

Guideline

Transfer Data from TM1 to IBM Cognos Controller with a TI Process

Product(s): IBM Cognos Controller

Area of Interest: Financial Management

Copyright and Trademarks

Licensed Materials - Property of IBM.

© Copyright IBM Corp. 2012

IBM, the IBM logo, and Cognos are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

While every attempt has been made to ensure that the information in this document is accurate and complete, some typographical errors or technical inaccuracies may exist. IBM does not accept responsibility for any kind of loss resulting from the use of information contained in this document. The information contained in this document is subject to change without notice.

This document is maintained by the **IBM Business Analytics Proven Practices** team. You can send comments, suggestions, and additions to cscogpp@ca.ibm.com.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Contents

1 INTRODUCTION 5

1.1 PURPOSE 5

1.2 APPLICABILITY 5

1.3 EXCLUSIONS AND EXCEPTIONS 5

2 TRANSFERRING DATA BETWEEN TM1 AND IBM COGNOS CONTROLLER WITH A TI PROCESS..... 5

2.1 OVERVIEW 5

2.2 STEPS TO BE COMPLETED AS PART OF DATA IMPORT 6

2.3 RESULTS IN CONTROLLER 13

1 Introduction

1.1 Purpose

This document describes how to transfer data from a IBM Cognos TM1 cube into IBM Cognos Controller using staging tables and a stored procedure to carry out the import automatically.

1.2 Applicability

IBM Cognos Controller

1.3 Exclusions and Exceptions

There are no known exclusions and exceptions at the time this document was created.

2 Transferring Data between TM1 and IBM Cognos Controller with a TI Process

2.1 Overview

This document describes how to transfer data from a TM1 cube into IBM Cognos Controller using staging tables and a stored procedure to carry out the import automatically.

If you need to transfer data from more than one cube, then additional TI processes can be created and then they can all be included in a single chore to fire them all off in one step.

Essentially it is possible to transfer data from TM1 to Controller with a single click or automatically on a scheduled basis.

The examples described here have been created in a specific copy of the UKFPM VMware Image.

The examples and instructions have been created so that it should be easy to replicate the steps in any install with TM1 and Controller.

An Import Specification must be created in Controller to import the data from the staging tables to the live tables, if there are no special steps to be carried out in Controller this can simply be a copy of the default specification provided with any Controller install, #ST_DATA.

Additional steps can be added to the Import Specification in Controller using the functionality available in designing Import Specifications.

2.2 Steps to be completed as part of data import

The following steps are required as part of the data import using Turbo Integrator:

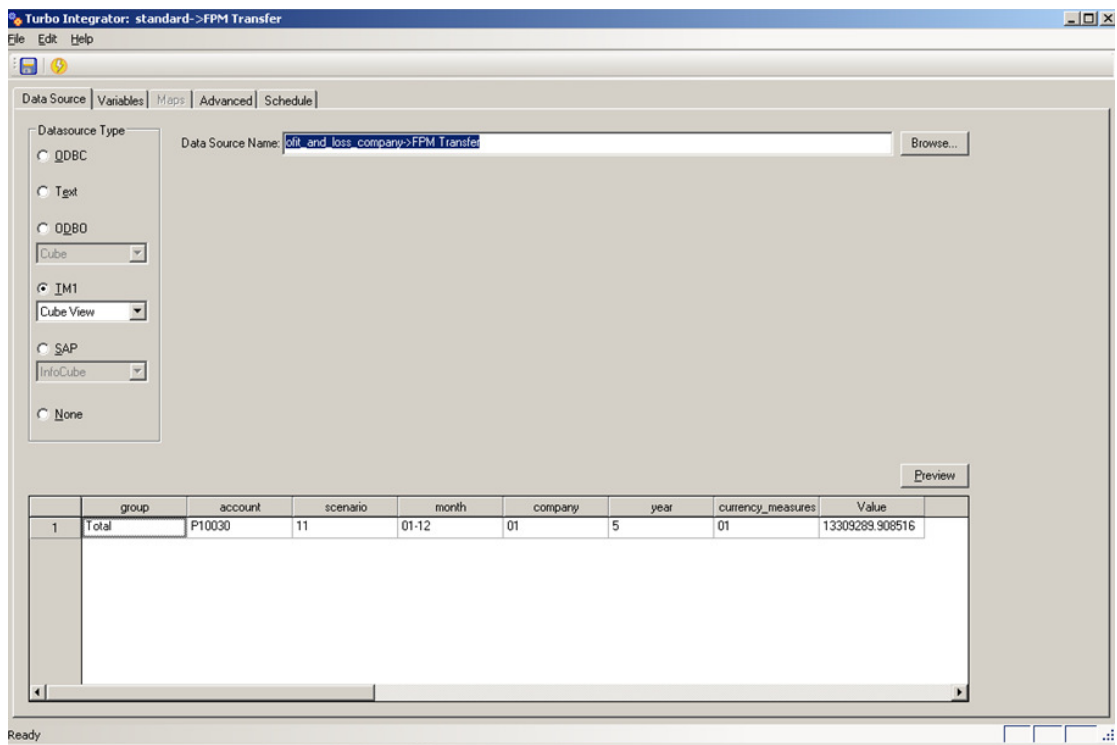
1. Create an ODBC connection to the Controller DB, under **Admin Tools/Data Sources/System DSN**.
2. Create a view in TM1 that contains the data you wish to transfer, selecting the right alias for the dimensions can make the view cleaner. The example view is called *FPM Transfer* in the UKFPM example.

