



Installation Guide

QAD Business Intelligence

78-0956-3.7
QAD Business Intelligence 3.7
September 2012

This document contains proprietary information that is protected by copyright and other intellectual property laws. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

QAD and MFG/PRO are registered trademarks of QAD Inc. The QAD logo is a trademark of QAD Inc.

Designations used by other companies to distinguish their products are often claimed as trademarks. In this document, the product names appear in initial capital or all capital letters. Contact the appropriate companies for more information regarding trademarks and registration.

Copyright ©2012 by QAD Inc.

BIPortal_IG_v307.pdf/r9m/r9m

QAD Inc.

100 Innovation Place
Santa Barbara, California 93108
Phone (805) 566-6000
<http://www.qad.com>

Contents

Change Summary	vii
Chapter 1 Installing QAD BI Analytical Modules	1
Pre-Install Information	2
Data Warehouse	2
Installation of QAD BI Analytical Modules	12
Requirements	13
Installation Steps	13
Install BI Analytical Modules	14
TM Module Setup	23
Add New Source to Existing Installation	24
Adding New Module to Existing Installation	24
Chapter 2 Installing QAD BI Portal Web Application	27
Requirements	28
Configuration Parameters	28
Installation Notes	28
Installation Steps	28
Install BI Portal Web Application	29
Note on Password Encryption in the BI Portal	34
Customizing the BI Portal Logo	34
Getting Started in the BI Portal Web Application	34
Install Standard Dashboards and Reports	34
Installing QAD BI Mobile App on the iPad	36
Chapter 3 QAD BI Portal CFO Dashboard	37
QAD Standard CFO Dashboard	38
Configuring the CFO Dashboard	38
Metric Example	39
Reports Tab	39
ReportLines Tab	39
ReportLineRanges Tab	40
ReportLineCalculations Tab	40
Providing the Workbook Data to the Data Warehouse	40

Chapter 4	Reference	43
Jobs		44
Reference Tables		47
Common Module		47
Financials Module		60
Order Management Module		69
Operations Module		73
Enterprise Asset Management Module		82
Transportation Management Module		85
Chapter 5	QAD BI Upgrade Instructions	95
Upgrade Information		96
Upgrade to QAD BI 3.7 from QAD BI 3.6.x		96
Preparation		96
Upgrade Steps for QAD BI Data Warehouse Designer		96
BI Portal Upgrade Procedure		97
Data Warehouse Tables Changed		101
Upgrade to QAD BI 3.6 from QAD BI 3.5		105
BI Metadata Upgrade Procedure		105
BI Portal Upgrade Procedure		109
Data Warehouse Tables Changed		110
Upgrade to QAD BI 3.5 from QAD BI 3.4.1		113
Appendix A	Sizing Guidelines	121
Introduction		122
Important Restrictions and Caveats on BI Sizing		122
Business Intelligence Deployment Options and Terminology		123
BI Terminology		123
BI Deployment Options		124
Sizing Terminology Used Throughout this Sizing Guide		125
Sizing Methodology for QAD BI		125
CPU Sizing		126
CPU Sizing for DWD/Scheduler Component		126
CPU Sizing for BI Portal Webapp (Server Side)		126
CPU Sizing for Concurrent User Workload (Interactive Portal Users)		126
RAM Sizing		127
RAM Sizing for DWD/Scheduler Component		127
RAM Sizing for BI Portal Webapp (Server-Side)		127
RAM Sizing for Concurrent Users (Interactive Portal Users)		127
Disk and IO Subsystem		128
Note on Importance of Appropriate Disk and IO Subsystem Sizing		128
IO Subsystem Sizing		128
Key IO Subsystem Sizing Notes		128

Notes on Database Disk Sizing	129
Network Considerations	130
Network Recommendations	130
Example Sizing	131
Reference: Estimation of Initial Database Size	132
Dependence of Initial Database Size on BI Modules	132
Estimation of Initial BI Data Warehouse Size	132
Allowance for Future Database Growth	134
Reference: Network Sizing Details	135
Terminology	135
Network Sizing Considerations - Database to BI Database Server ...	135
Network Sizing Considerations - Remote Client to BI Portal	136
WAN Considerations	137
Appendix B Customization Guidelines	139
Overview	140
Customization Challenges	140
QAD BI 3.x Customization Methodologies	143
Custom Path Methodology (Preferred)	143
Integrated Methodology (Last Resort)	145
Customization Needs for Additional Columns	145
Use Track Back Diagram to Check Tables	146
Double Checking the Data Path with a Column Track Back Report ..	148
Notes on Tables	150
Identifying Every Appearance of a Column in Existing Code	152
Naming Conventions for Custom Changes to Packaged QAD BI Modules ...	153
Custom Tables	153
Custom Columns Joined to Standard Module	153
Custom Columns Not Joined to the Baseline Module Tables	153
Guidelines for Creating Custom Path Custom Tables	153
Load Tables	154
Perm Tables and Pre-perm Stage Tables	154
All Other Stage, Extract	154
Adding Required xx_ Prefixed Columns	154
Adding Additional Columns to a Load Table	155
Get Column from Data Source	155
Naming/Column Placement/Data type	155
Transform	156
Create (Recreate) the Table	156
Adding Additional Custom Columns to Pre-perm Stage Tables	157
Connect	157
Transformation/Business Name	158
Create(Recreate) the Staging Table	158

Adding Additional Custom Columns to perm/dim/fact Tables	159
Data Source	159
Placement	159
Column Table Source and Transformation	159
Modify Table	160
Rebuild the Procedure	161
Adding Custom Path stage/extract Tables After the perm Tables	161
Drag and Drop New Table	161
Identify and Set the Columns	161
Identify the Code to be Used in the New Procedure	162
Change the Source Tables	163
Create (Recreate) the table	164
Build the procedure for the custom table	164
Load the table	165
Confirm Results	165
Adding Additional Custom Loading and Staging Tables to Integrate in Additional Columns	166
Additional Customization Topics	166
Appendix C Historically Reloading Modules	169
Historic Reload Instructions	170
Checklist	170
Historic Reload from Perm Tables	170

Change Summary

The following table summarizes significant differences between this document and the previous version.

Date/Version	Description	Reference
September 2012 QAD BI 3.7	Added new subsections to Chapter 1	--
	Added new modules to Estimation of Initial Data Warehouse Size tables	--
	Added Appendix C: Historically Reloading Modules	169
March 2012/QAD BI 3.6	Provided instructions on upgrading from previous version	--
	Added sizing guide	121
	Added customization guide	139
September 2011 QAD BI 3.5	Rebranded for QAD BI 3.5	--

Installing QAD BI Analytical Modules

This chapter describes the initial setup of the Data Warehouse and the installation and configuration steps for QAD Business Intelligence analytical modules.

It discusses the following topics:

Data Warehouse 2

Describes the initial setup of the Data Warehouse.

Installation of QAD BI Analytical Modules 12

Covers the installation and configuration steps for QAD Business Intelligence Analytical Modules.

Pre-Install Information

To save time during the installation, ask the customer in advance for information that you will need during the install. See Chapter 4 for information; use Table 4.8 on page 88 to get system information and Table 4.9 on page 89 to get Parameter information from the customer.

There are two installation packages for Progress Open Edge for Windows: one is for 64-bit, one for 32-bit.

Currently, you must install the 32-bit version; **do not** install the 64-bit version. Even if you have a 64-bit server, you must install the 32-bit package to get the 32-bit ODBC drivers. QAD's understanding of the Progress Open Edge knowledge base is that you cannot install both on the same server. For instance, do not install the 64-bit package on your 64-bit server and then install the 32-bit ODBC drivers. Install the 32-bit Open Edge package only.

Important The Progress Source database or databases must be running Progress version 10.1B03 or later.

Note In the event that the installation requires that the DWD be on a different server than the database server, contact QAD customer support to obtain documentation.

Data Warehouse

A Data Warehouse must be initialized properly before the installation of any QAD BI Analytical Modules.

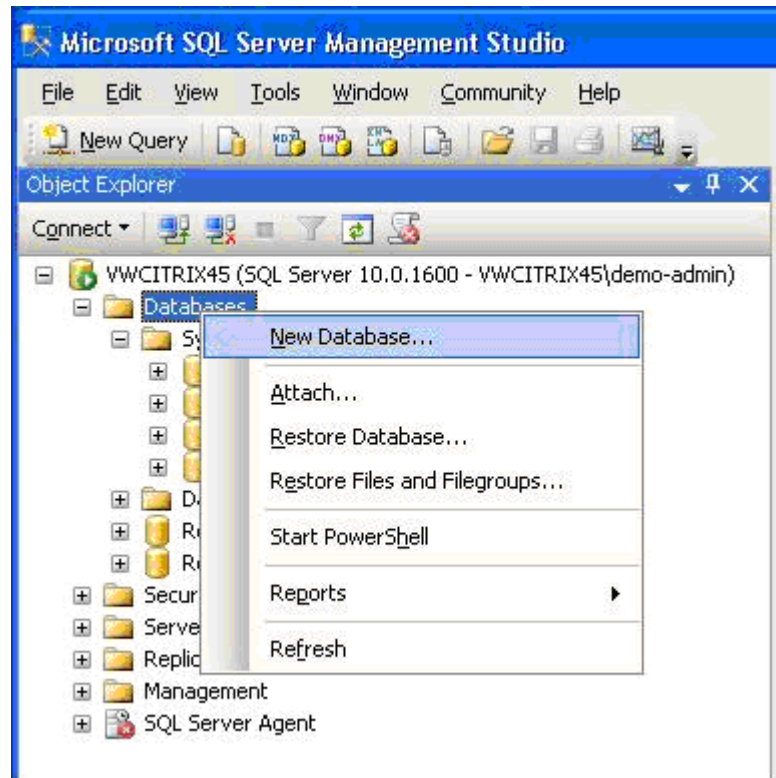
Follow the steps below to install and configure a new QAD BI Data Warehouse instance.

