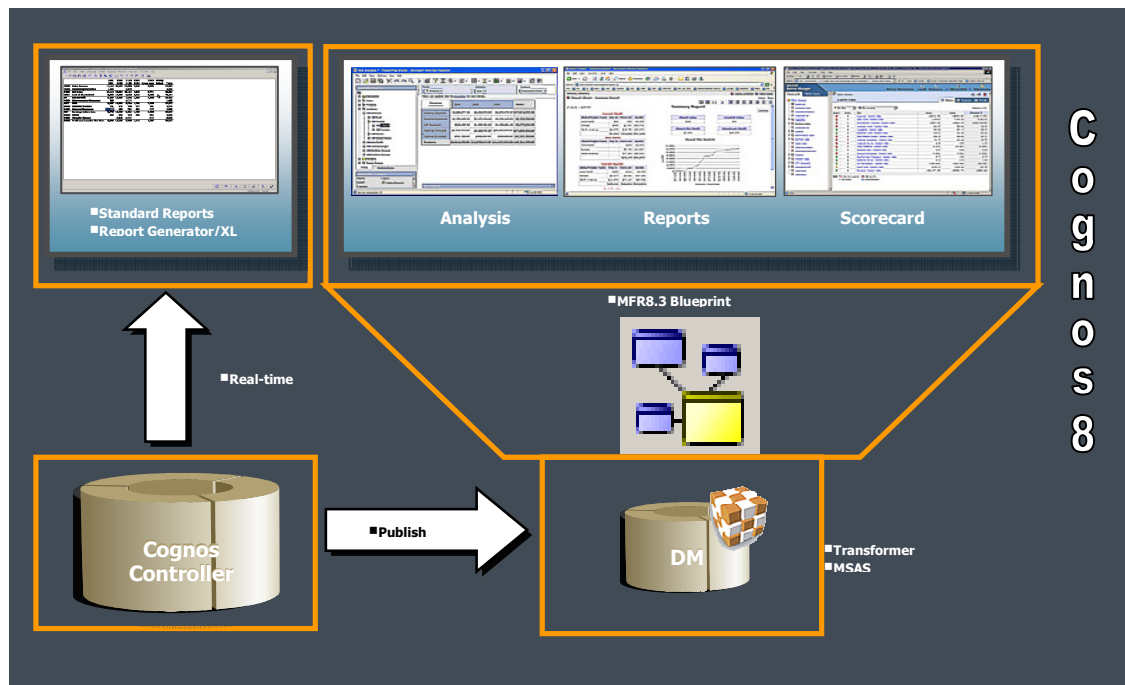
BLUEPRINTS

What is a *Blueprint*?

Before you start to implement the *MFR83 Blueprint* there are few things you should now about a *Blueprint*; what is it and how to use it.

A *Blueprint* is not a product on the Cognos pricelist and mustn't be used as such, AND it's not supported. A *Blueprint* is an example of how you can use our products, in this case, how to use Cognos 8 BI on top of the Controller Data Mart.

Blueprints are a fantastic enabler that will speed up the process of delivering a solution that has been delivered in the past. The MFR83 Blueprint provides the knowledge and resources to quickly integrate Controller with Cognos BI.



Integrating Cognos Controller and Cognos 8 BI allows for enhanced reporting from the consolidated data, providing scorecards, dashboards and the ability to create adhoc reports easily using Cognos 8 BI.

System requirements

These are the applications that are used in the *MFR83 Blueprint*:

- Cognos BI 8.3 sp1 +
- Cognos Controller 8.3 sp1 +
- Cognos Data Manager 8.3.81.20 (Custom installation)
- Cognos PowerPlay Transformer 8.3.818.0-0 (Custom installation)
- Cognos Access Manager 7.4.734.0 (Custom installation)

Please note that default installation of *MFR83* only requires Cognos BI and Cognos Controller.

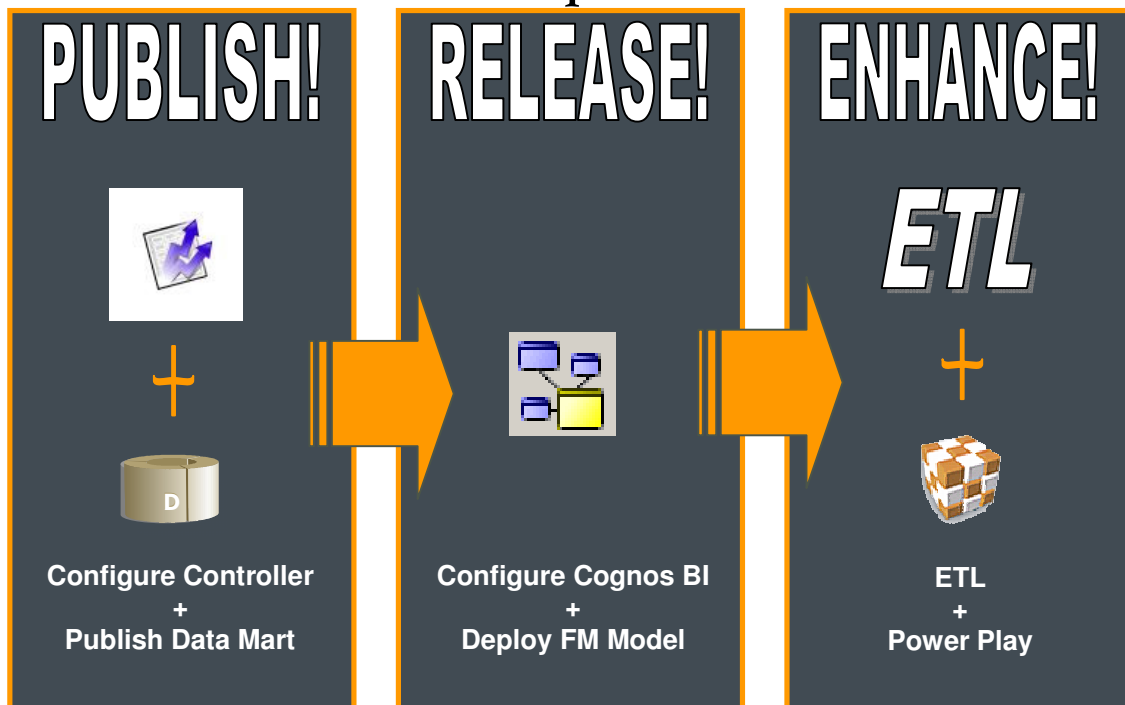
Something else to be aware of is that the MFR83 Blueprint is developed on top of MS SQL Server, the Blueprint is not tested on Oracle and might not work. Contact Cognos Innovation Center if you need an Oracle version of the MFR83 Blueprint.

MFR83 – Installation Guide

The *MFR83 Blueprint* is based on the Controller data mart and the FM model included on the installation DVD (CCR82_DM), but contains more logic and features, e.g. support for DMR and integration with Transformer.

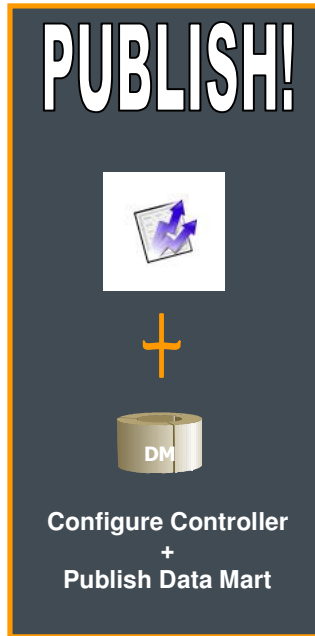
This document will guide you through the steps of adding the MFR Blueprint. Depending on your requirements you may need only to do the first step or couple of steps. However if you are planning to deliver professional financial reports using CognosBI we would recommend that you follow all three steps.

The 3 Step Process



- Step1:** **Create the Controller data mart used for reporting**
 Configure Controller to publish Data to the Data Mart
 Deals with automation of publishing the data mart
- Step 2:** **Adds the additional data sources to the Cognos8 Portal**
 Deploys the Framework Manager model
 Enables Reporting Using Relational and DMR views
- Step 3:** **Setup the ETL Process to enhance the data for use with PowerPlay**
 Load and alter the Power Play cubes
 Schedule the Enhancement and update of Power Play cubes

STEP 1 – Publish



Included in the Publish step is:

- Configure settings in Controller Configuration
- Publish data to the Controller data mart

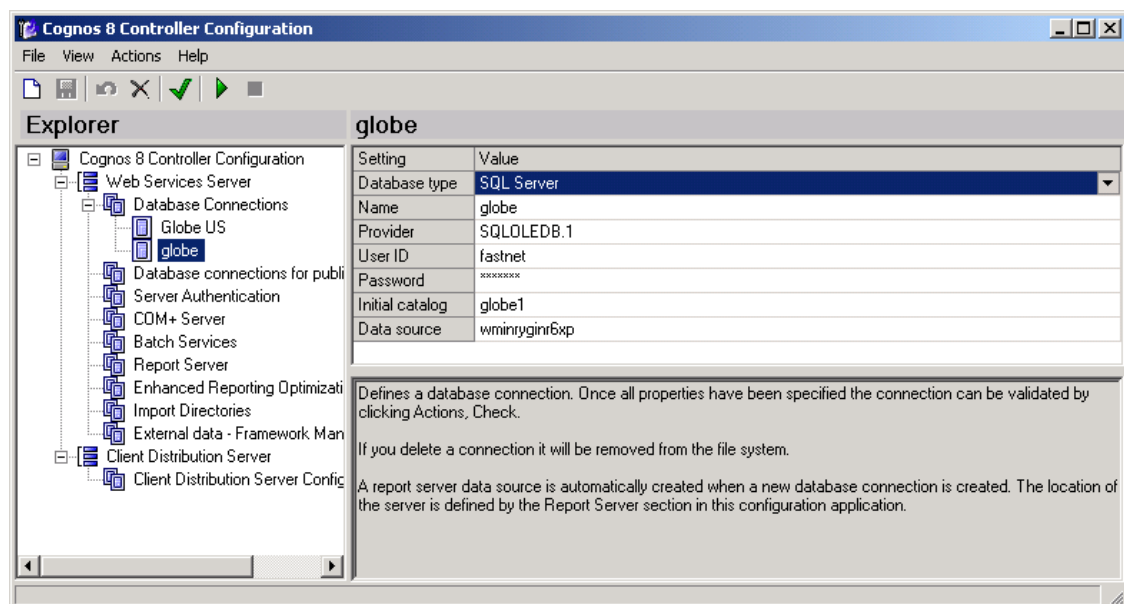
Create a New Database with fastnet as owner

Prior to starting the configuration a new database will need to be created to store the additional information. Appendix 1 covers how to create a blank database in sql server 2005.

Configure settings in Controller Configuration

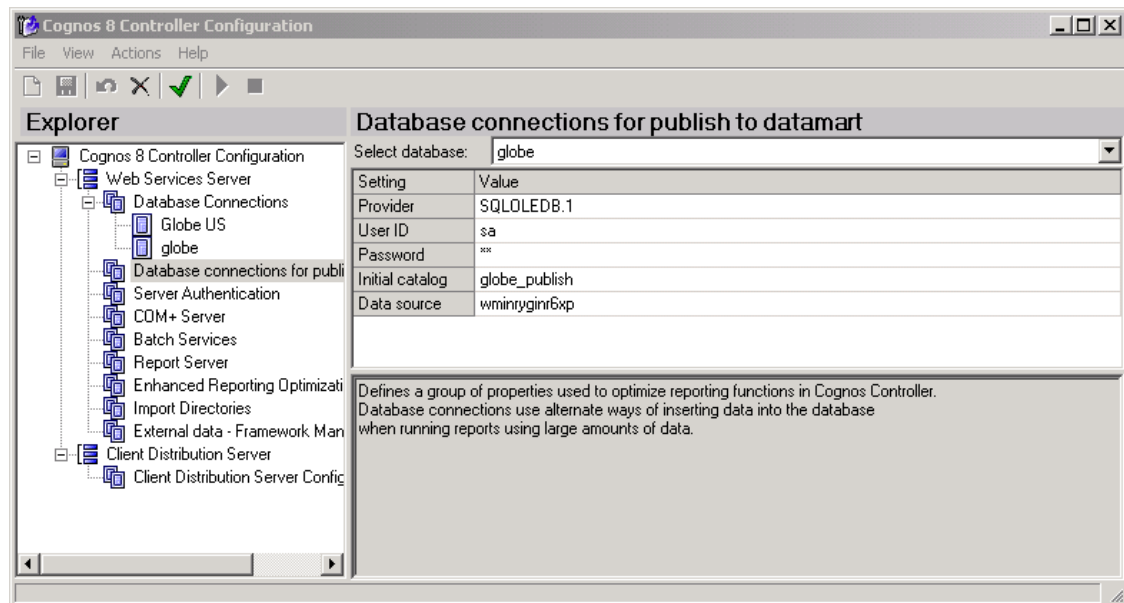
There are a few things that you have to configure before you can publish data to the Controller data mart:

1. Open up Cognos Controller Configuration.
2. Under Web Services Server > Database Connections for Publish to Data mart. Select the database connection for the controller database you would like to use as source. Configure the settings to point to your new data mart database. If this is on the same server as the controller database the settings should be the same as the Database Connections with the initial catalog being the new database name.
3. Save and test the connection. If the connection fails check the settings and the database security (per appendix 1).
4. Expand **Database Connections** and select the connection that you want to associate with the data mart.



5. Click on the green **Run** button to launch the **Database Conversion Utility**.

6. Click on the **DMConv** tab.
7. Select the correct udl file (default installation path C:\Program Files\cognos\c8\DMData\).
8. Click on **Run Steps** to create the Controller data mart, select **Yes** if asked if you want to create data mart tables. The actual dmversion should after conversion be set to **26**.
9. Click **Close**.
10. Click on **Database connections for publish to data mart**.



11. Select the correct database from the **Select Database** dropdown (the same one selected in step 2).
12. Click **Check** to test the connection.
13. Click **Save**.