The preview of the view in TI displays the member IDs, but the transfer of data uses the Aliases used in the View, this can be difficult to work with and care must be taken. See *Best Practice Notes* in Step 4 to ensure the transfer is robust.

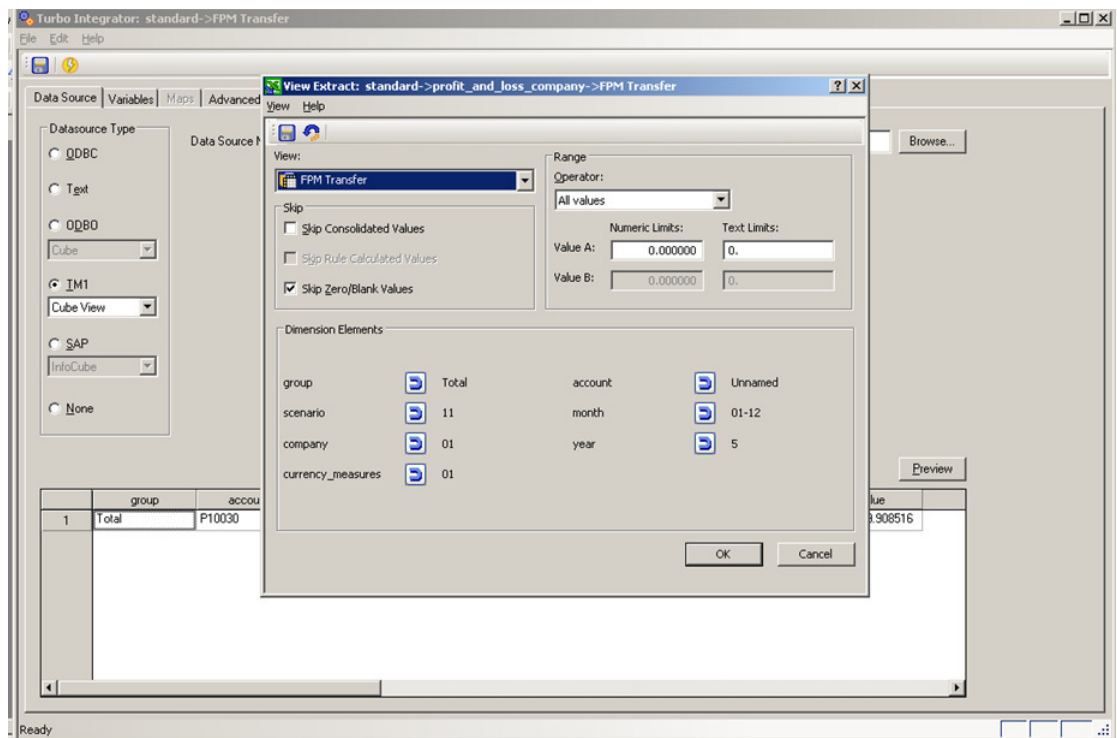
The screenshot shows a window titled 'Cube Viewer: standard->profit_and_loss_company->FPM Transfer'. The window contains a data table with the following columns: group, account, scenario, month, company, year, currency_measures, and a final column for values. The data is organized into a hierarchy, with a '+ Total' group at the bottom. The 'currency_measures' column is highlighted in yellow.