1 Create a database in SQL Server.

- a** From the Microsoft SQL Server Management Studio, right-click on Database, Create new database.

Fig. 1.1

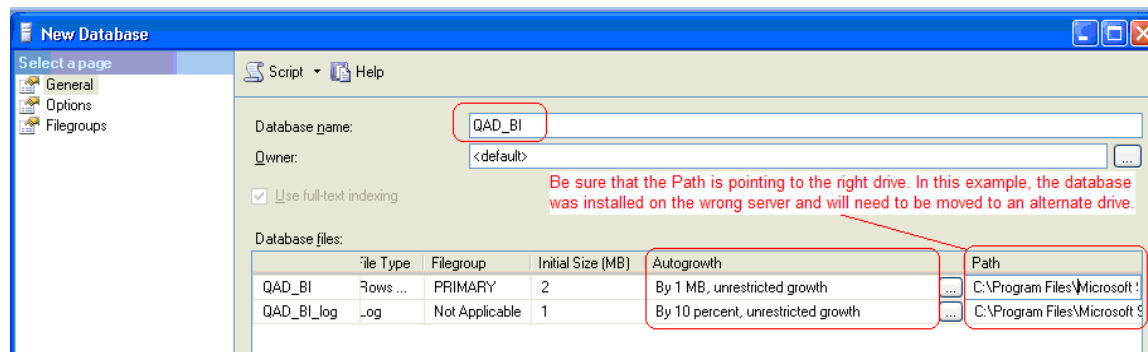
Create New Database 1



- b** Enter the new Database name. Make sure that the Path for the database is pointing to the correct server. Make a note for step c.

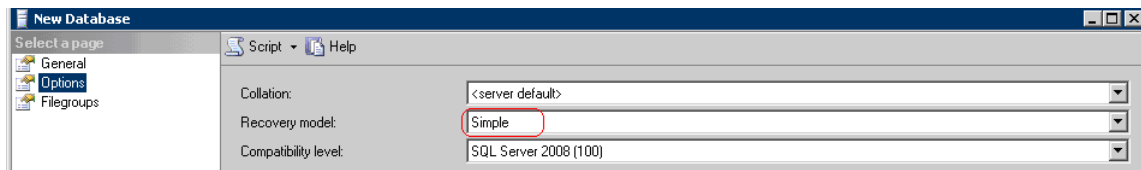
Fig. 1.2

Create New Database 2



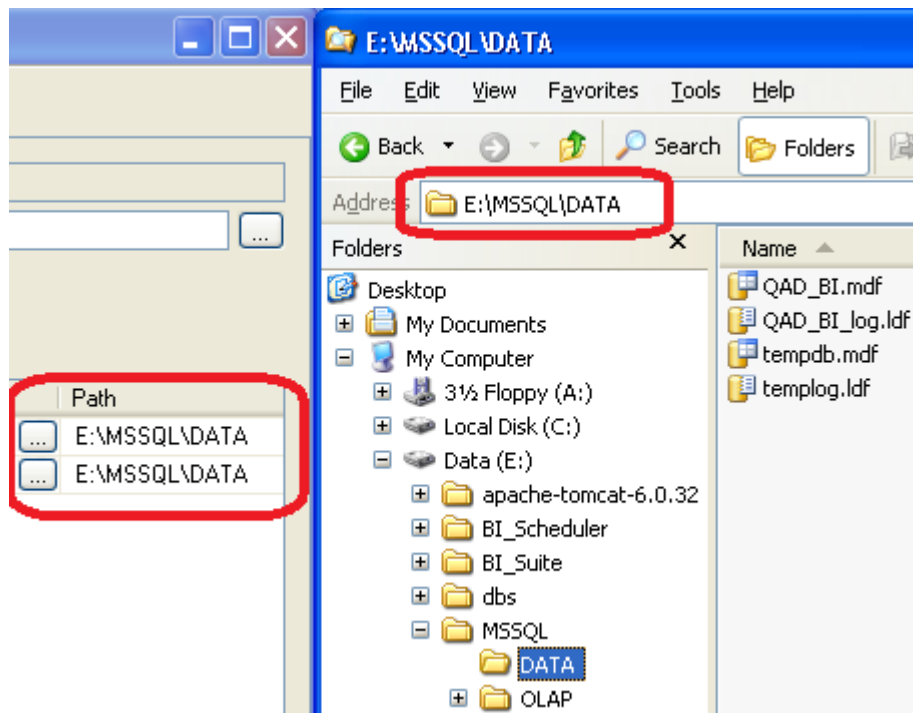
Important It is critical that you choose the Simple recovery mode from the Option tab.

Fig. 1.3
Create New Database3



- c If the Path from step b is correct and it is pointing to the correct server, skip to step 2. If the Path is incorrect, create a new folder on the correct server. Once created, put that directory in for the Path for the database.

Fig. 1.4
Create New Database 4



- d Following up on Step c, if the Path was wrong for the database, the temp database must be checked to confirm that it is being accessed on the right server. Launch a SQL Server database query against your new database and run the following query:

```
USE tempdb
go
exec sp_helpfile
go
```

If the results point to the correct directory, go to step 2. Otherwise, set the tempdb folder using step e.

- e To set the tempdb to the correct directory, reset to appropriate directory (E drive in example below; set as appropriate):

```
USE master
GO
ALTER DATABASE TempDB MODIFY FILE
(NAME = tempdev, FILENAME = 'e:\mssql\data\tempdb.mdf')
GO
```

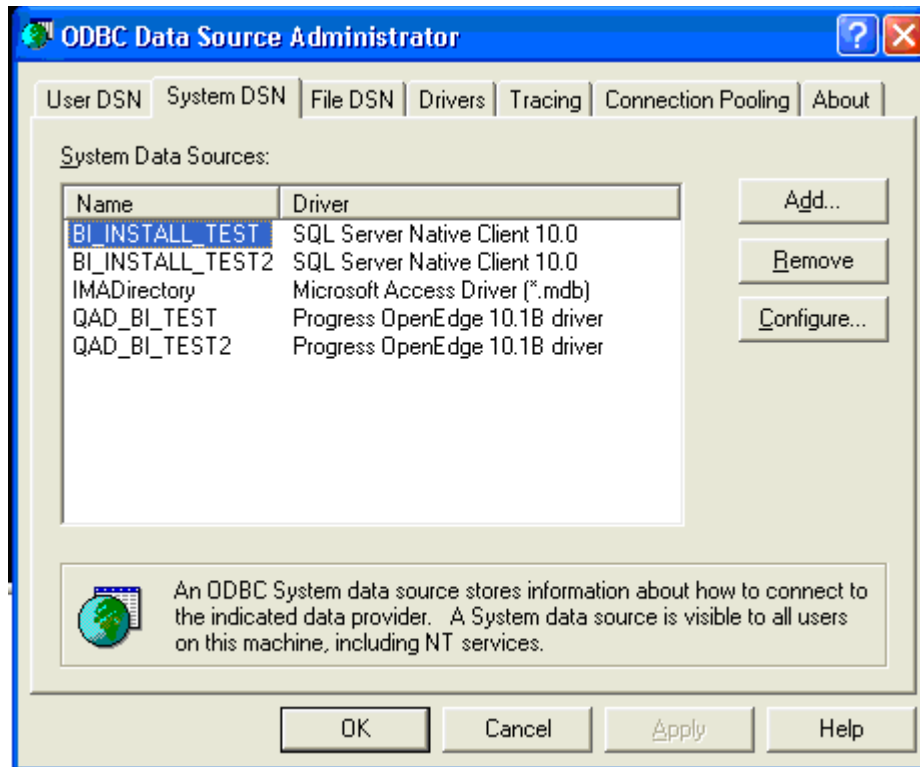
```
ALTER DATABASE TempDB MODIFY FILE
(NAME = templog, FILENAME = 'e:\mssql\data\templog.ldf')
GO
```

- f** For the setting changes to be accepted, stop and start your SQL Server. In SQL Server, go to the server name, not the database, in the top of the left hand pane. Right-click on the server name and select Restart. When the system prompts you to Restart your SQL Server session, click yes. Confirm that your new temp directories now contain files indicating that the database is writing to the correct directory.
- 2** Install QAD BI DWD (6.6.2 or above) if you have not already done so. In Setup Administrator|PC Setup, pick the QAD BI Data Warehouse Designer option and enter the customer and license key.
- 3** Launch the ODBC Data Source Administrator from the QAD BI Administrator so that ODBC connections to the Data Warehouse and to the Progress databases can be created. Using the BI Administrator in the PC Setup tab, select the Validate/Add Odbc Source option. In the resulting ODBC Source screen on the right, right-click and select Add Odbc entry to launch the ODBC Data Source Administrator.

Note ODBC is used by the Data Warehouse Designer (QAD BI DWD) to communicate with the Data Warehouse database. It also allows the DWD to communicate with QAD's Progress databases and extract data from them to be used in the data warehouse. Persons who are familiar with the ODBC Data Source Administrator can launch it through other means, but there is the possibility, in particular if on a 64-bit system, that the correct one is not being launched. The appropriate 32-bit ODBC driver, `odbcad32.exe`, must be used. By opening the ODBC Administrator via the QAD BI Administrator, you eliminate any uncertainty of which ODBC Administrator is correct.