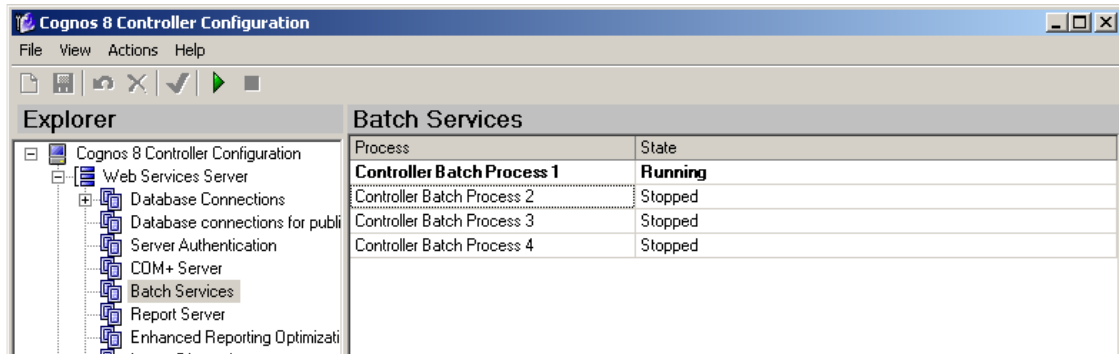
You must also configure Controller to use Cognos8 security.

1. Click **Server Authentication** in the Explorer pane.
2. Set **Server Authentication** to **Cognos8** (default is Native).

Please note: this step requires that you have defined and configured an external namespace, e.g. *Access Manager*, in *Cognos Configuration*.

The **Publish to Data Mart** job will run as a batch job and it's very important that at least one batch service is running.

1. Open Cognos Controller Configuration
2. Click Web Services Server > **Batch Services** in the Explorer pane.
3. If none of the processes are running start Controller batch Services 1
4. Select **Run** from the **Actions** menu, the State should change from **Stopped** to **Running**.



If Batch Processes were not running you will need to check that batch mode is set up within controller.

1. Launch **Cognos Controller**.
2. Go to Maintenance > Configuration > General > Server preferences tab. Ensure that there is a variable name 'BATCHMODE' with a variable value of 'TRUE'.

If this is the first time you have published in this controller database you will need to optimize your database. This creates the correct control tables.

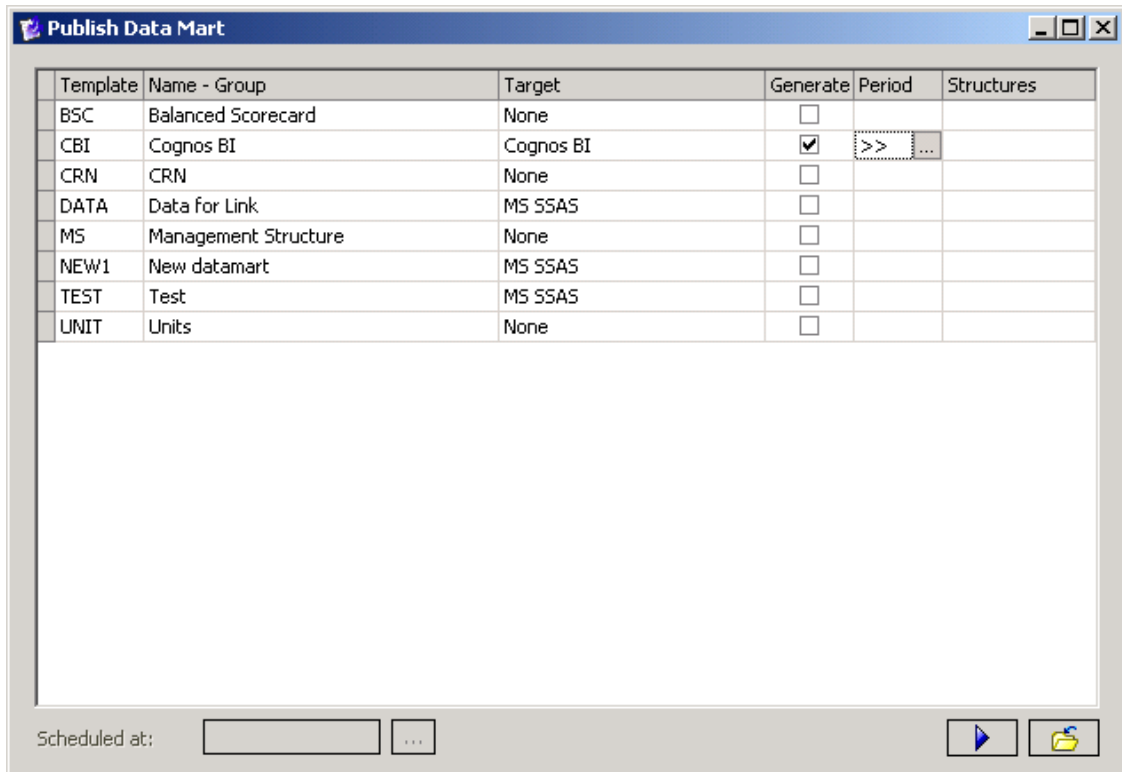
1. Launch **Cognos Controller**.
2. Go to Maintenance > User > Single Mode
3. Go to Maintenance > Database > optimize
4. Select Rebuild indexes and run.

Publish data to the Controller data mart

You have now configured Controller so that data can be published to the Controller Data Mart, you have also configured Controller to use Cognos8 security. The next step is to publish data to the Controller data mart.

1. Launch **Cognos Controller**.
2. Open Transfer/**Publish Data and Structures/Define Data Mart...**
3. Click **New** to define a new Template.
4. Select **Cognos BI** as **Additional Target**.
5. Define which companies, forms, actualities, currencies, etc to include. This will need some experiment to get the best solution for the customer. It is often the case that a form view is easier to use.
6. Click **Save**.

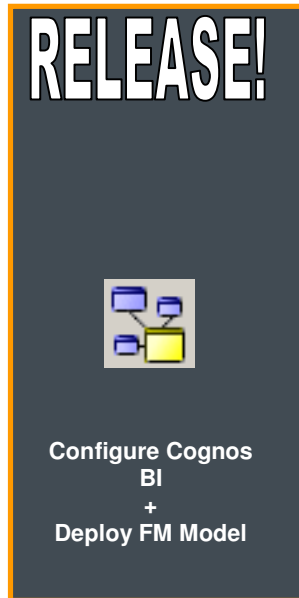
7. Open **Transfer/Publish Data and Structures/Publish Data Mart...**
8. Tick the **Generate** checkbox for the template you want to publish data for.
9. Select which periods to publish data for.
10. Click **Run**, a batch job will now been added to the Controller batch queue and you can monitor the process from **Maintain/Batch Queue/Manage...**



* If you have issues with the Publish of the data mart please use the following resources:

- 'File name or class name not found during Automation operation' - Go to appendix 2.
- Invalid object name 'dmversion' – Go to Appendix 3
- Invalid object name 'dmversion' – Ensure that sqloled.1 setting in database connections and database connections for publish to data mart is in the same case.
- If this is the first time you have published data in a data mart you may need to optimize the database for data to appear.
- If you are using a demo database or database created on a different server you may need to delete existing definitions and recreate them.
- Ensure that TESTMODE in the server preferences is not set to yes.
- 'No values found for Template' – Go to Appendix 4

STEP 2 – Release



Create Data Source Connections in Cognos Connection

Data has now been published to the Controller data mart. The next step is to define two new data sources in **Cognos Connection** (these will be used by framework manager). The first one points to the Controller data mart:

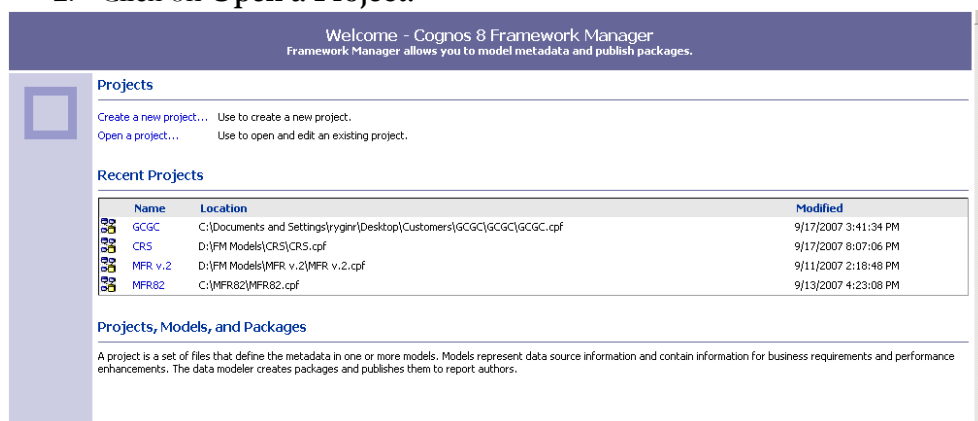
1. Launch **Cognos Connection**.
2. Select **Launch/Cognos Administration**
3. Click the **Configuration** tab
4. Select the **Data Sources** tab.
5. Click **New Data Source**.
6. Define **CCR82_DM** as **Name** and click **next**.
7. Select the correct database type, click **next**.
8. Specify the correct **connection string** and **signon**, the data source should point to the Controller data mart.
9. Click **Test the connection** to verify that it works.
10. Click **Finish**

Repeat above steps to create a data source that points to the Controller database. Specify **Cognos Controller** as name.

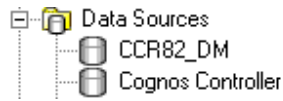
You can use other names of the data source connections, but above names are used in the FM model, i.e. you have to modify the FM model if you are using other names.

Modify the MFR83 FM model

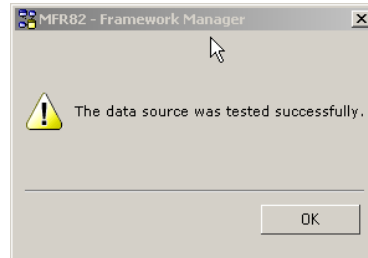
1. Launch **Cognos Framework Manager**.
2. Click on **Open a Project**.



3. Browse for the MFR83 Blueprint model (**MFR83.cpf**).
4. Expand the **Data Sources** tab.



5. Right-click **CCR82_DM** and select **Test**. You should receive a message like this:



6. Repeat above step for the **Cognos Controller** data source.

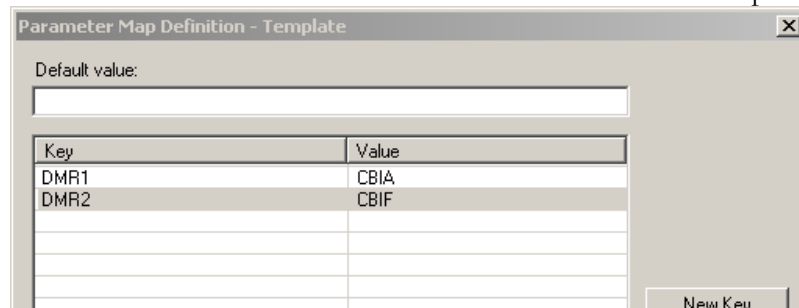
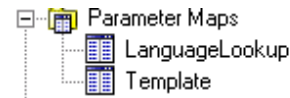
The next step is to verify that your query subjects actually retrieve data.

1. Expand the **IMPORT VIEW**
2. Right-click **DMACT** and select **Test**

If working, this will show you **Actualities** that you have included in the data mart.

The MFR83 FM model contain support for Dimensionally Modeled Relational data (DMR) and the DMR layer is filtered by **Template**, i.e. only one Template can be used in each DMR view.

1. Expand the **Parameter Maps** folder
2. Select **Template**, right-click and select **Edit Definition...**
3. Click on **DMR** and change **Value** to the template you want to use in the **DMR** layer (This is the code in the define data mart screen – 'CBP' in the example used above)



You can now make sure the correct filter is used.

1. Expand **MODEL VIEWS/DMR**.
2. Select **ACCOUNT**, right-click and select **Test**.
3. You should receive data; you can also scroll to the right and verify that **TEMPLATE ID** is the correct one, i.e. the template you defined in the parameter map.

Publish Packages

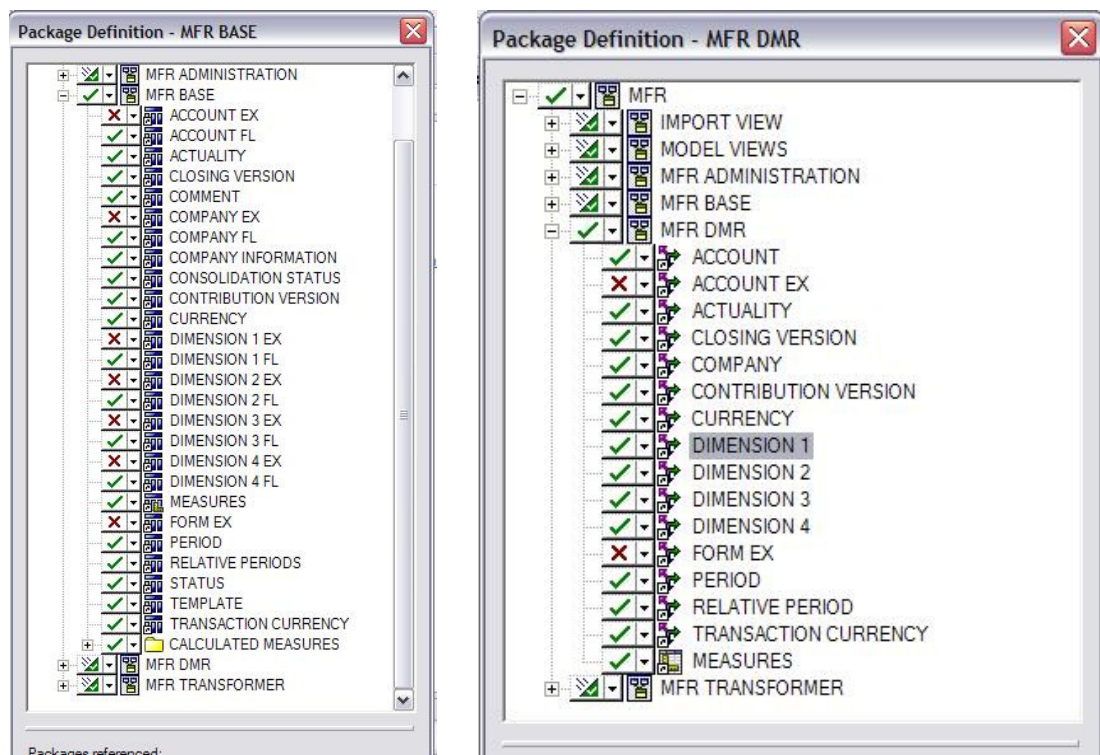
There are four default packages included in the MFR83 FM model:

- **MFR ADMINISTRATION**, used to automate the publishing process. Tables created in stage 3
- **MFR BASE**, the default relational layer
- **MFR DMR**, the DMR layer
- **MFR TRANSFORMER**, this layer that will be used in Transformer. Tables populated in stage 3.