group	account	scenario	month	company	year	currency_measures	
	+ P10030	Actual					13,309,289.91
	+ P10040	Actual					6,028,086.77
	P10050	Actual					0.00
	+ P10060	Actual					0.00
	P10070	Actual					0.00
	P11010	Actual					0.00
	P11020	Actual					0.00
	P11030	Actual					0.00
	P11040	Actual					0.00
+ Total	+ P20010	Actual	+ Dec YTD	Bobs Bike Store	2008	Local Currency	1,323,715.83
	+ P20020	Actual					2,478,768.80
	+ P20030	Actual					(4,565.88)
	+ P20040	Actual					16,299.42
	+ P21010	Actual					33,487.35
	P21020	Actual					0.00
	P21030	Actual					0.00
	P21040	Actual					0.00
	+ P22010	Actual					19,438.83
	P23010	Actual					0.00

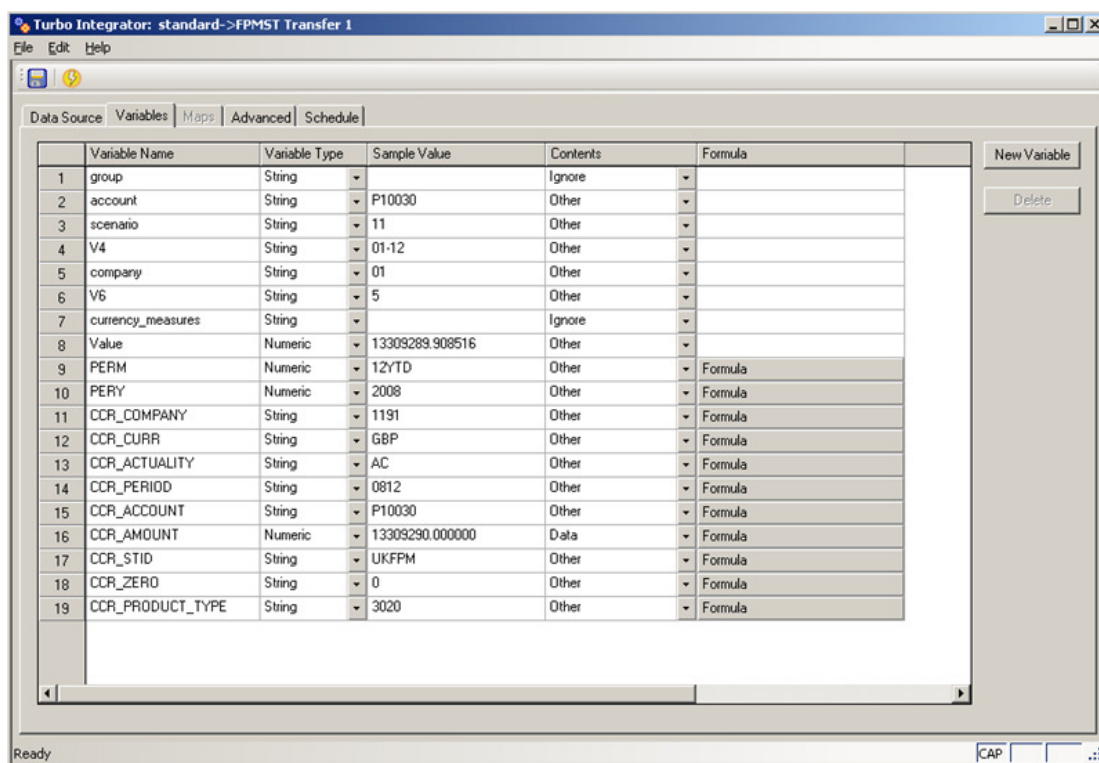
3. Create a TI process. The data source should be a TM1 Cube View, you can browse to the correct cube to view. It will then display the first line in the **Preview** window named *FPM Transfer* in the UKFPM example.