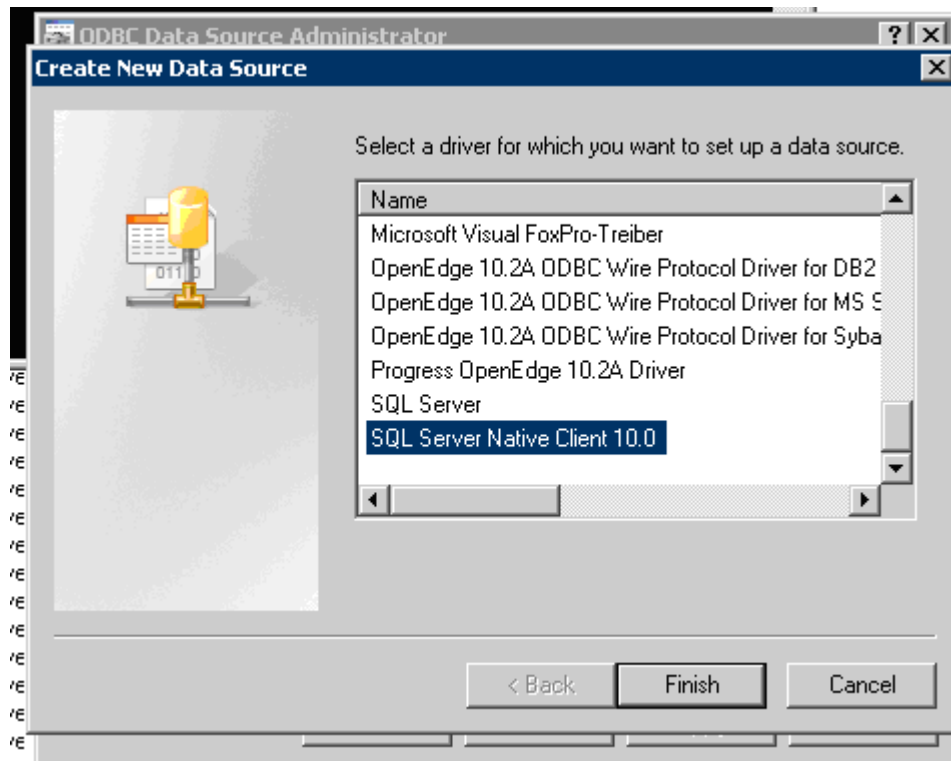
- 4 Create the ODBC connection to the SQL Server Database. This is a System ODBC connection, using the SQL Server Native Client driver and Windows Authentication:
 - a Select the System DSN tab and click Add.

Fig. 1.5
Create ODBC SQL1



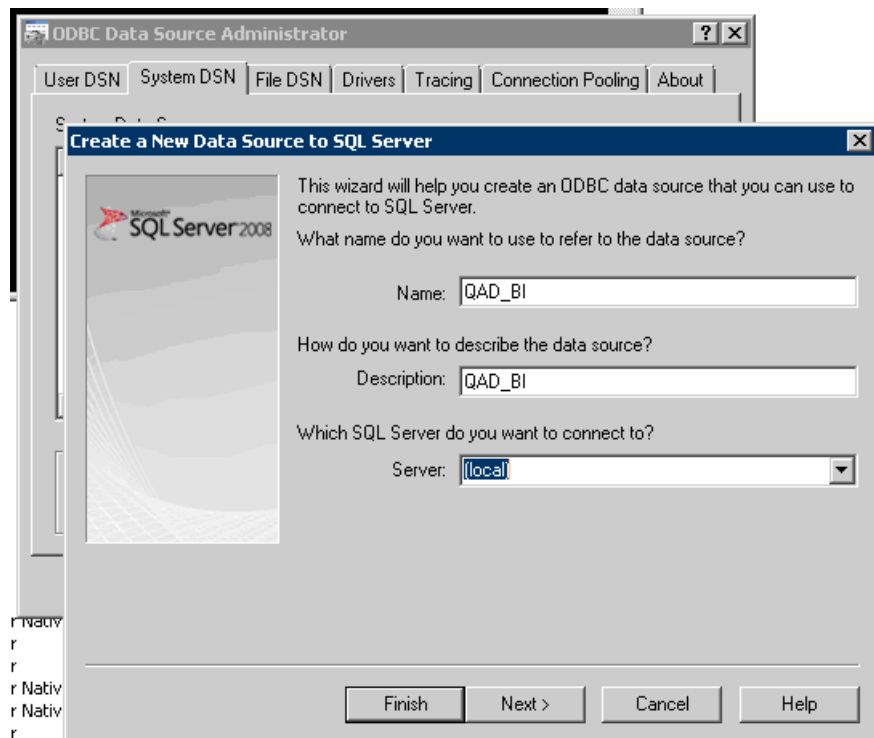
- b When Create New Data Source is displayed, select SQL Server Native Client 10.0 and click Finish.

Fig. 1.6
Create ODBC SQL 2



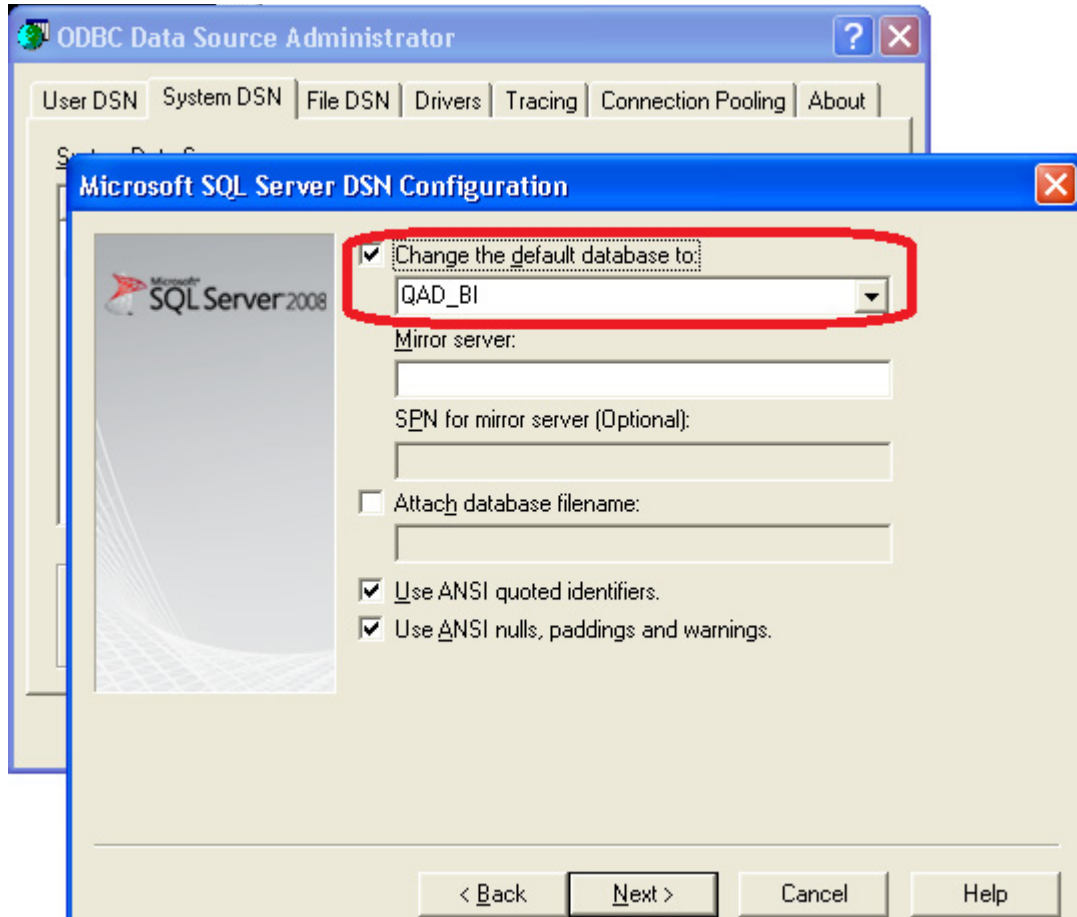
- c Enter the Name, Description and Server for the Data Warehouse and click Next.

Fig. 1.7
Create ODBC SQL 3



- d Choose the option With Integrated Windows authentication unless otherwise directed.
- e Select the *Change the default database to:* check box and change the database from master to the data warehouse database. Click Next, then click Finish.