At this point only **MFR BASE & MFR DMR** can be published. **MFR TRANSFORMED & MFR ADMINISTRATION** are published in step 3

The MFR83 DMR & BASE packages will contain two query subjects, **ACCOUNT EXPANDED** and **FORM EXPANDED**, that will not work based on a default installation, if you are not intending to complete stage 3 then to deploy MFR Base & MFR DMR you will need to remove the expanded items (ending EX) within the package:

1. Expand the **Packages** folder
2. Select the package you want to publish, e.g. **MFR Base** or **MFR DMR**, right-click and select **Edit Definition**. Then make the amendments shown below:



You must publish the package before it could be used as source in Cognos Connection.

3. Expand the **Packages** folder

4. Select the package you want to publish, e.g. **MFR DMR**, right-click and select **Publish Packages...**

At this point only the MFR DMR and MFR Base can be published. Once Stage 3 is complete then the other packages can be deployed.

Within Cognos Connection you can now select one of the data sources published and create a report.



STEP 3 – Enhance



This step is broken into 3 sections:

- Enhance the account and form structures with the use of Extract Transform & Load (ETL) techniques.
- Create a PowerPlay Cube for fast multidimensional reporting
- Automate the ETL and update process

ENHANCE:

The *MFR83 Blueprint* contains some extra features that require a special installation, e.g. an expanded account/form hierarchy and a way to automate the creation of these structures and the cube.

Expanded Account/form hierarchy

One of the new features in the Controller 8.2 data mart is that hierarchies are expanded, i.e., the number of columns in the dimension table is dependent on how many levels there is in the hierarchy. Controller 8.1 stores hierarchies as parent/child instead, that is, only two columns describe the hierarchy. An

expanded hierarchy is easier to use as you can drill up and down in the hierarchy, a parent/child allow only drilling one level (parent to child).

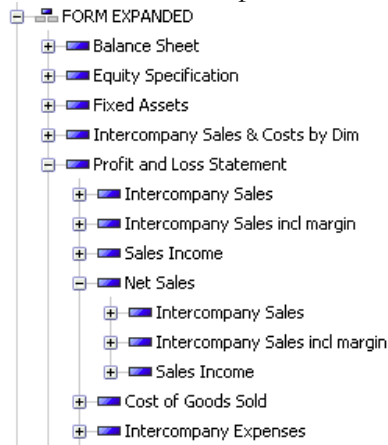
A limitation is that the account/form hierarchy is not expanded in the default data mart. This must be done as a manual step after data has been published to the Controller data mart.

The *MFR83 Blueprint* contains two different versions of an expanded account hierarchy: one is based on a template that publishes accounts and the other is based on a template that publishes forms. The account-based alternative expands all the accounts that are selected in Publish Define, while the form-based alternative lists all selected forms, but also expands all accounts that are included in the form. This could be very useful when you want to see the details to a summation in a form.

Here is an example of the expanded account-based hierarchy:



And this is an example of a form-based hierarchy:



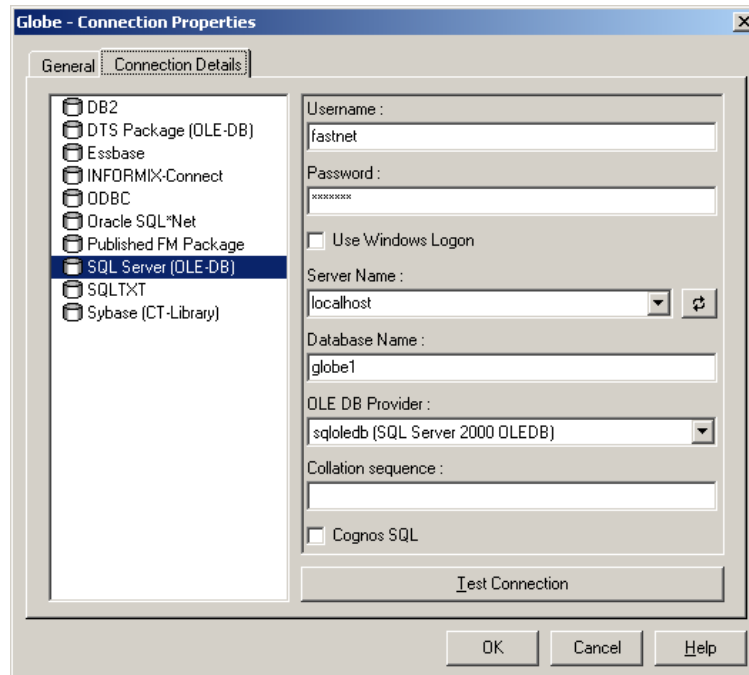
Expand the hierarchies:

There are three ways to expand the account/form hierarchy:

- Cognos Data Manager (See Below)
- Microsoft SQL Server 2000 DTS (go to Page 18)
- Microsoft SQL Server 2005 SSIS (go to Page 21)

Cognos Data Manager:

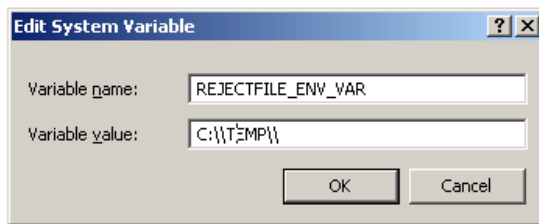
1. Launch **Cognos Data Manager** and restore the **MFR83** catalog (MFR83.ctg) included in the *MFR83 Blueprint*.
2. Expand the **Connections** folder.
3. Select the **CONTROLLER** data source and verify that it connects to the correct Controller database. This data source reads the “live” Controller database.



4. Do the same for the **CCR82_DM** connection. This data source should be connected to the Controller data mart.
5. Locate the **GENERATE_MFRACCOUNT** Job stream.
6. Select **Actions/Execute...** The account hierarchy will be expanded and stored in the **MFRACCOUNT** table in the Controller data mart. **
7. Repeat above step for the **GENERATE_MFRFORM** Job stream. This job stream will expand the account hierarchy based on forms and store the data in the **MFRFORM** table. **

** You might receive an error message regarding a missing **System Variable** called **REJECTFILE_ENV_VAR**. Do like this to add it:

1. Open **System Properties** (e.g. My Computer, right-click and select Properties).
2. Select the **Advanced** tab.
3. Click **Environment Variables**.
4. Click **New** to create a new **System Variable**.
5. Specify **REJECTFILE_ENV_VAR** as **Variable name**.
6. Specify **C:\\TEMP** as **Variable value**.
7. Reboot the machine.



Test the expanded Hierarchies:

You can now open Framework Manager and test the **ACCOUNT EXPANDED** and **FORM EXPANDED** query subjects. The result of this test is of course dependent on how you have published data to the data mart and defined the Template parameter map.

Next step is to publish the **DMR** and/or **BASE** package. **ACCOUNT EXPANDED** and **FORM EXPANDED** can now be used as source in a report.

In step 2 the original framework manager packages the expanded objects were removed to enable publishing. However now the tables have been created these can be re-enabled and the packages re published.

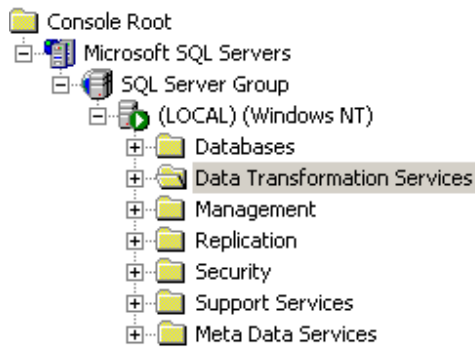
Enable publishing. However now the tables have been created these can be re-enabled and the packages re published.

Now go to Cognos Transformer section (page 23).

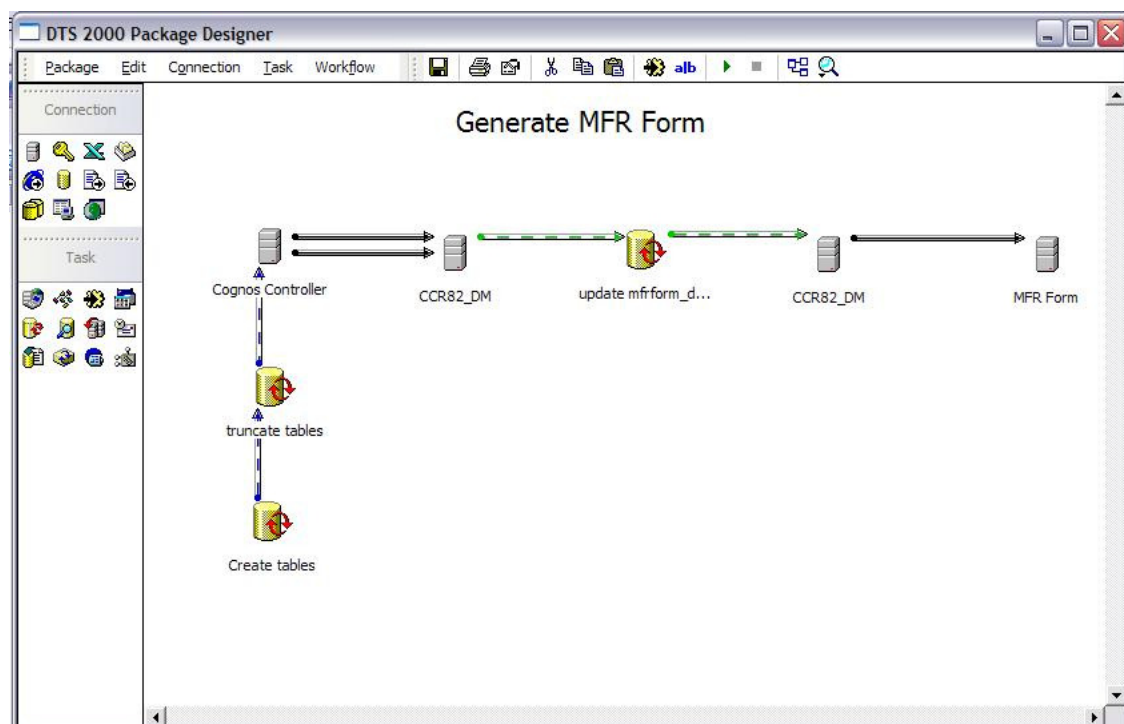
Microsoft Sql Server 2000 DTS:

You can also expand the account/form hierarchy by using functionality in MS SQL Server DTS (Data Transformation Services)

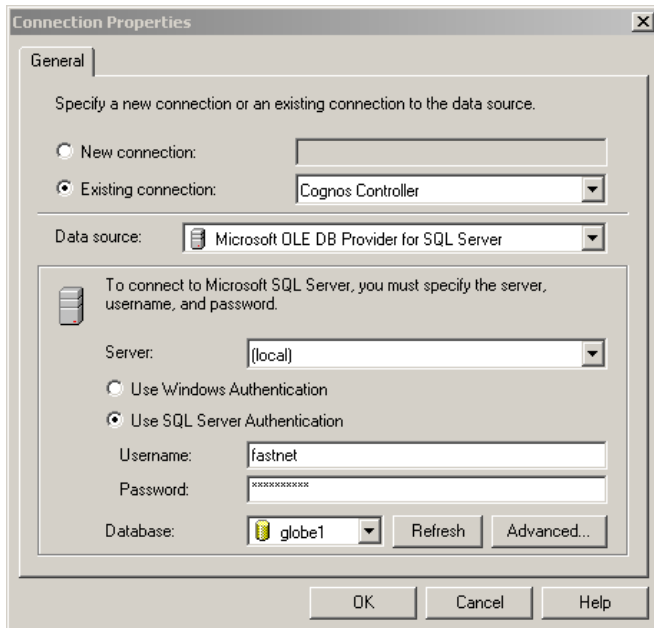
1. Launch **MS SQL Server Enterprise Manager**
2. Expand the server and select the **Data Transformation Services** folder



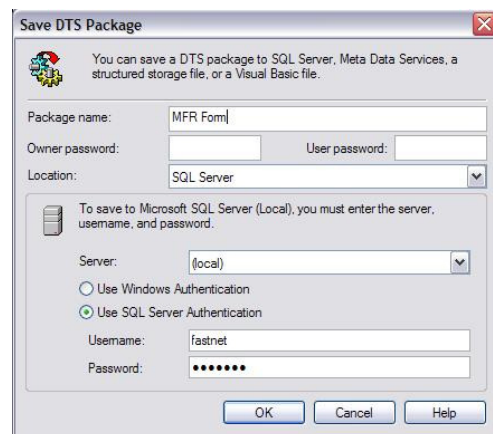
3. Right-click and select **Open Package...**
4. Browse for **MFR Form.dts**
5. Click **Open**



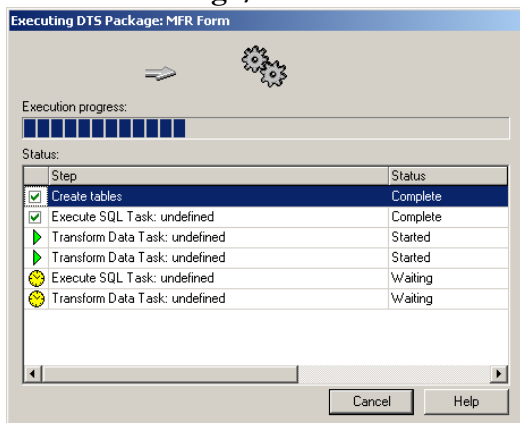
6. Double-click the **Cognos Controller** connection
7. Verify that the properties to the Controller database are correct



8. Repeat above step for the **CCR82_DM** and **MFR FORM** connection. Both these connections should point to the Controller data mart.
9. **Save As** the package and change the location to the SQL Server whilst ensuring the username is fastnet. This allows the package to be automated later in the process.



10. Select **Package/Execute** to execute the package



Note, the 'end result' of this job, i.e. what is stored in the MFRFORM table in the database is dependent on how the Template is defined. This job will only expand the hierarchy if Forms are exported, it will not work if the Template is defined to publish Accounts.

The form hierarchy has now been expanded and is stored in the **MFRFORM** table in the Controller database.