When you are browsing for the View, there is an option to **Create View**. You can check the settings on this to ensure the view is correctly configured. For instance, if you are looking at aggregated values you clear the **Skip Consolidated Values** check box.



- On the **Variables** tab, select **Other** in the **Contents** column for all items that you want to use. One item should be selected as **Data**.



You can calculate additional variables to supplement the information from the View or manipulate information from the View.

Best Practice Notes:

If the member IDs in TM1 are different to Controller codes, then you can use an Alias to hold the Controller type code on each member. In the example an Alias called *CCR* has been created for several dimensions.

The view can be set to display the most appropriate ID or Alias for users and then within the variables it is possible to ensure that the correct Alias is sent to Controller by using additional variables and TM1 formulae.

For instance: in the Scenario dimension, the following Aliases are used

Member ID = "11",
 Member_Caption = "Actual",
 English = "Actual",
 Deutsch = "Ist"
 CCR = "AC"

Using the TM1 formula to calculate a new variable:
 CCR_ACTUALITY=ATTRS('SCENARIO',SCENARIO,'CCR');
 you can ensure that it is always the CCR Alias that is delivered to
 Controller.

Another example is calculating the correct form for the Period code, in
 this example TM1 has separate dimensions for Month and Year, so the
 correct aliases are collected for both Month and Year and then
 combined creating a new variable using the following formulae

```
PERM=ATTRS('MONTH',V4,'CCR');
```

```
PERY=ATTRS('YEAR',V6,'MEMBER_CAPTION');
```

```
CCR_PERIOD=INSRT(SUBST(PERY,3,2),SUBST(PERM,1,2),1);
```

For help on these functions look in TM1 Help under *Rules Functions*.

5. In the **Advanced** tab / **Data** you need to add some SQL. It will transfer the data directly into the staging table xstagefact. An *st_id* of *UKFPM* is used at this step, it will be updated later. The example SQL is divided into four sections, as TI can only handle 255 characters in one string. It has been divided as follows:

SQL1 – contains the “insert” columns for all the core Controller dimensions

SQL2 – contains the “insert” columns for some additional dimensions

SQL3 – contains the “value” items for all the columns listed in SQL1

SQL4 – contains the “value” items for all the columns listed in SQL2

You can add or remove items in the SQL if required, but both the insert and value lines have to be changed.

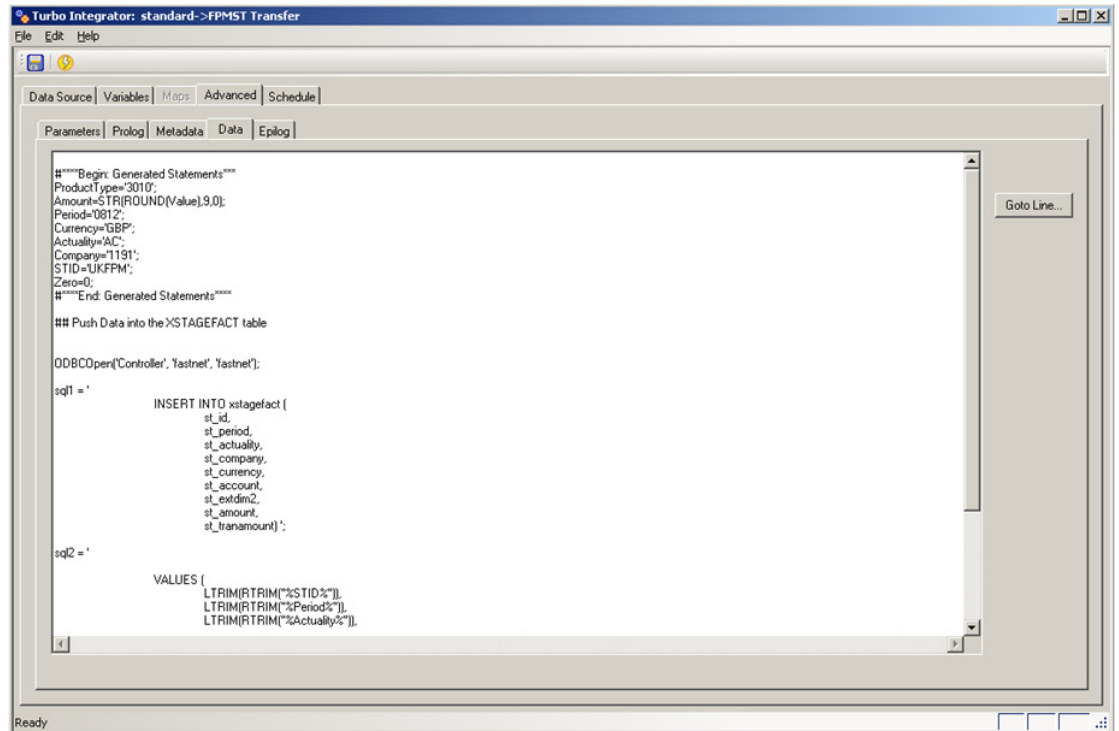
In the **Values** section the names need to be changed to match the Variable names you have defined, for instance in this case the variable for **Period** is *CCR_PERIOD*.

```
VALUES (
                                LTRIM(RTRIM('%CCR_STID%')),
    LTRIM(RTRIM('%CCR_PERIOD%')),
    LTRIM(RTRIM('%CCR_ACTUALITY%')),
    LTRIM(RTRIM('%CCR_COMPANY%')),
    LTRIM(RTRIM('%CCR_CURR%')),
    LTRIM(RTRIM('%CCR_ACCOUNT%')),
    %CCR_AMOUNT%,
```