Fig. 1.8
Create ODBC SQL 4



- f Click Test Data Source.
 - g Confirm that the database connection works properly. If successful, click OK.
- 5 If they do not already exist, use the Progress ODBC driver to create the ODBC connections to the Progress database that the data warehouse extracts from. ODBC is used by the Data Warehouse Engine (QAD BI DWD) to access each Data Source. You need the following information:
- If they have a preferred connection name
 - Host name
 - Port number
 - Database name
 - User ID
 - Password

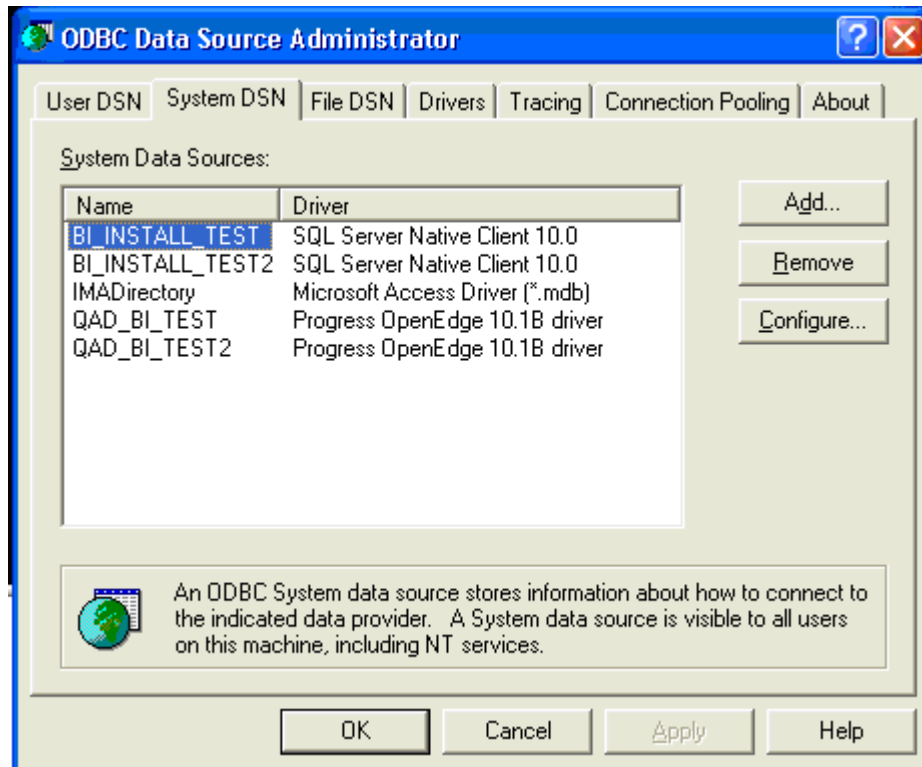
Note that the port must be for a broker that accepts SQL connections. The user ID must have read permissions on the QAD tables.

- 6 If the Progress ODBC connects already exist, go to steps d through f to test the connection and confirm that the Default Isolation Level is set to READ UNCOMMITTED on the Advanced tab of the DSN configuration.

In the ODBC Administrator:

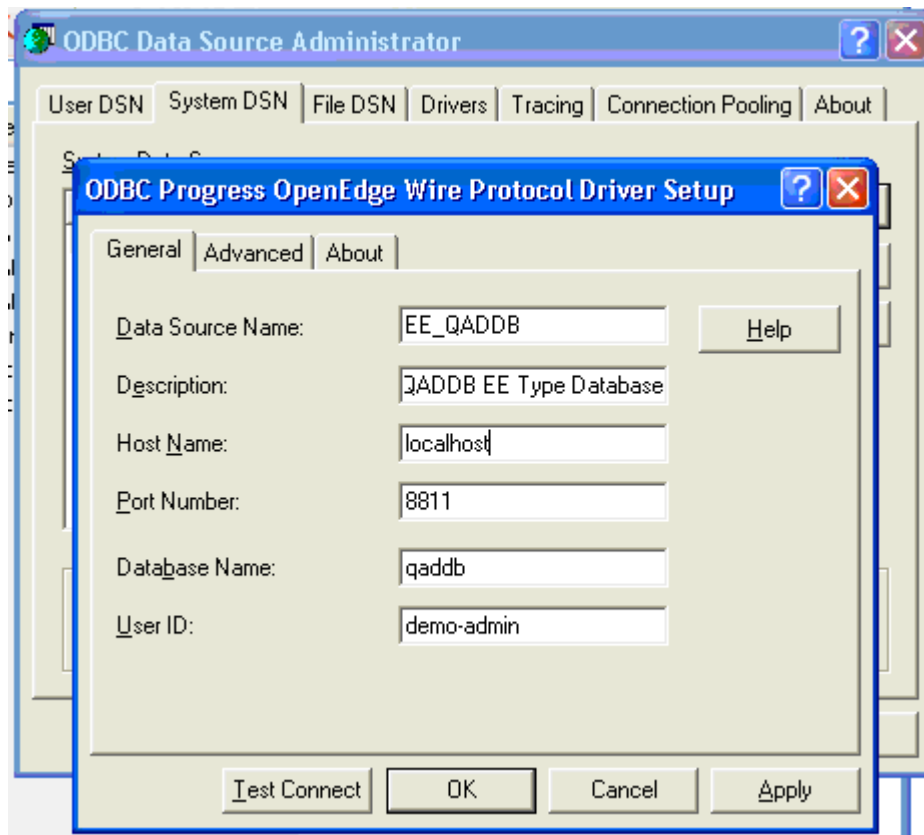
- a Click the System DSN tab and click Add.

Fig. 1.9
Create ODBC Progress 1



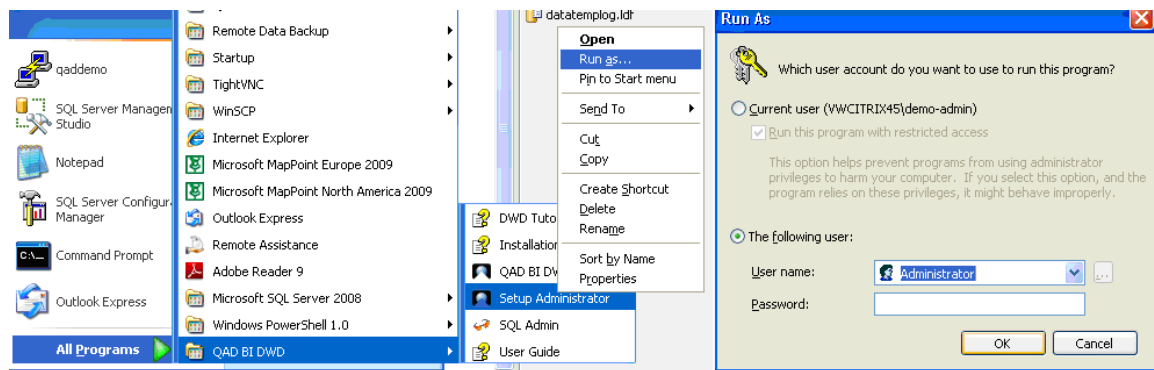
- b Pick the driver type Progress OpenEdge 10.1B driver. Click Finish.
- c Input the values provided to establish the connection to the Progress database.

Fig. 1.10
Create ODBC Progress 3



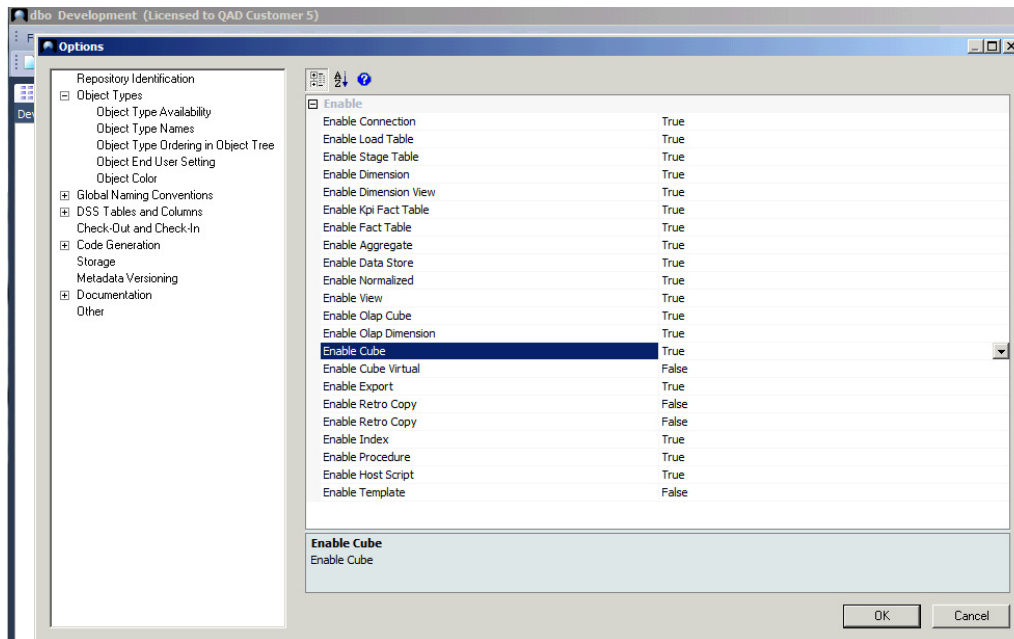
- d** Click Test Connect.
 - e** Enter the Progress database password and click OK.
 - f** Click the Advanced tab and set the Default Isolation Level to READ UNCOMMITTED.
- 7** Set up any additional ODBC connections to outside data sources, if the client requires any. This can be done later if that information is not yet available.
 - 8** If you are using Windows 2008 Server, close the Setup Administrator. Relaunch the Setup Administrator as an Administrator, so that it is able to create the Scheduler as a Windows Service.

Fig. 1.11
Setup Administrator



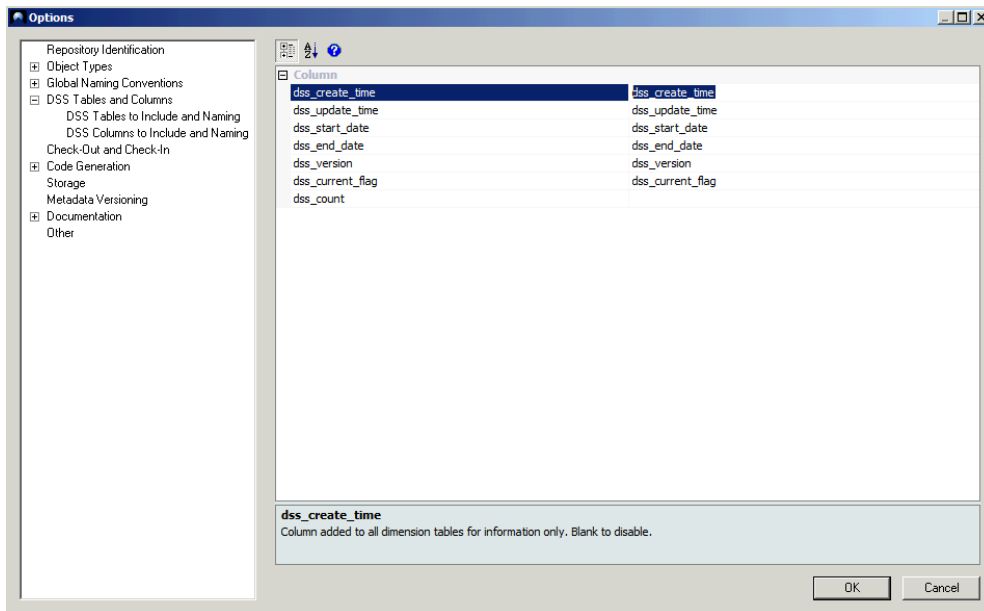
- 9 Choose Metadata Repository, and then Create Metadata Repository. Select the ODBC connection created in the previous step. Under Define FileGroups, leave them blank. The Repository Type should be All Licensed Objects.
- 10 When QAD BI DWD is launched after the repository is created, activate Cube objects. From the Tools navigation menu choose Options|Object Types|Object Type Availability. Click on Enable Cube and select True from the drop-down list.