11. Repeat above steps to expand the **account hierarchy**. The expanded account hierarchy is stored in the MFRACCOUNT table

Test the expanded Hierarchies:

You can now open Framework Manager and test the **ACCOUNT EXPANDED** and **FORM EXPANDED** query subjects. The result of this test is of course dependent on how you have published data to the data mart and defined the Template parameter map.

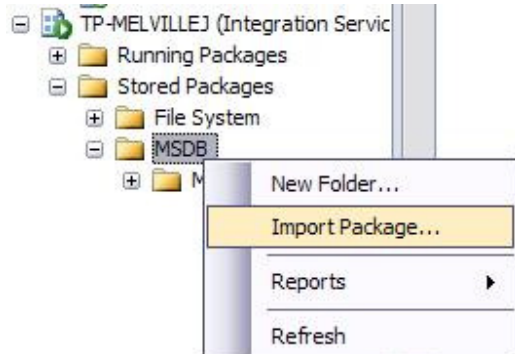
Next step is to publish the **DMR** and/or **BASE** package. **ACCOUNT EXPANDED** and **FORM EXPANDED** can now be used as source in a report.

In step 2 the original framework manager packages the expanded objects were removed to enable publishing. However now the tables have been created these can be re-enabled and the packages re published.

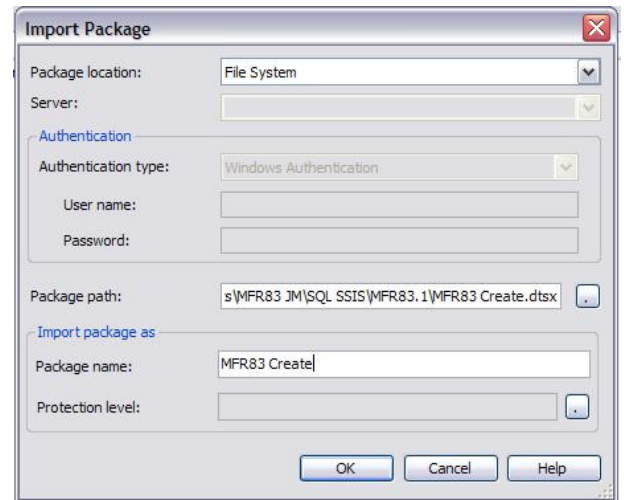
Now go to Cognos Transformer section (page 23).

Microsoft SQL Server 2005 SSIS:

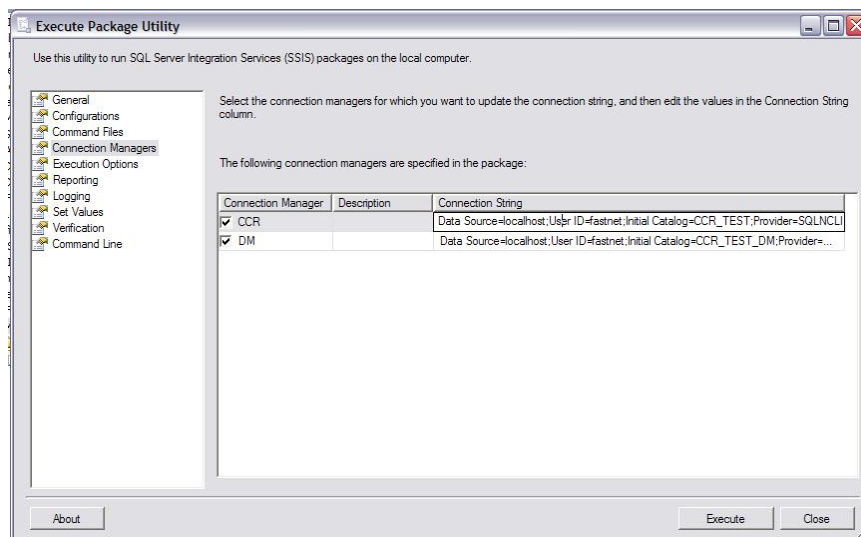
You can also expand the account/form hierarchy by using functionality in MS SQL Server 2005 Integration Services (SSIS):



1. Launch **MS SQL Server Management Studio**
2. Connect to **Integration Services**
3. Navigate down to **Stored Procedures > MSDB**
4. Right click over MSDB and go to **Import Package**



5. Select 'File System' under Package Location
6. Package Path find MFR83 Create.dtsx
7. Leave the package name as default.
8. Select OK.
9. Right click over the package 'MFR83 Create' and go to 'Run Package'
10. Change the connection to your Data Mart and controller Database settings.
11. Execute the Package



This will create and populate the MFRFORM and MFRACCOUNT tables.

Test the expanded Hierarchies:

You can now open Framework Manager and test the **ACCOUNT EXPANDED** and **FORM EXPANDED** query subjects. The result of this test is of course dependent on how you have published data to the data mart and defined the Template parameter map.

Next step is to publish the **DMR** and/or **BASE** package. **ACCOUNT EXPANDED** and **FORM EXPANDED** can now be used as source in a report.

In step 2 the original framework manager packages the expanded objects were removed to enable publishing. However now the tables have been created these can be re-enabled and the packages re published.

Cognos Transformer

Another new feature in the *MFR83 Blueprint* is the support for **Cognos Transformer**. The support means that a PowerPlay Transformer cube (Powercube) can be created on top of the Controller data mart. The new PowerPlay cube will require the implementation of the MFR Transformer model which requires these steps:

- Publish the **MFR TRANSFORMER** package
- Modify included Transformer model

Publish the MFR TRANSFORMER package

The default Transformer model included in the *MFR83 Blueprint* uses query subjects in the MFR TRANSFORMER package as data source

The **MFR TRANSFORMER** is filtered, just like **MFR DMR**, by Template.

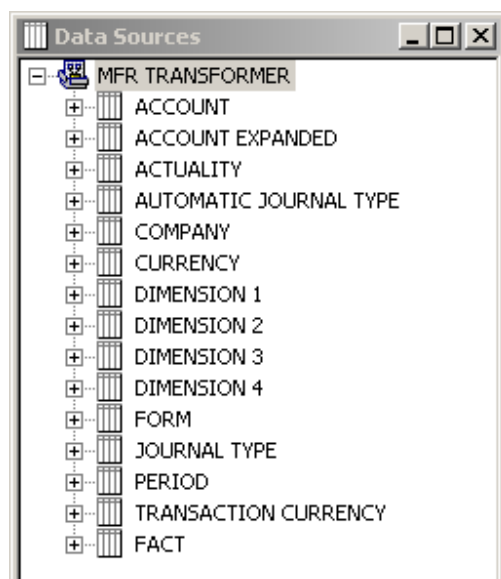
1. Launch **Framework Manager** and open the **MFR83 FM model**
2. Locate the **MFR TRANSFORMER** business view
3. Open one of the query subjects and secure that the **Template filter** points to the correct **Template** i.e. ‘**CBI**’

You can rather easily create more “Transformer views”, just copy the existing one and make sure that the filter points to the correct template.

1. Publish the **MFR TRANSFORMER** package

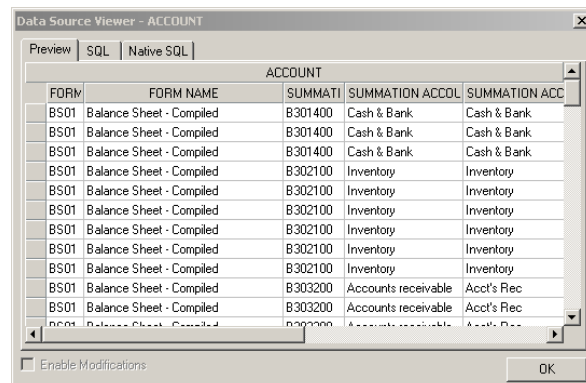
Modify included Transformer model

A default Transformer model is included in the *MFR83 Blueprint*, and you have to secure that the data source connection works.

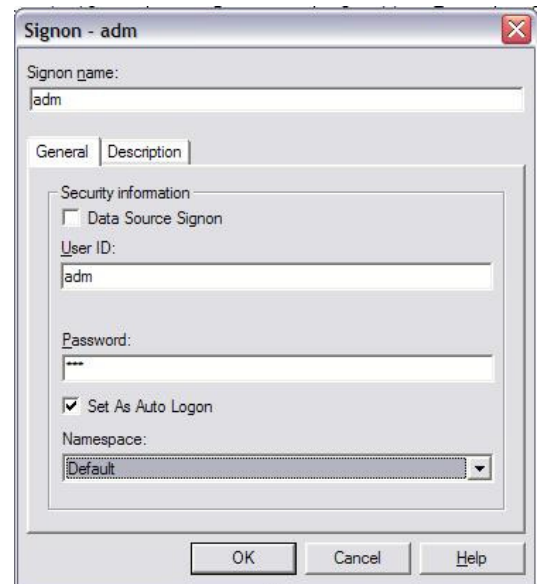


1. Launch **PowerPlay Transformer**
2. Select **File/Open** and browse for the Transformer model (**MFR83.pyj**) included in the MFR83 Blueprint.
3. Locate the **Data Sources** view
4. Right-click the **ACCOUNT** dimension
5. Select **Data Source Viewer**

This step is to verify that the data source connection works

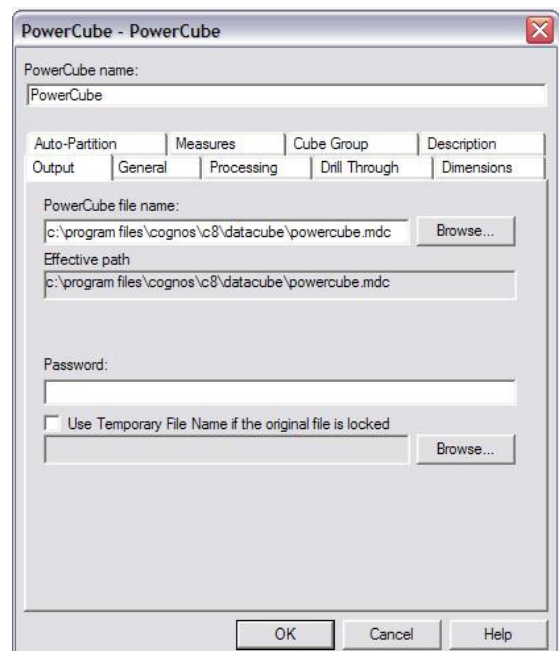


1. If you have published a data mart template that uses forms you could delete the '**account expanded**' dimension by right clicking on 'account expanded' in the dimension map and go to '**delete dimension**'. If you published an account template you can delete the '**form dimension**'.
2. Click **Check Model** to verify that there are no potential problems in the model
3. To automate the update of the cube we need to give the model signons. To do this click View > Signons to bring up the signon window.
4. Then right click in the signons window and go to insert signon. If the signon window is not present then go to view > Signons
- 5.
6. You will need to login as an administrator
7. Add a new signon name.
8. Ensure 'Data Source Signon' is un ticked.
9. Ensure **Set As Auto logon** is ticked.
10. Click **OK**



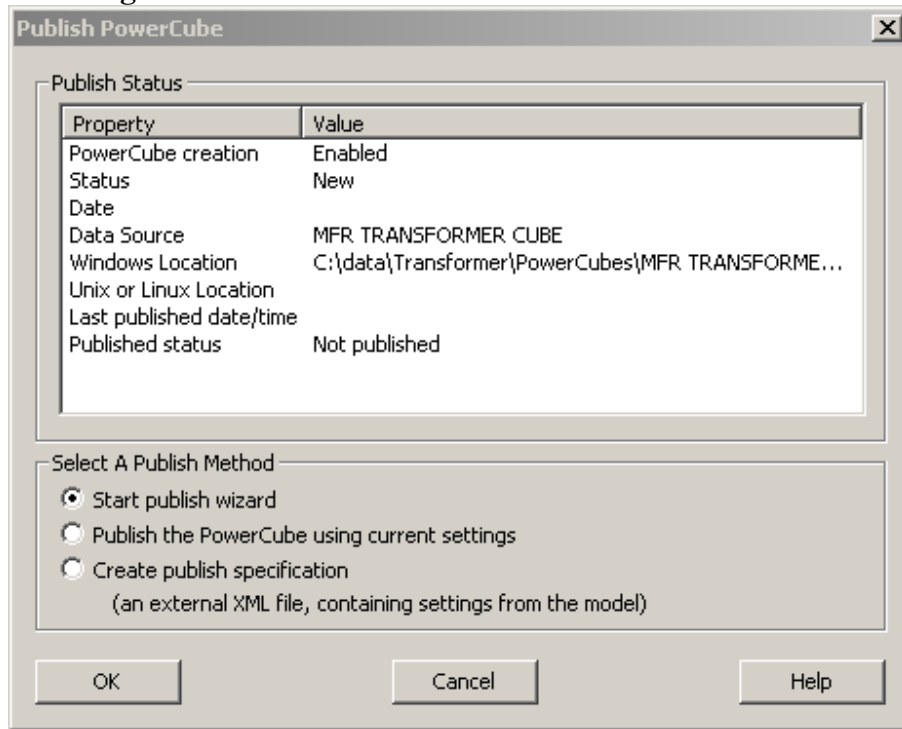
Next we need to create a powercube

1. Create the powercube by right clicking in the powercube box and going to insert power cube.
2. Give the Cube a name and place it in a location where there are no sub directories or files. Later in the automation process the copy functionality will copy all items in this directory to the live reporting area.
3. '**Save as**' an mdl file in a common data directory.
4. Select **Run/Create Powercubes** to create the cube

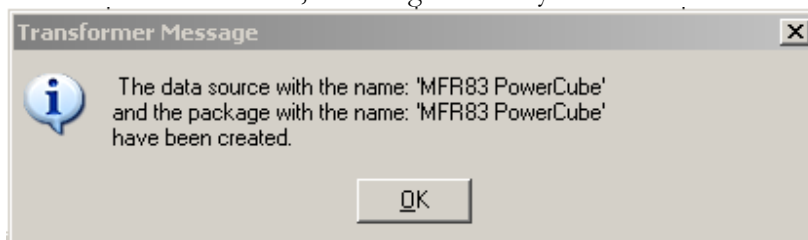


The Transformer model is now updated and a Powercube is created. The next step is to publish the cube to **Cognos Connection**.

1. Right-click the cube and select **Publish PowerCube as Data Source and Package...**



2. Click **OK** to start the publish wizard
3. Click **Next**
4. Specify a name and location for the new data source
5. Click **Next**
6. Click **Test the Connection** to verify that the connection to the cube works
7. Click **Finish** to create the new data source
8. Click **OK** to create a new package
9. Specify a name and location for the new package
10. Click **Finish**, a message will tell you that the data source and package was created



The cube has now been published to Cognos Connection and could be used as source in reports.