In the line

```
ODBCOutPut( 'Controller', sql1, sql2, EXPAND(sql3), EXPAND(sql4));
```

the EXPAND function turns the %CCR_PERIOD% etc. into the correct values at runtime.



<<< SQL Example Code <<<

```
## Push Data into the XSTAGEFACT table

ODBCOpen('Controller', 'fastnet', 'fastnet');

sql1 = ' INSERT INTO xstagefact (
                                st_id,
                                st_period,
                                st_actuality,
                                st_company,
                                st_currency,
                                st_account,
                                st_amount, '

sql2 = '                                st_extdim1,
                                st_extdim2,
                                st_extdim3,
                                st_extdim4,
```

```

        st_c_company,
        st_trancurr,
        st_tranamount,
        st_freel)';

sql3 = '
        VALUES (
                LTRIM(RTRIM('%CCR_STID%')),
        LTRIM(RTRIM('%CCR_PERIOD%')),
        LTRIM(RTRIM('%CCR_ACTUALITY%')),
        LTRIM(RTRIM('%CCR_COMPANY%')),
        LTRIM(RTRIM('%CCR_CURR%')),
        LTRIM(RTRIM('%CCR_ACCOUNT%')),
        %CCR_AMOUNT%,';

sql4 = '
        NULL,
        LTRIM(RTRIM('%CCR_PRODUCT_TYPE%')),
        NULL,
        NULL,
        NULL,
        NULL,
        %CCR_ZERO%,
        NULL)';

ODBCOutPut( 'Controller', sql1, sql2, EXPAND(sql3), EXPAND(sql4));

ODBCClose('Controller');

>>