Fig. 1.12
Cube



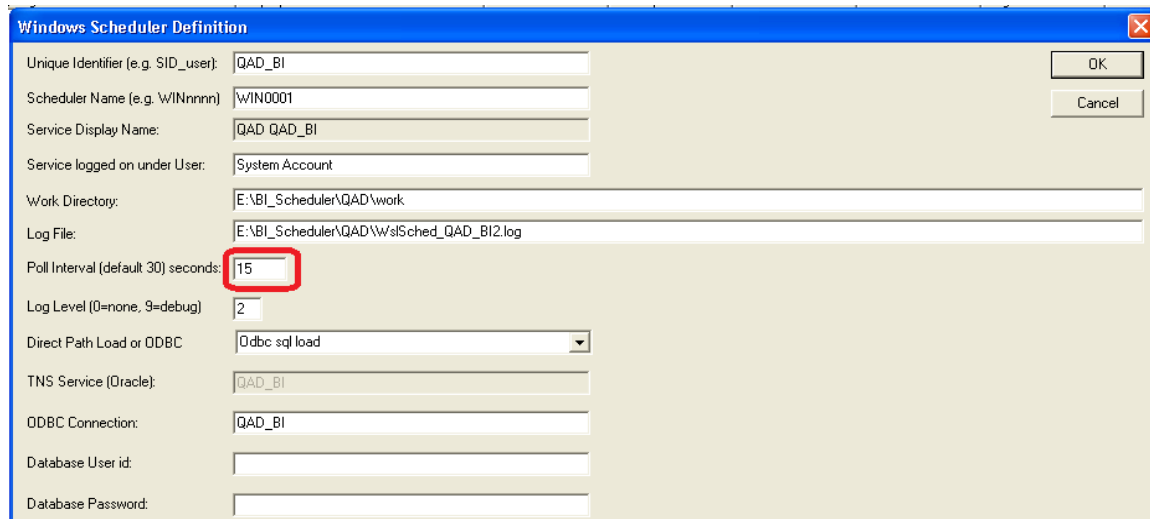
- 11 While in the Options window, choose DSS Tables and Columns|DSS Columns to Include and Naming. Enter dss_create_time in the dss_create_time field and then click OK to close the Options dialog box.

Fig. 1.13
DSS Columns Setting



- 12** In the Setup Administrator, create a Scheduler for the Data Warehouse. This is the Windows service that runs scheduled jobs. Use the ODBC connection that was created for the Data Warehouse. Set the Poll Interval to 15 from the default of 30 seconds. Any time less than 15 seconds raises the possibility of overloading the scheduler jobs if they do not gracefully close quickly after use. Thirty seconds is acceptable, but due to the large number of scheduled jobs we have, reducing this is a big time saver.

Fig. 1.14
Poll Interval



Installation of QAD BI Analytical Modules

This topic includes installation and configuration steps for the following QAD Business Intelligence analytical modules:

- Financials
- Enterprise Asset Management
- Order Management
- Transportation Management
- Operations

Requirements

- QAD BI DWD (version 6.6.2 or later), Data Warehouse installed and Scheduler running.
- QAD Business Intelligence Metadata installation files, in the form of QAD BI DWD application files.

Note This consists of one set of application files for each analytical module to be installed, plus the Common set of application files.

Installation Steps

Follow the steps below to install the QAD Business Intelligence metadata modules.

The previous installation task created the System ODBC connections to the QAD Progress databases. Now, for each one of these ODBC connections, in the QAD BI DWD in the Browser pane, create a Connection object for each:

- 1 Right-click Connection in the QAD BI DWD Metadata Browser and choose New Object.
- 2 Enter a short descriptive name for the data source; for example, QAD_MAIN, QAD_US, QAD_EMEA, or QAD_ASIA. Click OK;
- 3 Select a Connection Type of ODBC and select the ODBC connection.
- 4 Enter a work directory. This directory should be different for each QAD Data Source. Create this directory in the file system. You will create a matching directory in step 6.
- 5 Enter the Extract user ID and password required for the Progress database. This user must have read permission for the QAD tables.
- 6 Using Windows Explorer, create a directory to match the directory you entered as the Work Directory for the connection you just set up.
- 7 Repeat all steps for each data source.

Fig. 1.15
Object Connection Definition

QAD_CHINA definition	
Connection Properties	Notes
Data Warehouse	☐
Connection Name	QAD_CHINA
Connection Type	Odbc
ODBC Source	QAD_CHINA
Work Directory	e:\data_source\qad_china
Extract User ID	root
Admin User ID	
Extract Password	
Admin Password	
SSIS Connect String	
Data Type Mapping Set	(Default)

Install BI Analytical Modules

- 1 The QAD Business Intelligence analytical modules are packaged as zip files. The zip files contain metadata files for the Common module and all necessary analytical modules. Unzip the contents of the .zip files into a directory that you reference for the installation.

Note The Common module must be installed before installing any of the analytical modules except TMS. The Common module contains core components used by all modules except TMS.

Note For users installing TMS only, you do not need to install the Common module application files into the DWD. If you want to install the technical module documentation or any translation files, you need to obtain these items from the Common module media.

- 2 In Setup Administrator, confirm that the job scheduler has started. If the job scheduler has not started, start the service and confirm that it does not stop.

Note The job scheduler is created when the data warehouse is created.

- 3 In the directory where you extracted the Common module, there is a subdirectory named `sqlserver`. In this directory, there is a SQL script named `misc.sql`. In the SQL Server Management Studio, open `misc.sql`, change the database to your Data Warehouse, and execute the `misc.sql` script. This adds a new tab called Technical Info for the following types of objects:
- Dimension Column
 - Dimension View Column
 - Fact Table Column
 - Kpi Fact Table Column
 - Datastore Table Column
 - View Column
- 4 These objects can be seen from the DWD under Tools|Options|Documentation.

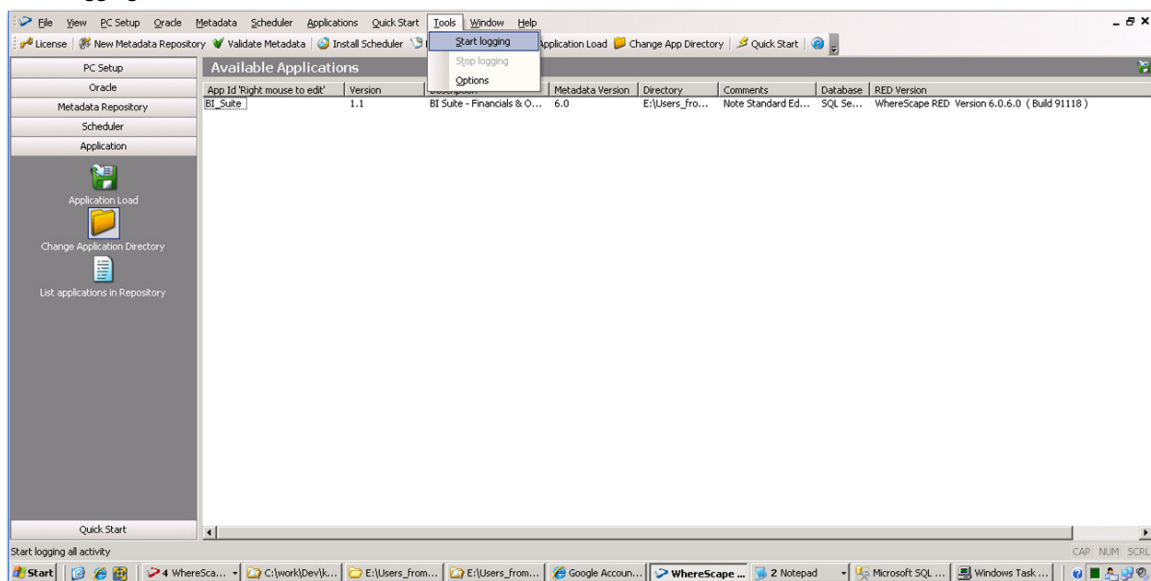
Install BI Common Module

- 5 Currently the Common module is required if you are installing any of these analytical modules:
- Order Management (OM)
 - Financials (both SE and EE) (FIN)
 - Operations (OPS)
 - EAM

Note Skip this section and see “Install BI Analytical Module” if the analytical module you are installing does not require the Common module.