Please note that User Rights in Controller are not defined in the cube, which is something that must be done manually.

Transformer Automation:

Prior to setting up the trigger to automatically update the cube an infrastructure needs to be put into place to enable the automation. To update the cube it must not be locked which occurs when a cube is being read by Cognos BI. To avoid this it is best practice to have a copy of a cube that can be updated and copy this to a location to be read.

Implementing PowerPlay Automation:

Copy the PowerPlayCubeBuilder directory onto the server. For the examples in the blueprint we will use a new directory called “C:\Program Files\cognos\c8\PowerPlayCubeBuilder”. The purpose of this is as follows:

1. Refresh the cube
2. Copy the cube to a live directory where the cube is not locked (files are locked when being read by CognosBI).
3. Change the Cognos connections to look at the new cube.

As a cube is locked when being read and can not be refreshed or replaced with another cube we have one directory where the cube that is refreshed by Transformer is stored. We then have 2 directories where the cube can be read i.e. Live1 or Live 2. If Live1 is being used the cube file that is not read is refreshed and then copied to Live2 and Cognos connections is changed to point at Live2. Next time the cube is refreshed it is copied to Live1 (that will no longer be locked) and Cognos Connections is changed to point to Live2.

Now follow the configuration steps:

1. Right click on the **run.bat** file and go to **Edit**
2. Ensure that the java directory exists:


```
if "%JAVA_HOME%" == "" set JAVA_HOME=C:\Program
Files\Cognos\c8\bin\jre\1.5.0
```
3. Right click on run1.bat & run2.bat and change the location to the location of the PowerPlayCubeBuilder directory.
4. Open **C8cubebuilder.properties** in note pad and check the following entries:
 - a. `c8.url=` **Cognos 8 directory**
 - b. `c8.ns=` **Cognos 8 namespace**
 - c. `c8.user=` **Cognos 8 User**
 - d. `c8.pwd=` **Cognos 8 Users Password**
 - e. `c8.cogtr=` **The directory where the cogtr.exe file is stored (default =C:\Program Files\cognos\c8\bin\cogtr.exe)**
5. Go to the examples directory and open up Live.xml. Check the following directories:
 - a. Check the model build directory to ensure that it represents the directory where the ‘mdl’ model is stored i.e.


```
<modelbuild modelpath="C:\Documents and
Settings\xxxxxx\My Documents\Transformer\MFRTransformer.mdl"
enabled="true" stoponerror="true">
```
 - b. Check the login details as set up in the PowerPlay cube i.e.


```
<arg>-IADM=ADM/kbs</arg>
```

- c. Change the Source director of the originally created cube

```
<filecopy src="C:\Documents and Settings\xxxxx\My
Documents\Transformer\PowerCubes" delete="false" stoponerror="true">
```

- d. Change the destination location to the PowerPlayCubeBuilder\Live1 & Live2 location

```
<dest id="1" dir="C:\Program Files\cognos\c8\PowerPlay\Live1"/>
<dest id="2" dir="C:\Program Files\cognos\c8\PowerPlay\Live2"/>
```

- e. When the cube is published a data source name is created. Change the file to the new data source name.

```
<connswap name="MFR83" stoponerror="true">
```

- f. Change the destination & file names if different to the location where the PowerPlayCubeBuilder\Live1 & Live2 location.

```
<cube src id="1" winloc="C:\Program
Files\cognos\c8\PowerPlay\Live1\MFR83.mdc"/>
<cube src id="2" winloc="C:\Program
Files\cognos\c8\PowerPlay\Live2\MFR83.mdc"/>
```

To test the automation open up the command prompt **Start > Run > cmd:**

1. Navigate to the directory where the run.bat file is held i.e.
Cd C:\Program files\cognos\c8\PowerPlayCube Builder
2. Run the **run.bat** command with the correct variables:
 - a. **-f examples/live.xl** The directory of the xml file
 - b. **-i 1** This is the version that will send the file to either Live1 or Live2 and then use this to change the connection to point to the new directory.

If this fails it may be that the wrong version of Sun Java is loaded. Run the “jdk-1_5_0_16-windows-i586-p.exe” file stored in the PowerPlayCubeBuilder directory and ensure that the run.bat file is pointing to the correct java home directory.

MFR Trigger

Expanding the account/form hierarchy and updating the power cube is a manual step that has to be executed after the data has been published to the Controller data mart. Included in the *MFR83 Blueprint* is an example of how to automate this process. The basis is a trigger that “knows” when data has been published to the data mart, an Event will then be triggered.

The methods used for the trigger are:

- Cognos Data Manager (See below)
- Microsoft SQL Server 2000 DTS (Page 34)
- Microsoft SQL Server 2005 SSIS (Page 37)

Cognos Data Manager:

A new table is required to keep track of data has been published to the data mart.

Create the MFRPUBLOG table

1. Launch **Cognos Data Manager** and open the **MFR83** catalog.
2. Execute the **CREATE_MFRPUBLOG** jobstream.

This job will create a new table, **MFRPUBLOG** which will be used to keep track of when data has been published to the data mart. Another job, **UPDATE_MFRPUBLOG**, will be used later to update the **MFR_ID** in **MFRPUBLOG** to enable you to trigger the update on changes.

Update the **UPDATE_POWERCUBE** pointing the sql script to look at the new PowerPlayCubeBuilder Directory.

1. Right click on ‘generate cubes’ task
 2. Go to properties
 3. Go to the SQL tab
 4. change the following lines to point at the new location:
- ```
exec master.dbo.xp_cmdshell 'C:\Program
Files\cognos\c8\PowerPlayCubeBuilder\Run1.bat '
&
exec master.dbo.xp_cmdshell 'C:\Program
Files\cognos\c8\PowerPlayCubeBuilder\Run2.bat '
```

#### Publish Data Movement Tasks

The jobs that update the log and the jobs that expand the account/form hierarchy must be published as **Data Movement tasks** so that they can be referenced and executed from Cognos Connection.

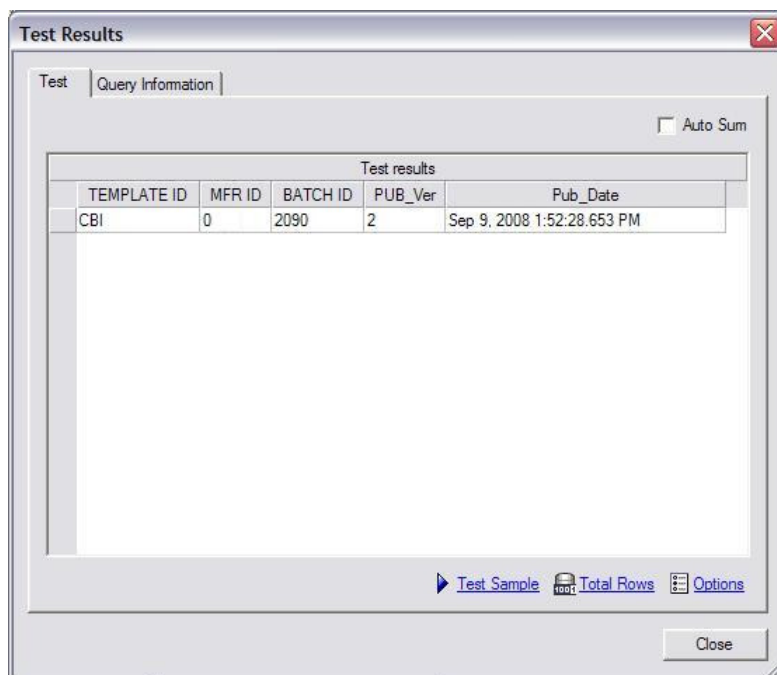
1. Select **Action/Publish Data Movement Task**.
2. Select **UPDATE\_MFRPUBLOG**
3. Click **OK**
4. Repeat above steps for **UPDATE\_POWERCUBE**, **GENERATE\_MFRACCOUNT** and **GENERATE\_MFRFORM**

**UPDATE\_MFRPUBLOG, GENERATE\_MFRACCOUNT, GENERATE\_MFRFORM and GENERATE\_POWERCUBE** will now be included in a folder called **dataMovementTasks** in Cognos Connection.

### **Publish the MFR ADMINISTRATION package**

The table that you just created, **MFRPUBLOG**, is included in the MFR83 FM model, and you have to publish the **MFR ADMINISTRATION** package before they could be used in Cognos Connection.

1. Launch **Framework Manager** and open the **MFR83** FM model.
2. Expand **MFR/MODEL VIEWS/ADMINISTRATION** and select the **MFR PUBLISH LOG** query subject.
3. Right-click and select **Test**. You should see something like below, the **MFR ID** should be set to 0.



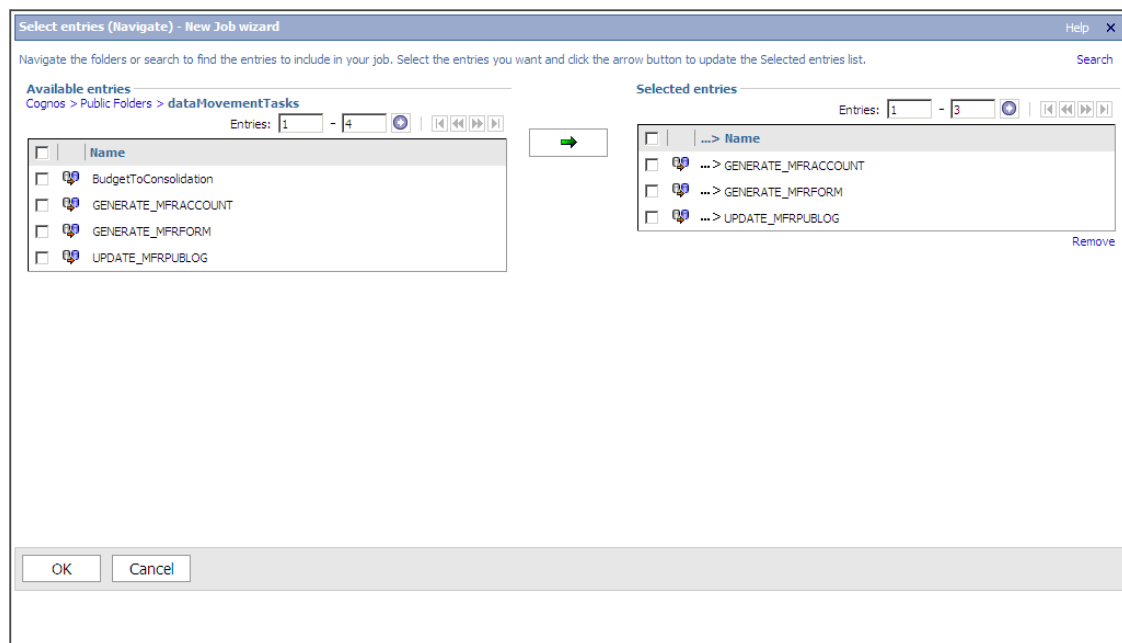
4. Expand the **Packages** folder.
5. Select **MFR ADMINISTRATION**, right-click and select **Publish**.

The **MFR ADMINISTRATION** package has now been published and could be used as source in Cognos Connection.

## Create a Job

A new Job will be created that executes the different tasks in sequence, i.e. a job will not start to run until the first job is complete. The final step is to update the MFR ID in the **MFRPUBLOG** table, this means that the event will not be triggered until the next time data has been published to the Controller data mart.

1. In **Cognos Connection**, click **New Job**.
2. Give the Job a name.
3. Specify in which folder to store the Job.
4. Click **Next**.
5. Click **Add** to add tasks to the Job.
6. Browse for the **dataMovementTasks** folder.
7. Tick the checkbox for the jobstreams that you want to include, e.g. **GENERATE\_MFRACCOUNT**, **GENERATE\_MFRFORM**, **UPDATE\_POWERCUBE** and **UPDATE\_MFRPUBLOG**. Verify that the tasks are ordered in this order.



6. Click the green arrow to move these tasks to the **selected entries** frame.
7. Click **OK**.
8. Select In sequence, this means that each job will run individually.
9. Click **Next**.
10. You can now choose if you want to save the job and run it, save and schedule or just save.
11. Click **Finish**.

## Create Event

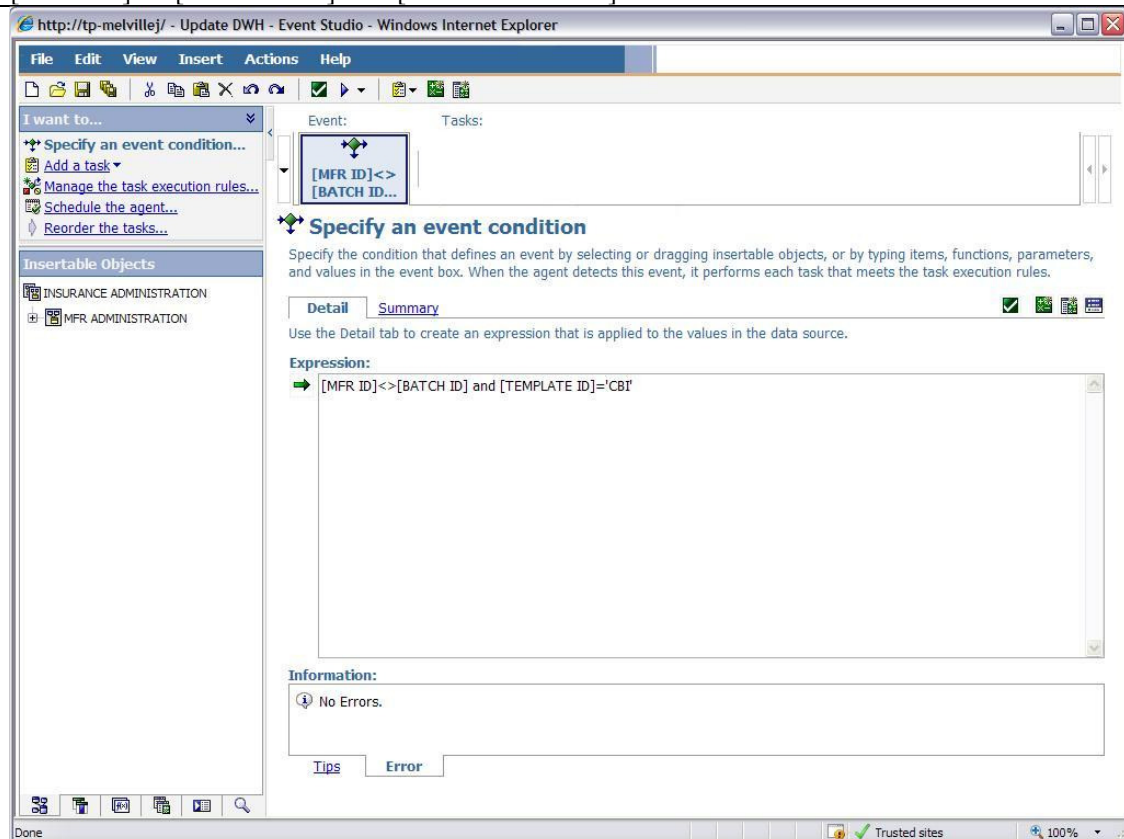
Event Studio will be used to trigger the generation of the security files, expansion of the account hierarchy and the creation of the cube after that data has been published to the Controller data mart. Data from the **MFR PUBLISH LOG** query subject will be used to determine when data has been published. I.e. the event will be triggered if MFR ID is different than BATCH ID. The MFR ID will be updated if the event is triggered.