```

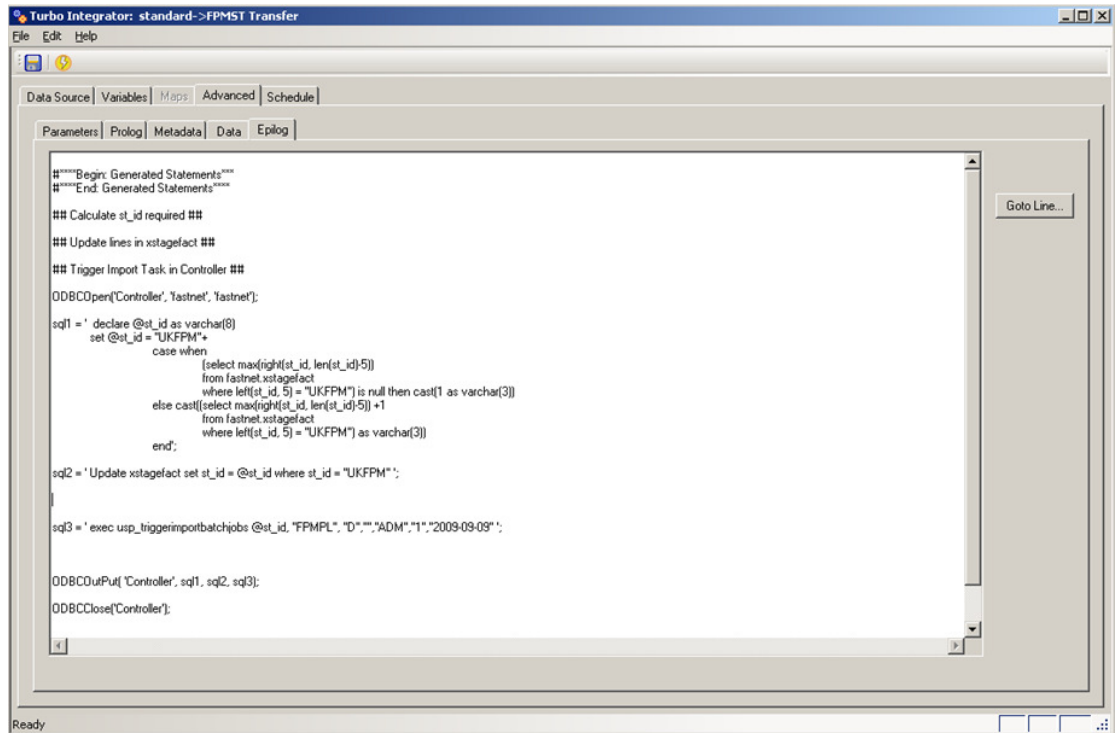
- In the **Advanced / Epilog** tab add the SQL to create the correct st_id. Update the st_id for the lines just inserted into xstagefact and send the trigger for the stored procedure in Controller.
- The first section of SQL works out and create a *Job Name* to be inserted into the column st_id. This looks at the table to find if there are any Job Names starting with UKFPM. If there are, it will add 1 to the highest number of that job to create a new Job Name.
- The second section updates all items that have a Job Name of *UKFPM* with the unique Job Name created above.
- The third section sends the trigger for the stored procedure.

- Within Controller an Import Specification FPMPL has been created to load the data from the Staging Tables to the live tables. This can include operations and lookups where required.

The trigger is:

```
exec usp_triggerimportbatchjobs @st_id, @spec,
'D', '', 'ADM', '1', '2009-09-09'
```

The "1" tells Controller to import immediately.



<<< SQL Example Code <<<

```

## SQL1 = Calculate st_id required ##

## SQL2 = Update lines in xstagefact ##

## SQL3 = Trigger Import Task in Controller ##

## NB Update the name of the Import Specification in the section sql3
##

ODBCOpen('Controller', 'fastnet', 'fastnet');

sql1 = ' declare @st_id as varchar(8)
        set @st_id = 'UKFPM'+
    
```

```
case when
  (select max(right(st_id, len(st_id)-5))
   from fastnet.xstagefact
  where left(st_id, 5) = 'UKFPM') is null then cast(1 as
                                     varchar(3))
  else cast((select max(right(st_id, len(st_id)-5)) +1
   from fastnet.xstagefact where left(st_id, 5) = 'UKFPM') as
  varchar(3))
end';

sql2 = ' Update xstagefact set st_id = @st_id where st_id = 'UKFPM'
';

sql3 = ' declare @spec as varchar(12)
        set @spec = 'FPMPL'

exec usp_triggerimportbatchjobs @st_id, @spec,
'D','','ADM','1','2009-09-09' ';

ODBCOutPut( 'Controller', sql1, sql2, sql3);

ODBCClose('Controller');

>>>
```

2.3 Results in Controller

Under **Batch Queue / Manage** you will be able to see the Job Name after a successful execution. It will have been allocated the next available batch ID number.

An *Import External Data Log Report* is also created. This log report can records the number of lines read and any rejections.

If the specification has been activated for the Tracking Report it will also be there. The example in MWM has been activated.

Data will have been loaded for the appropriate Company and the Status will have been updated.