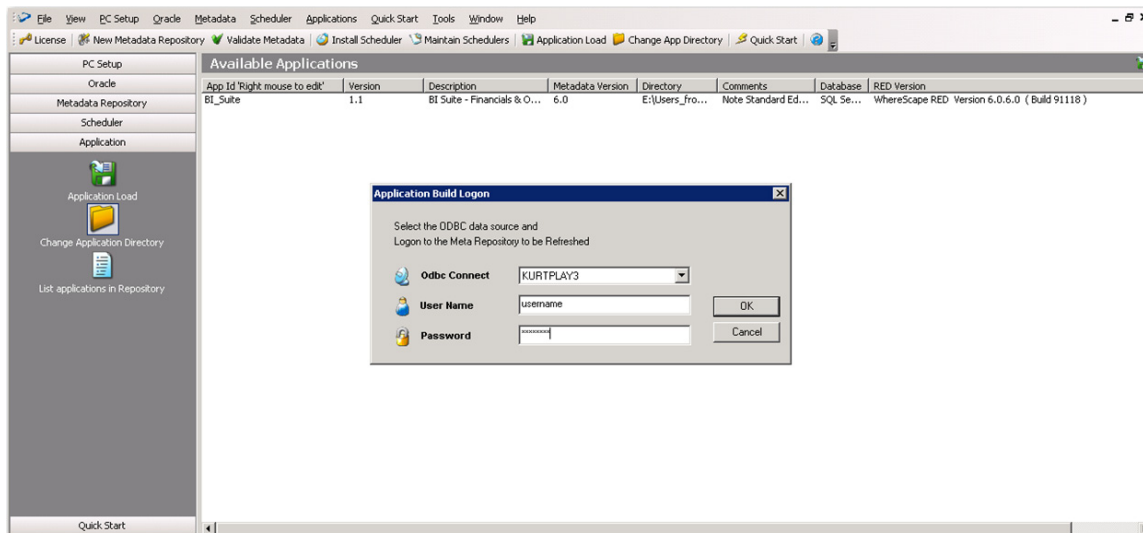
- 6 Go to the Application option in Setup Administrator. Change Application Directory to the directory into which you extracted the QAD Business Intelligence Common Module metadata. The contents consist of a series of `.wst` files.
- 7 Go to Tools and Start Logging.

Fig. 1.16
Start Logging



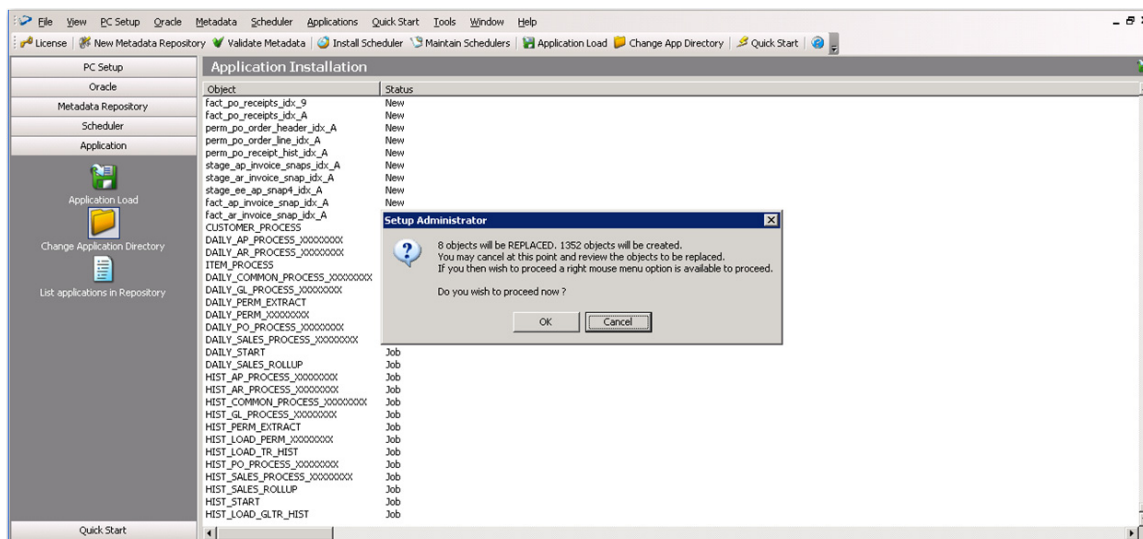
- 8 Choose a file name for your log and a directory for it to write to.
- 9 Install the application through Application Load. Right-click on the Common module and choose Install.

Fig. 1.17
Application Load



- 10 If a message appears saying that files may be overwritten, choose Cancel and confirm that the files are OK to be overwritten. If the files can be overwritten, go back to step 6 and repeat.

Fig. 1.18
Confirm File Overwrite

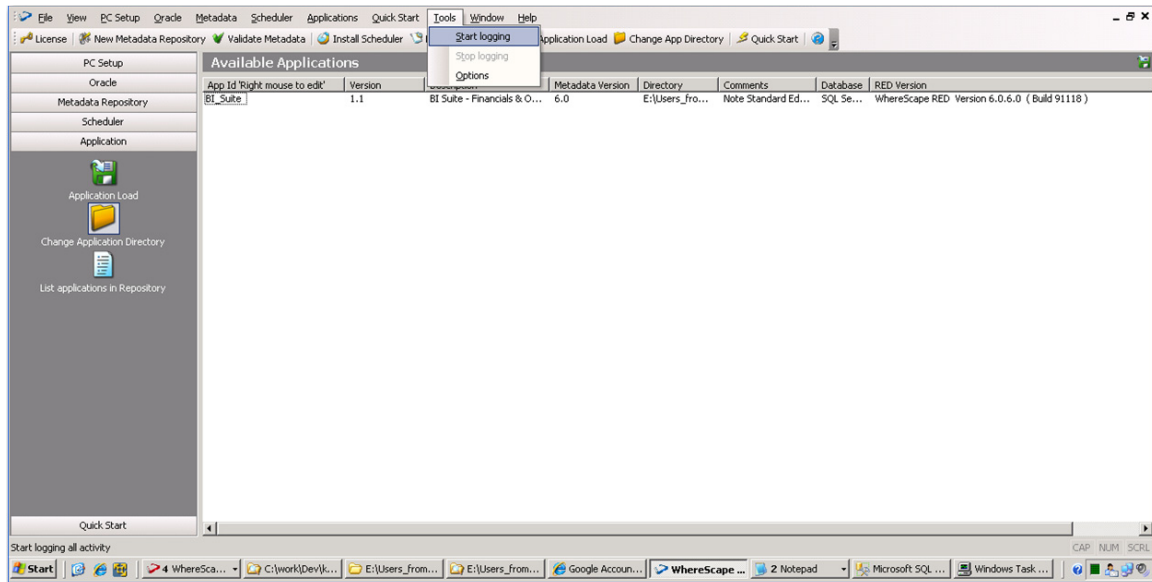


Install BI Analytical Module

- 11 Next, go back to the Application option in Setup Administrator. Change the Application Directory to the directory into which you extracted the first Analytical Module you want to install.

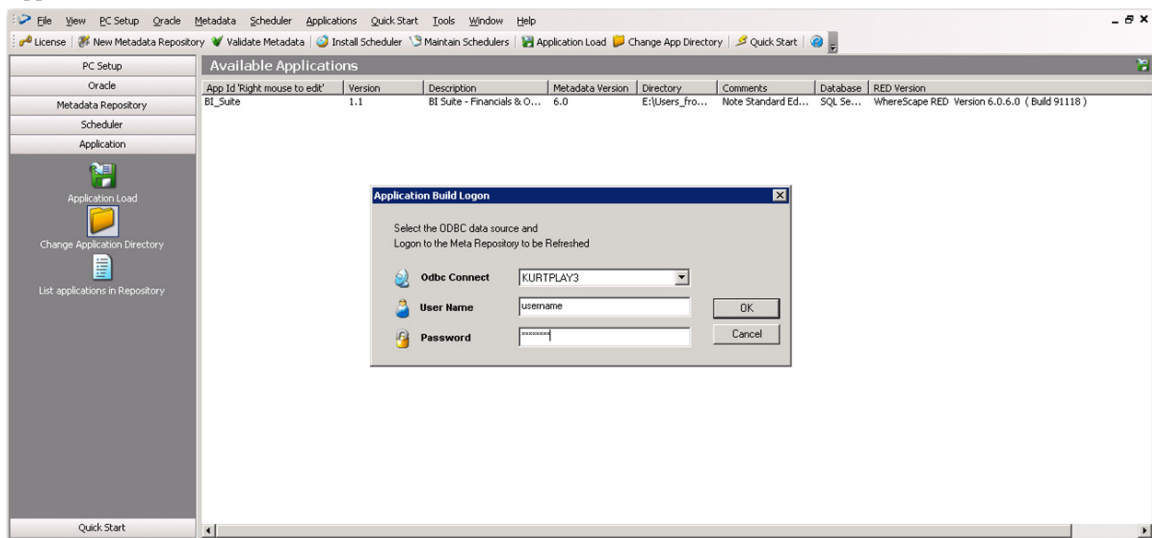
- 12 Select Start Logging from the Tools drop-down menu.