1. In **Cognos Connection**: Launch **Event Studio**.
2. Select **MFR ADMINISTRATION** as the package to use.
3. Expand **MFR ADMINISTRATION/MFR PUBLISH LOG**.
4. Select **MFR ID** and drag it into the expression area.
5. Do the same for **BATCH ID** and define the expression such as:

MFR ID <> BATCH ID

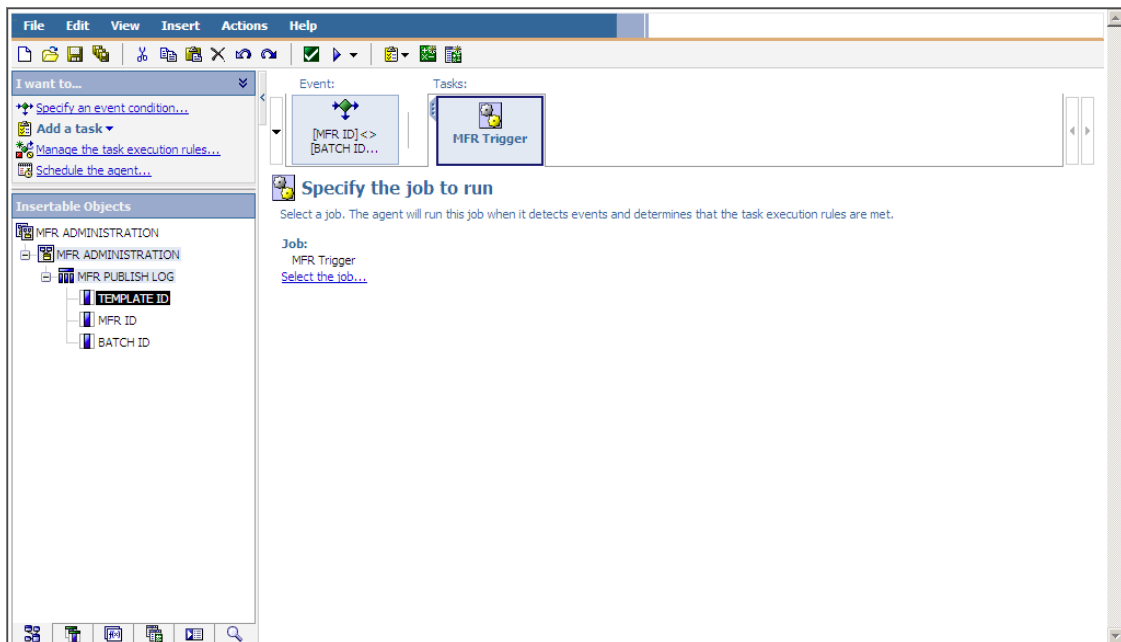
6. You can also filter the event by Template id, i.e. it should only be applicable for one specific Template

[MFR ID]<>[BATCH ID] and [TEMPLATE ID]='CBI'



Above event will return True if data has been published to the data mart for the specified Template, in this case MFRD.

7. Select **Add a task/Run a Job...**
8. Click **Select the job...** and browse for the job you created above.
9. **Save** the event.



You can also include that e.g. a News Item will be published after that the Transformer cube is generated, but that is not included in this example.

The trigger event consists of two steps:

10. Check the MFR ID and the BATCH ID, the trigger job will only run if there is a difference between MFR ID and BATCH ID, i.e. data has been published to the Controller data mart.

### Test the automated process

1. In **Cognos Controller**: publish data to the data mart. Verify that the job runs successfully. There should now be a difference between the MFR ID and the BATCH ID.
2. Start the newly created Event, you can monitor it from the **Status** screen (Launch/Cognos Administration).
3. You should now see that the account/form hierarchy has been expanded
4. Run the job again, the mfr\_id is not equal to the batch\_id and the job will not be executed.

### Schedule the event

Above event will trigger the job, and we want this event to be triggered directly after that data has been published to the Controller data mart. One way to do this is to schedule the event to run e.g. every minute.

1. In **Cognos Connection**: locate the event you just created
2. Click the **Schedule** button
3. Specify the frequency you want to run the event

**Schedule - MFR\_TRIGGER\_PUBLISH**

Schedule the entry to run at a recurring date and time. You can disable the schedule without losing any of its details.

☐ Disable the schedule

**Frequency:**  
Select the frequency by clicking on a link.

☒ **By Day**   [By Week](#)   [By Month](#)   [By Year](#)   [By Trigger](#)

☒ Every  minute(s)  
☐ Every  hour(s)  
☐ Every  day(s)

**Start:**

**End:**  
☒ No end date  
☐ End by:

**Tasks:** Entries: 1 - 1

| Name        |
|-------------|
| MFR_TRIGGER |

**Prompt values**

☐ Override the default values  
 No values saved

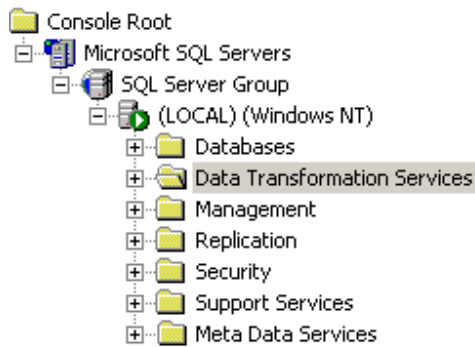
1. Click **OK**
2. The job will now run automatically when data has been published to the Controller Data Mart
- 3.

Congratulations you have now implemented the MFR BluePrint.

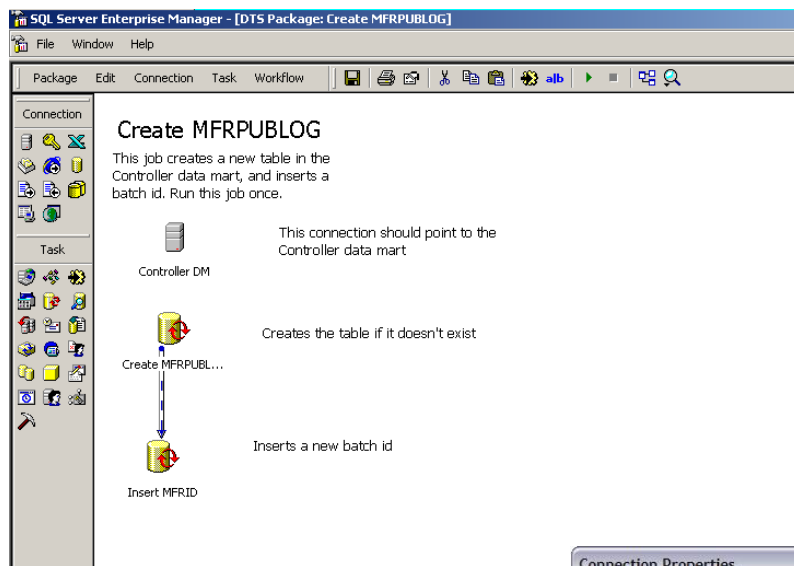
### ***MFR Trigger - MS SQL 2000 DTS:***

You can also use functionality in MS SQL Server DTS (Data Transformation Services) to execute the job.

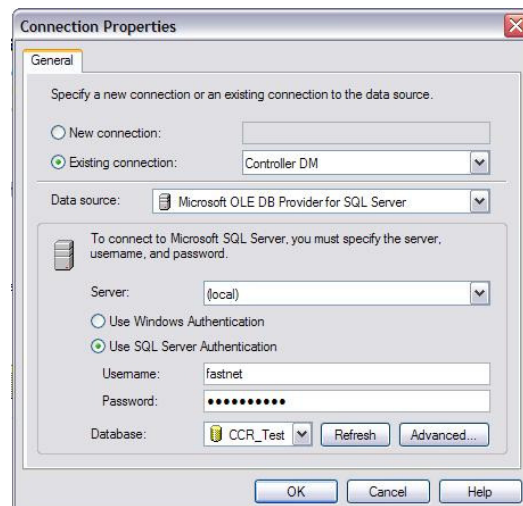
1. Launch **MS SQL Server Enterprise Manager**
2. Expand the server and locate the **Data Transformation Services** folder



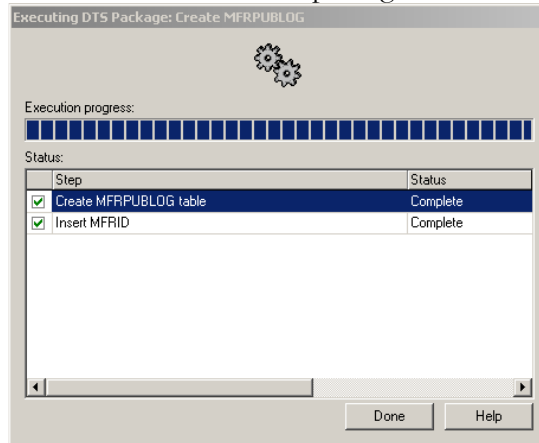
5. Right-click and select **Open Package...**
6. Browse for **Create MFRPUBLOG.dts**
7. Click **Open**



8. Double-click the **Controller DM** connection
9. Specify the correct connection properties, select the Controller data mart as the Database.
10. Click **OK**

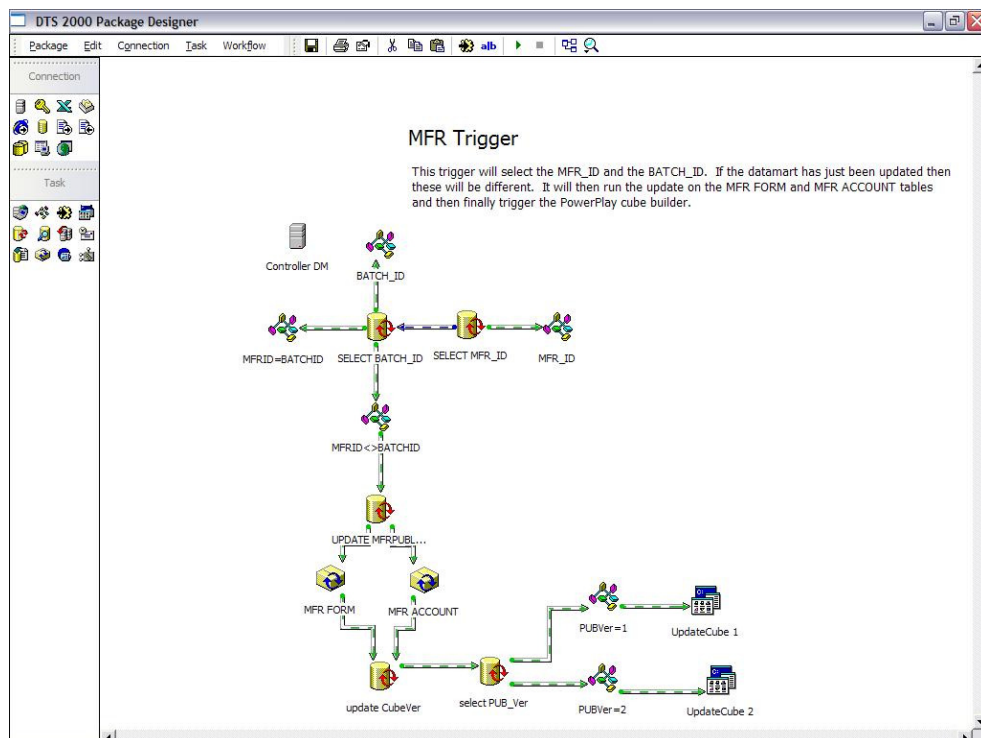


11. Click **Save**
12. Select **Package/Execute** to execute the package



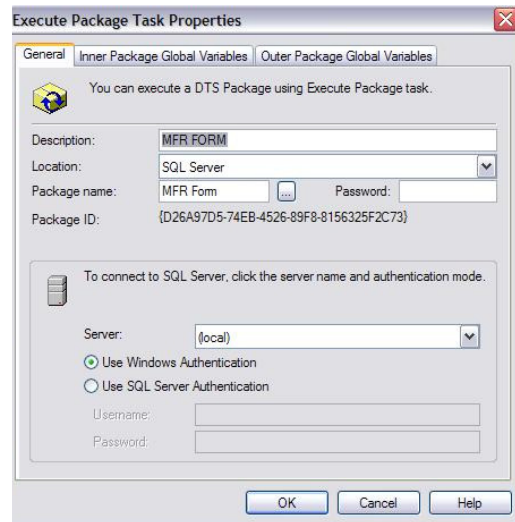
The **MFRPUBLOG** table has now been created and a batch id has been inserted

13. Close the package
14. Back in **MS SQL Enterprise Manager**, right-click **Data Transformation Services**
15. Select **Open Package...**
16. Browse for **MFR Trigger.dts**

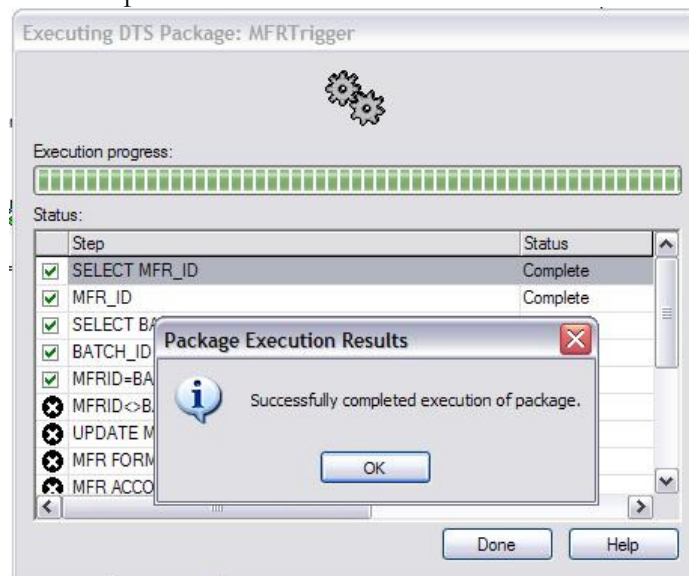


17. Double-click the **Controller DM** connection and verify that the connection properties are correct, the connection should point to the Controller data mart

18. Select the **MFR Form** task  and go to **properties**.
19. Ensure that the Package Name and Location are correct.
20. Select the **MFR Account** task and go to **properties**
21. Ensure that the package Name and Locations are correct.



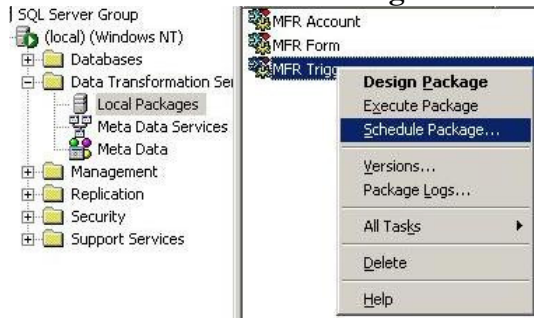
22. Check the **properties** of the UpdateCube1 and UpdateCube2 processes ensuring that the Win Process32 path points to the location of your PowerPlayCubeBuilder directory.
23. Click **Save AS** and save the package in the Location of SQL Server.
24. Execute the package, you should now see that different steps are executed dependent on if data has been published or not.



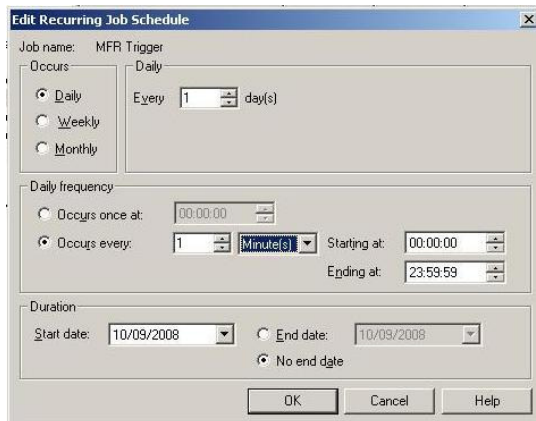
25. Close the package

The next step is to automate the MFR Trigger dts job to check for changes in the database. For this to work you will need to ensure that the db administrator has started SQL server agent. Now we schedule the dts job:

1. Right click of over the **MFR Trigger dts job** .
2. Select **Schedule Package**



3. In the schedule screen select the frequency you would like the database to check for updates. We have selected every minute.



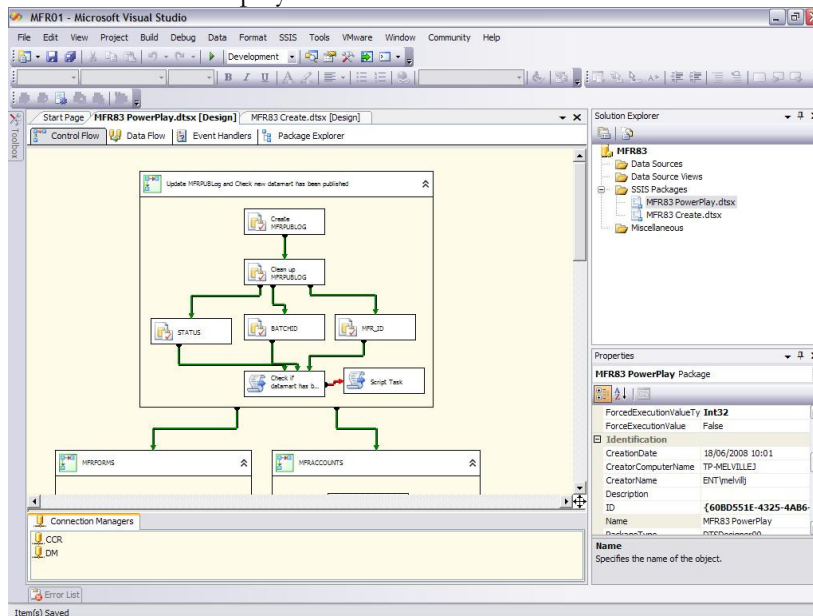
4. Now select **OK**

Congratulations you have now implemented the MFR BluePrint.

### ***MFR Trigger - MS SQL 2005 SSIS:***

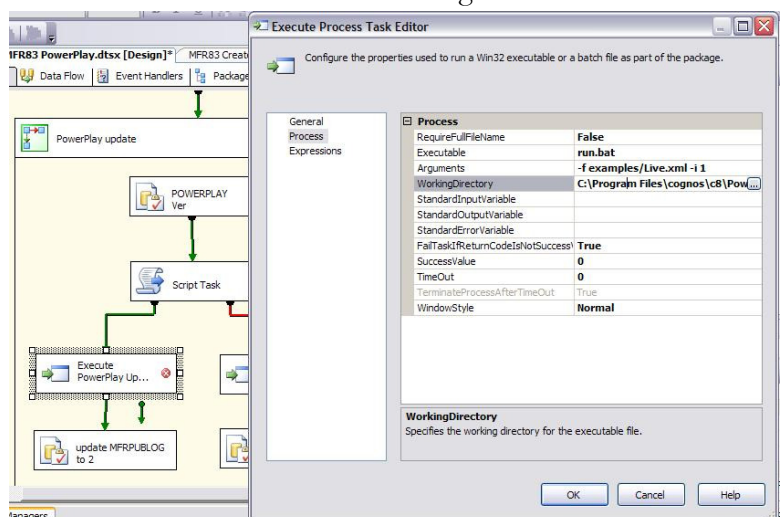
The MFR Trigger can also be achieved using MS SQL 2005 SSIS. To run through the steps below SQL Server Business Intelligence Development Studio needs to be installed.

1. Open up SQL Server Business Intelligence Development Studio.
2. Go to **File > Open** and find the **MFR01.sln** file included in the blueprint.
3. In the solution window (if not present go to **View > Solution Explorer**) double click on the **MFR83 PowerPlay.dtsx**
4. This should display the solution in the main window:



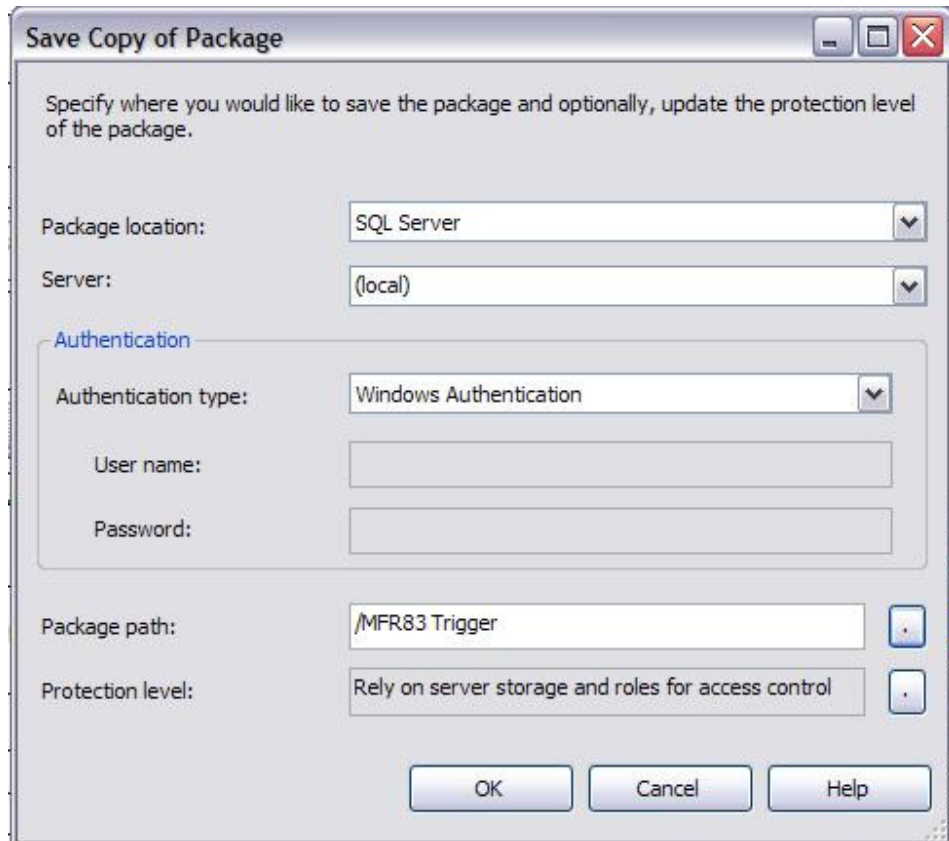
The MFR83 PowerPlay.dtsx checks to see if an update has happened to the data mart. It then updates the form and account tables and finally updates the PowerPlay cube.

5. In the connection manager window. Double click on **CCR** and change the connection to look at the Controller Database
6. In the connection manager window. Double click on **DM** and change the connection to look at the Data Mart Database
7. In the flow chart in the centre screen scroll down to the PowerPlay update section and double click on **Execute PowerPlay Update1**
8. Click on the **Process** section
9. Change the Working Directory value to the location of the PowerPlayCubeBuilder.
10. Follow steps 8-9 for **Execute PowerPlay Update2**.



Finally we need to save the ssis package to the sql server to enable us to automate it.

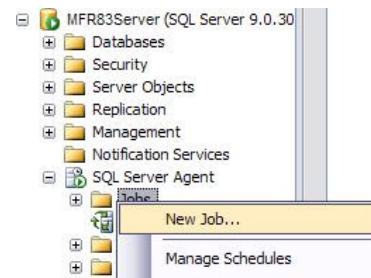
1. Go to **file > Save Copy of MFR83 PowerPlay.dtsx** as
2. **Ensure the Package Location** is set to **SQL Server**
3. Give the Package Path name **MFR83 Trigger**
4. Ensure the Protection Level is: **Rely on server storage and roles for access control.**

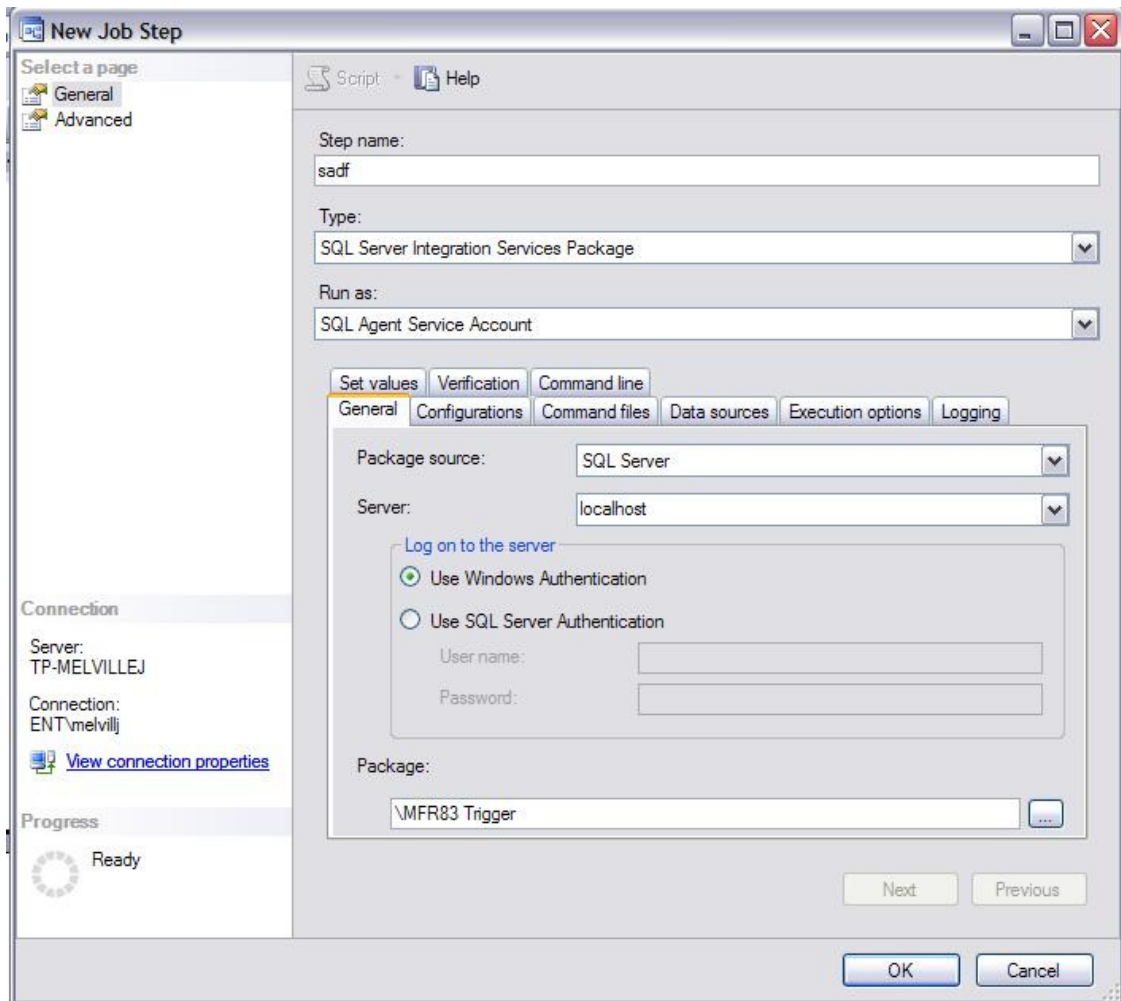


5. Click **OK**

The next step is to schedule the SSIS job. To do this we are going to create a sql agent job.