Fig. 1.19
Start Logging



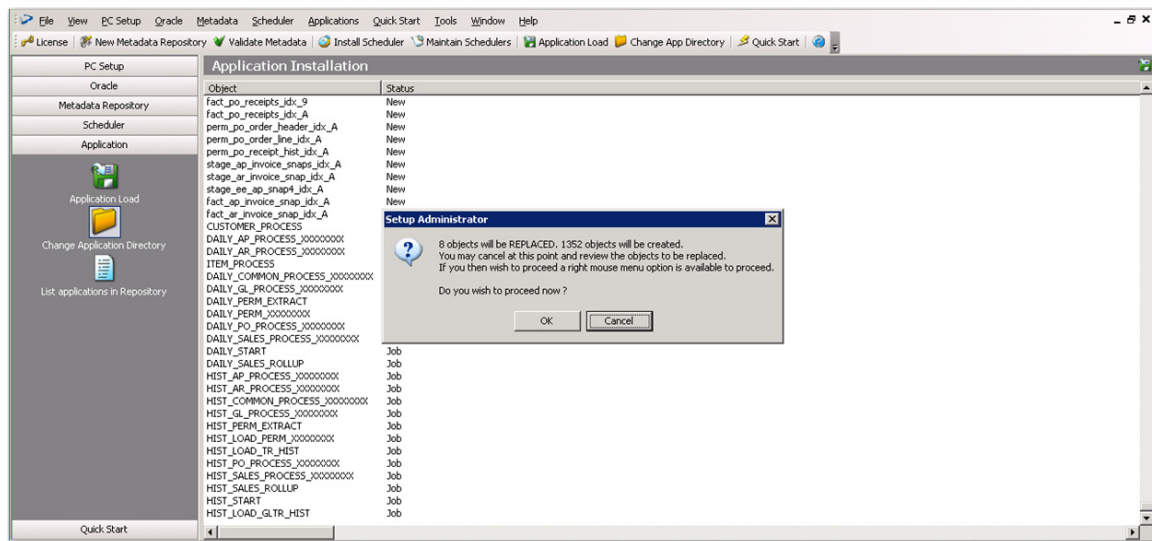
- 13 Choose a file name for your log and a directory for it to write to.
- 14 Install the application through Application Load. Right-click on the desired module and choose Install.

Fig. 1.20
Application Load



- 15 If a message appears about files being overwritten, choose Cancel and confirm that the files are OK to be overwritten. If the files can be overwritten, go back to step 11 and repeat.

Fig. 1.21
Confirm File Overwrite



16 Repeat for any other modules that are required.

17 Review the message log. The files that are replaced are identified at the top of the log. To proceed, right-click within the log and choose Proceed to Load. Once the main application load is running, the operation may take as long as 5-10 minutes.

Note If the top of the DWD screen displays *Not Responding*, wait up to 30 minutes for the job to finish. If it takes any longer, close the screen and start over.

18 After loading all the module components, you can refresh the DWD by clicking on All Objects in the Browser pane and pressing F5. If this does not work, you can log off and log back into the DWD.

Initial Parameter Setup

Note In QAD BI DWD, choose Parameters from the Tools menu to display the current Parameter definitions. These need to be customized for your specific installation.

19 Set runtime parameters to the proper settings based on the customer feedback to the parameters (Table 4.9 on page 89). Also refer to “Jobs” on page 44 and Table 4.1 on page 47 for additional assistance. In particular, change the HIST_LOAD_SOURCE_Sxx, DAILY_LOAD_SOURCE_Sxx and INITIAL_JOB_SETUP_CONNECTION_xx parameters to point at the source databases.

Note Parameters that MUST be set before running INITIAL_JOB_SETUP should have an initial value of <some value between these symbols >. See Chapter 4, “Reference,” on page 43 for more details.

Initial Job Setup

Note The INITIAL_JOB_SETUP job will create the proper chained job flow for all the source connections. This is not used for the Transportation Management (TM) module. For more information, see “TM Module Setup” on page 23.

- 20** Set the INITIAL_JOB_SETUP_CONNECTION_01 parameter to your first QAD ERP source.
- 21** Populate the INITIAL_JOB_SETUP_CONNECTION_01_TYPE parameter with the type of source system (EE, SE) in order for the chained job flow to be set up with correct jobs for the source type.

Note If you are using the EAM analytical module, the EAM system cannot be your first source. A QAD ERP source must precede the EAM source.

- 22** Populate the INITIAL_JOB_SETUP_CONNECTION_01_DAILY_MODULES with the list of analytical modules you want to include in your first source's processing. Typically the list includes all your analytical modules for an ERP source. If you have more than one module installed, list them in the order in which you want them processed, each separated by a comma. The Common module must be listed first for ERP sources.

The valid module identifiers for this list are:

- **COM.** For the Common module, ERP sources
 - **OP.** For the Operations module, ERP sources
 - **FIN.** For the Financials module, ERP sources
 - **OM.** For the Order Management (Sales) module, ERP sources
 - **EAM.** For the Enterprise Application Module – EAM sources only
- 23** Populate the INITIAL_JOB_SETUP_CONNECTION_01_HIST_MODULES with the list of analytical modules you want to include in your first source's historical loads. Typically the list includes all analytic modules the first time you install.

Note If you install an additional analytical module later, you can list just that module and re-run INITIAL_JOB_SETUP. This method creates a Historical job flow that loads just the new module, so you don't need to repeat any historical loads for other modules.

Finally, set INITIAL_JOB_SETUP_CONNECTION_01_RUN to N. This indicates you have not run a historic load for this source yet.

- 24** If you have more than one source system, set INITIAL_JOB_SETUP_CONNECTION_02 to your second connection (EE, SE, or EAM) and set INITIAL_JOB_SETUP_CONNECTION_02_TYPE appropriately.

Note If you are loading an EAM source, a valid QAD ERP source must come before the EAM source in the source list.

- 25** You can continue adding as many INITIAL_JOB_SETUP_CONNECTION_XX* and parameters as necessary. Make sure you add all the required parameters for each connection:
- INITIAL_JOB_SETUP_CONNECTION_XX
 - INITIAL_JOB_SETUP_CONNECTION_XX_TYPE
 - INITIAL_JOB_SETUP_CONNECTION_XX_RUN
 - INITIAL_JOB_SETUP_CONNECTION_XX_DAILY_MODULES
 - INITIAL_JOB_SETUP_CONNECTION_XX_HIST_MODULES

- 26** Eliminate the values for the connections that are not being used.

- 27** Set the INITIAL_JOB_SETUP_ENABLED parameter to Y, and in the Scheduler, execute the INITIAL_JOB_SETUP job.

Setup Steps:

- a In QAD BI DWD, choose the Scheduler menu option.
 - b Change the view to All Jobs.
 - c Locate the job named INITIAL_JOB_SETUP (status On Hold).
 - d Right-click the job name and start the job.
 - e The status will change to Waiting.
 - f Refresh the screen until the job completes.
 - g If the job fails to start, check the scheduler status.
- 28 After INITIAL_JOB_SETUP has run, check Scheduled jobs to confirm that you now have SET_CONNECTION jobs with names that end with all connections you have created. If they have not been created, check the Parameters. Make sure that the INITIAL_JOB_SETUP_ENABLED parameter is set to Y and run INITIAL_JOB_SETUP again.
- 29 Verify that your parameters are all set properly for your installed modules. See Table 4.9 on page 89.
- 30 Some modules require some special settings for the History load as opposed to the Daily load. For instance, GL_LOAD_DAYS and GL_LOAD_LOOKBACK_DAYS should be set to a value large enough to capture all the data you require for historical purposes. Once the History load has completed and loaded the historical data, set GL_LOAD_DAYS and GL_LOAD_LOOKBACK_DAYS to smaller values so as not to extract too much data with each daily load. They should be set just large enough to capture any GL transactions you might back-date. GL_LOAD_DAYS should always be larger than GL_LOAD_LOOKBACK_DAYS. Typically GL_LOAD_DAYS would be set to GL_LOAD_LOOKBACK_DAYS + 2.
- 31 Confirm that your DEFAULT_* parameters are set properly. In particular, the DEFAULT_ACCOUNT and DEFAULT_SUB_ACCOUNT values must not be actual account values in any of your ERP sources.
- Important** Failure to confirm that these parameter values are not duplicates of actual account values may result in errors in the history load, requiring reloading of GL history.
- 32 You are now ready to populate the Data Warehouse with your ERP history. In the scheduler, find the HIST_START job, right-click on it, and select *Start the Job*. Depending on the size of the source tables, this can take quite some time. It does not hurt to check in throughout this load process to make sure that the jobs are all running successfully.

Install Metadata Language Translation Files

Install the metadata language translations files, if any are required.

- 33 In the BI DWD tool, choose Tools|Options.
- 34 Click on the Languages tab and add the following codes for the language files you want to install:

Language	Code
Czech	cs
German	de
Spanish	es
Spanish (Mexico)	es_MX
French	fr
Italian	it
Japanese	ja
Korean	ko
Dutch	nl
Polish	pl
Portuguese (Brazilian)	pt
Chinese (Simplified)	zh
Chinese (Traditional)	zh_TW

- 35** The complete set of files are in the metadata package, under metadata/translations. The file names are in the format lang_<code>.wst. For example, the file for traditional Chinese is lang_zh_TW.wst.

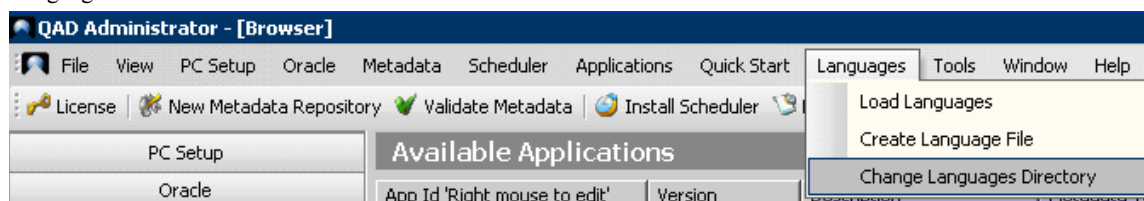
Note The UI can be a little misleading here because you need to click Add Language and then type in a language or abbreviation in the Language drop-down menu. The code you enter helps you find language options in the Analysis Services Languages drop-down list, but the language that displays in the drop-down list may not be the language you want. The code you enter for the Language is simply used for reference purposes in later steps. For example, if you enter *es*, the language code for Spanish, the Analysis Services Language drop-down list displays Estonian, because that starts with *es*. If this happens, click on the Analysis Services Language drop-down list, scroll down, and select the Spanish option.