1. Open **SSQ Server Management studio**
2. Connect to the server.
3. In the Object Explorer move down to **SQL Server Agent > Jobs**
4. Give the job a name in the **General Page**.
5. Move to the **Steps Page**
6. Click **New**
7. Give the step a name
8. Change the Type to **SQL Server Integration Services Package**





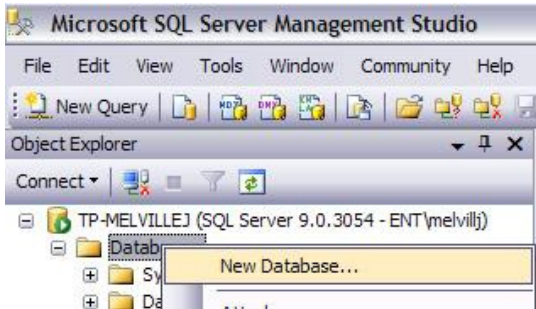
9. Ensure Package Source: is set to **SQL Server**
10. Set the Server to be the correct server.
11. Select the **MFR83 Trigger** package
12. Select OK.
13. go to the **Schedules Page**
14. Click **New**
15. Give the schedule a name and then select the frequency that you require.
16. Click **OK** on the Job Schedule Properties
17. Click **OK** on the Job Step Properties

Congratulations you have now implemented the MFR BluePrint.

## Appendix 1

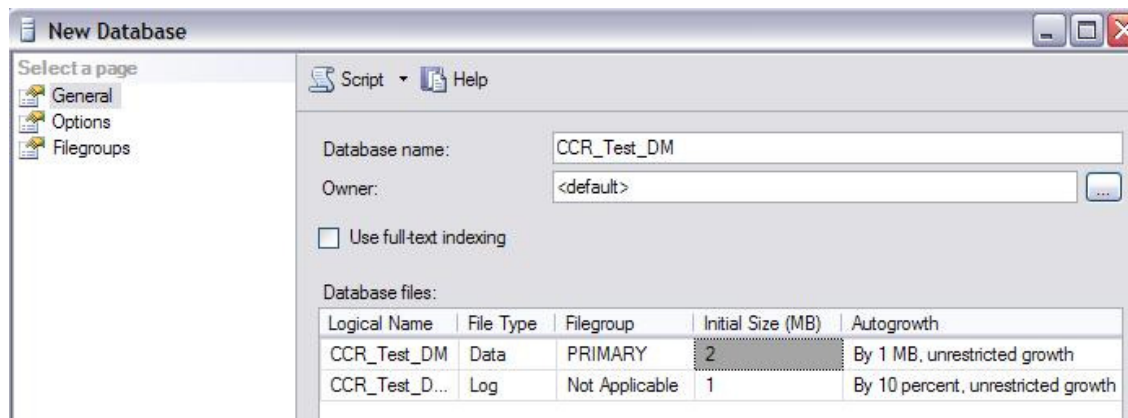
### Creating a SQL Server Database

Open up Microsoft SQL Server management Studio.

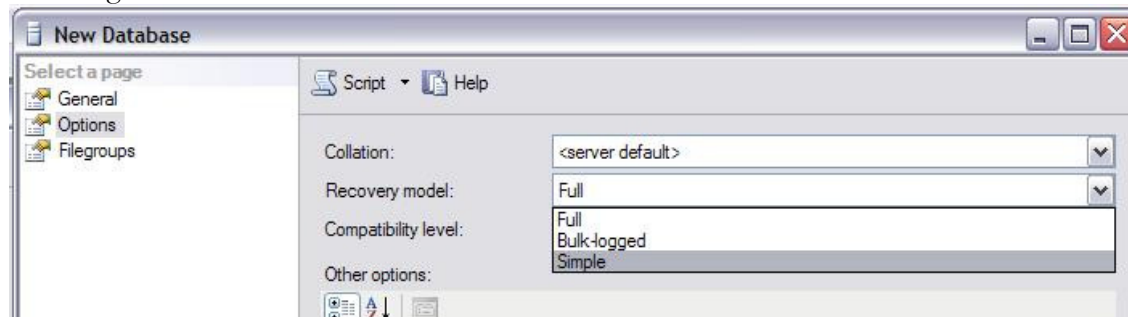


Expand down the tree to find databases and then right click to create a new database

Under general options put in the database name and ensure that it is being created in the correct path by scrolling to the right of the bottom scroll bar.



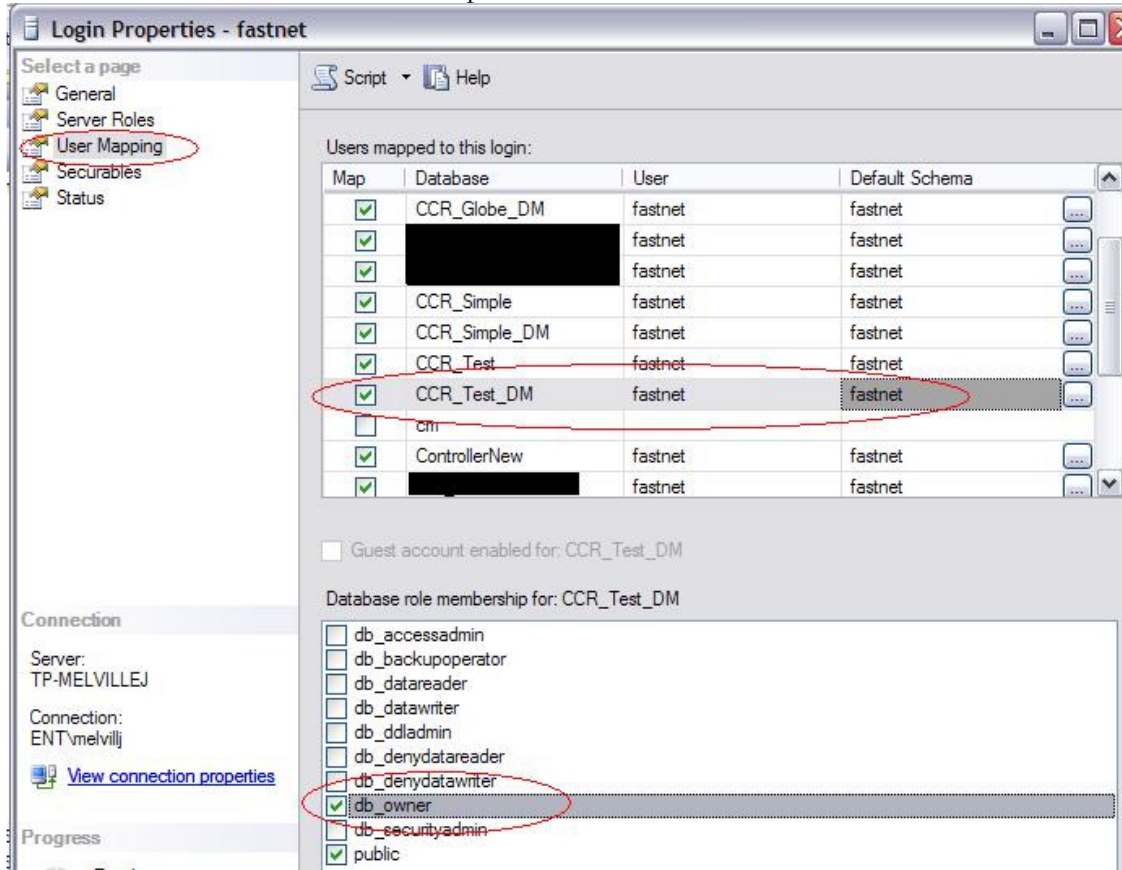
Go to options and change the recovery model to simple. This will reduce the size of the log file being created.



Once complete click on the ok button.

The next step is to set up the security so that fastnet can access the new database and to create the new tables required by controller. To do this, move down the console tree to **Securities > Logins > fastnet**. Right click on fastnet and go to properties and then select **User Mapping**.

Here you will need to tick the 'Map' tick box for your new database and add the default schema to be fastnet. At the bottom part of the screen 'Database role membership' click on **db\_owner**. Then click on OK to complete.



## Appendix 2

### Issues with Publish Data mart - Error

|                    |                                                                                            |
|--------------------|--------------------------------------------------------------------------------------------|
| <b>Title:</b>      | Error 'File name or class name not found during Automation operation' publishing data mart |
| <b>Document#:</b>  | 1037269.1                                                                                  |
| <b>Updated:</b>    | Apr 04, 2008 by RICHARD COLLINS                                                            |
| <b>Applies To:</b> | Controller (Frango) 8.2                                                                    |
| <b>SRs:</b>        | 2                                                                                          |
| <b>Collection:</b> | KB                                                                                         |

#### Description:

User clicks on 'Transfer - Publish Data and Structures - Publish Data Mart'.  
 User selects a template whose 'target' is set to 'Cognos BI'  
 User clicks on blue 'Run' button  
 User receives error message.

#### Error Message:

```
Error
File name or class name not found during Automation operation
OK
```

#### Root Cause:

Each separate Controller application database requires a corresponding 'Data Mart' database defined inside 'Database connections for publish to datamart'.

If this section is not filled in correctly, then the corresponding connection 'UDL' file does not exist (or is invalid) so the Data Mart publish process does not have the connection information to reach the target database.

Also, after creating the connection, the Controller Batch Service must be restarted, so that it recognises the new UDL file.

#### Solution:

Check UDL file exists, and ensure that all details are correct (e.g. correct SQL login username/password).

#### Steps:

1. Logon to the Controller application server as an administrator
2. Launch Cognos Controller Configuration from the Start Menu
3. Open the menu item 'Database connections for publish to data mart'
4. Ensure that this section is filled in for \*all\* relevant databases (chosen under ♦Select database♦)

TIP: Ensure that the ♦Initial catalog♦ is set to your \*separate\* (dedicated) Data Mart database (i.e. not a shared database, and not the same as your Controller application database♦s repository database).

5. Click green ♦tick♦ to check connection, and ensure that receive success message

6. Obtain short period of downtime for system, and then stop & restart the **Controller Batch Process 1** Windows service

TIP: For most systems, only have **Controller Batch Process 1** set to **Running** (leave all the others as **Stopped**).

*TIP: For more detail, including print-screens, see attached document 'KB\_1037269\_.pdf'.*

**Related documents:**

KB 1037026 - Test system restore: Data mart publish fails to File name or class name not found

## Appendix 3

### Issues with Publish Data mart – ‘dmversion’

|                    |                                                                                         |
|--------------------|-----------------------------------------------------------------------------------------|
| <b>Title:</b>      | Error - Invalid object name 'dmversion' - when publishing Data Mart to Cognos BI format |
| <b>Document#:</b>  | 1035260.3                                                                               |
| <b>Updated:</b>    | Apr 15, 2008 by CHRISTINE SIMPSON                                                       |
| <b>Applies To:</b> | Cognos 8 BI Framework Manager 8.2   Controller (Frango) 8.2                             |
| <b>SRs:</b>        | 5                                                                                       |
| <b>Collection:</b> | KB                                                                                      |

---

#### Description:

User clicks on 'Transfer - Publish Data and Structures - Publish Data Mart'. User selects a template that has a target of 'Cognos BI'. User ticks 'Generate' and selects 'Period' and 'Structures' and then clicks 'Run', but receives an error message.

#### Error Message:

```
Error
Invalid object name 'dmversion'.
OK
```

There is also a record in the Event Log:

```
Event Type: Warning
Event Source: Cognos Controller
Event Category: None
Event ID: 0
Date: 31/08/2007
Time: 14:15:56
User: N/A
Computer: SERVERNAME
Description:
Error occurred at 31/08/2007 14:15:56 in Cognos Controller, Error No=-2147217865,
Source=DMPublishD.GetPublishDBVersion, Description=Invalid object name 'dmversion'.,
HelpFile= HelpContext=0
```

#### Root Cause:

##### Scenario #1

The administrator has not run the DmConv utility to populate the corresponding 'Data Mart database' with the correct tables, for this particular database connection.

##### Scenario #2

The user has not recently run a Database Optimise, which creates relevant tables required for the data mart publish.

#### Solution:

##### Scenario #1

Check that there is a valid corresponding database connection inside 'Database Conversion Utility'. Then click 'run' on the database connection, and choose the Data Mart database. Click on "run steps".

### Scenario #2

Choose 'single user' mode in Controller, and run a database optimise.

#### **Steps:**

### Scenario #1

1. Logon to the application server
2. Launch Controller Configuration from the Start Menu
3. Open 'Database Connections for publish to data mart'
4. Ensure that this section is correctly filled in, and (**important**) click 'save' to register this in the system
5. Open 'Database Connections' and highlight the relevant connection name
6. Click on 'run' button to launch 'Database Conversion Utility'
7. Click on 'DmConv' tab
8. Click on the 'browse' button and locate the relevant UDL file (e.g. C:\Program Files\cognos\c8\DMData\Controller\_Live.udl) and **note** that there is more than one UDL file, this one is NOT the same as the one for the main database in \data\ folder.
9. Look at 'DM Version'. Notice that 'actual' is zero, and DmConv is higher (e.g. 26 for Controller 8.2 RTM)
10. Click 'Run Steps' and then click 'Yes' to confirm
11. After seeing the message ALL PROCESSES COMPLETE, click close.

#### **Related Documents:**

KB 1039214 - How to use the 'DmConv' tab inside Controller Configuration to publish CognosBI Data Mart  
KB 1037539 - Error 'No values found for Template' when trying to export data from Controller into a Data Mart  
Attached guide

## Appendix 4

### Issues with Publish Data mart – ‘No value found for Template’

|                    |                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------|
| <b>Title:</b>      | Error 'No values found for Template' when trying to export data from Controller into a Data Mart |
| <b>Document#:</b>  | 1037539.2                                                                                        |
| <b>Updated:</b>    | Jul 18, 2008 by RICHARD COLLINS                                                                  |
| <b>Applies To:</b> | Controller (Frango) 8.2                                                                          |
| <b>SRs:</b>        | 2                                                                                                |
| <b>Collection:</b> | KB                                                                                               |

---

#### **Description:**

User publishes data mart for period where there is data, but receives error.

#### **Error Message:**

##### English:

Details Section xxxx  
No values found for Template

##### French:

Section de details  
Libelle  
Aucune valeur trouvee pour le modele

#### **Root Cause:**

The Controller system has been fundamentally changed (e.g. it is in development phase, and account structure has recently changed) but the database has not been 'optimised' since. This means that some tables (which the Data Mart publish requires) have not been updated.

#### **Solution:**

Perform a database optimisation, then re-try the Data Mart publish.

#### **Steps:**

1. Maintain - User - Single User Mode
2. Maintain - Database - Optimise
3. Select all options, and click 'run'
4. Re-publish.

#### **Related documents:**

KB 1027824 - How to handle periods with no values in Cube/OLAP Tables.