Fig. 1.22
Language Install

The screenshot shows the 'Options' dialog box with the 'Languages' tab selected. The 'Language' dropdown menu is set to 'SP'. Below it, the 'Language Description' field is empty. The 'Analysis Services Language' dropdown menu is set to 'Spanish (Mexico)'. There are buttons for 'Add Language' and 'Delete Language' on the right side of the dialog.

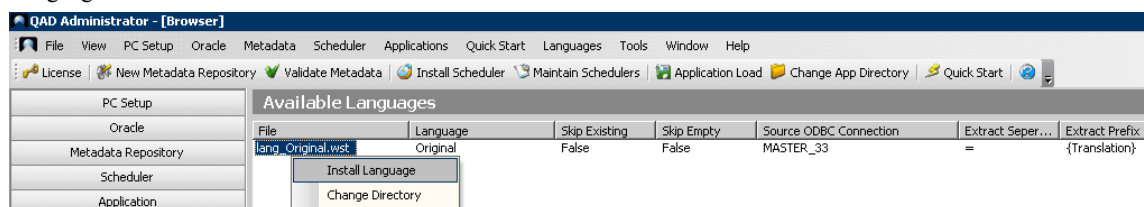
- 36** After adding the language codes, launch the QAD BI Data Warehouse Setup Administrator.
- 37** Select Languages|Change Languages Directory. Navigate to the directory metadata/translations directory of your package.

Fig. 1.23
Language Install 2



- 38 Choose Languages|Load Languages. Right-click on the language file you want to install and choose Install Language.
- 39 Select the Data Warehouse's ODBC DSN. Then choose the language to which you want to map this language file. You are allowed to choose one of the language codes you entered in the QAD BI DWD in step 34.

Fig. 1.24
Language Install 3



- 40 Repeat these steps for each language you want to install.

Creating and Processing Cubes

BI DWD provides the ability to create cubes in SQL Server Analytical Services (SSAS) from any of the fact tables.

- 41 Some of your analytical modules come with cube metadata already created. To use these cubes, you must first create them in SSAS.
- 42 Make sure that the DataWarehouse connection in the DWD has the As a Cube Data Source section of its Properties tab filled out correctly:
 - The Provider Type is set to SQLOLEDB.
 - The Server and Database fields contains the name of the SQL Server instance and the name of the DWD SQL Server Database.
 - The QAD_BI_Cube connection is set to the correct instance of SSAS.
- 43 Locate the cube in the Metadata Browser. Right-click on the cube and choose Properties.
- 44 Ensure the following:
 - The Cube Database Connection is set to your QAD_BI_Cube connection object.
 - The Data Source Connection is set to your DataWarehouse connection object.
- 45 Use the default Cube Database name or optionally create a different name for the SSAS database. Click OK to save.
- 46 Right-click on the cube and choose Create (Alter) Cube.

Note For the older, non-OLAP cubes, the option is just Create Cube. Verify that this completes successfully.

- 47 The final step is to process the cube, which populates the cube with data from the fact tables. For the analytical modules that come with cubes, there are standard jobs that process cubes; for instance, the job POSTPROCESS_<module>_CUBES.

Schedule this job to run at specific intervals, or add it to the end of the DAILY job chain to run it automatically after the fact tables have been updated by the daily jobs.

Note The *QAD Data Warehouse Designer User Guide* contains additional information on creating and modifying OLAP cubes, including the following:

- A table that describes the types of modifications to an OLAP cube's metadata that can be processed using the *Create (Alter) Cube* action in the Data Warehouse without making any changes in SSAS first.
- Which modifications require the user to delete the SSAS cube database first prior to re-creating the cube from the Data Warehouse.

TM Module Setup

- 1 The Transportation Management (TM) module does not use the same INITIAL_JOB_SETUP logic as the ERP and EAM modules. It has a single pre-set list of jobs that are run in a specific order for each TM source. The important parameters to configure are:

- **TM_LOAD_SOURCE_S01.** Enter the name of the first TMS source.
- **TM_LOAD_SOURCE_SXX.** If you have more than one source, add additional parameters for each source; for example, TM_LOAD_SOURCE_S02, TM_LOAD_SOURCE_S03.
- **TM_LOAD_TRANS_TYPES_S01.** Enter the transaction types you want to load from TMS source, single quotes and comma separators; for example, 'S','T'.
- **TM_LOAD_TRAN_TYPES SXX.** If you have more than one source, add additional parameters for each source; for example, TM_LOAD_TRANS_TYPES_S02, TM_LOAD_TRANS_TYPES_S03.
- **TM_LOAD_SINGLE_SOURCE.** Set this value to Y or N, depending on whether you want to process all source systems in a single extract or just specific source system.
- **TM_LOAD_LOOKBACK_DAYS.** Set this to a large number for the initial data extract. The number should be large enough to ensure that you extract all the data you want to include in the BI system for historical analysis. After the first data load completes, set this to a smaller value in order to limit each daily load to just the period in which there might be new or modified transactions.

- 2 To run the TM load, start the job DAILY_START_TM.

- 3 The TM module is set by default to handle SQL Server-based sources. If your TMS system is based in Progress or Oracle, run the TM_ConvertSource job. Set these parameters prior to running the job:

- **TM_SourceConvert_SchemaORA.** If your source is in Oracle, enter the schema name. Note that for Progress databases, the schema is assumed to be *PUB*.
- **TM_SourceConvert_TargetSource.** Enter *oracle* or *progress*.

Add New Source to Existing Installation

Follow these steps to install a new source system if you have been running your system for some time:

- 1 Create a new ODBC DSN for your new source. In DWD, set up a new Connection object pointing to your new source DSN.
- 2 Add new `DAILY_SOURCE_SXX` and `HIST_SOURCE_SXX` values set to your new source connection.
- 3 Set up a new set of `INITIAL_JOB_SETUP_CONNECTION_XX` parameters for your new source, setting `XX` to the next available source number:
 - `INITIAL_JOB_SETUP_CONNECTION_XX` – name of source connection
 - `INITIAL_JOB_SETUP_CONNECTION_XX_TYPE`
 - `INITIAL_JOB_SETUP_CONNECTION_XX_RUN` – set to N
 - `INITIAL_JOB_SETUP_CONNECTION_XX_DAILY_MODULES`
 - `INITIAL_JOB_SETUP_CONNECTION_XX_HIST_MODULES`
- 4 Verify that all other `INITIAL_JOB_SETUP_CONNECTION_XX_RUN` parameters for the source are set to Y so that their history loads will not be rerun.
- 5 Set `INITIAL_JOB_SETUP_ENABLED` to Y.
- 6 Run `INITIAL_JOB_SETUP`.
- 7 Run `HIST_START`. Verify that the new source is the only one being loaded.
- 8 After the new source's history has been loaded, daily processing can resume.

Adding New Module to Existing Installation

Follow these steps if you have been running a system for a while with a specific set of analytical modules and now need to add a new module and load the data for all your sources:

- 1 Install the new module. See “Install BI Analytical Modules” on page 14.
- 2 For each source that now needs the history loaded for this module only, set the `INITIAL_JOB_SETUP_CONNECTION_XX_HIST_MODULES` to include COM and JUST THE NEW MODULE. For example, if you had previously installed COM, OM, and FIN, the value for this parameter should be “COM,OM,FIN”. If you want to add the Operations module, set the `HIST_MODULES` value to “COM,OPS”.

Important Do not include any modules for which you have already loaded the history.

- 3 For each source that you want to process the new module's data on a daily basis, add the new module to the `INITIAL_JOB_SETUP_CONNECTION_XX_DAILY_MODULES` list. For example, if you are processing COM,OM,FIN on a daily basis, and are adding the Operations module, change the `DAILY_MODULES` list to “COM,OM,FIN,OPS”.
- 4 For each source that needs history loaded for the new module, set the `INITIAL_JOB_SETUP_CONNECTION_XX_RUN` value to N.
- 5 Set the `INITIAL_JOB_SETUP_ENABLED` flag to Y.

- 6 Run INITIAL_JOB_SETUP.
- 7 Run HIST_START.
- 8 When the history load completes, daily processing can resume.

Installing QAD BI Portal Web Application

This chapter provides information about the installation and configuration steps for the QAD BI Portal Web application.

It discusses the following topics:

Requirements 28

Lists requirements for installation.

Configuration Parameters 28

States configuration parameters.

Installation Steps 28

Lists the installation steps for the BI Portal Web Application.

Requirements

- Tomcat 5.5, 6, or 7 installed and configured
- Java 1.5 JRE or higher
- Java 6 JDK or higher
- SQL Server 2008 installed and configured. SQL Server 2005 is acceptable, but not recommended due to poor performance.
- SQL Server has TCP enabled, in order to connect via JDBC
- QAD BI Data Warehouse Designer installed and configured
- Data Warehouse created in SQL Server. In the QAD BI DWD: QAD BI Analytical Modules
- QAD BI DWD, customizations made, data loaded, Scheduler created, desired language translations installed

Note End users need Adobe Flash Player installed for their browser.

Configuration Parameters

The following configuration parameters are required during installation. Write them down before you start:

- **Tomcat Port.** Port in which Tomcat is running; for example, 80 or 8080.
- **SQL Server JDBC URL.** URL required to connect to SQL Server database (host name, port number, and database name).
- **QAD BI Data Warehouse Designer user / pwd.** User name and password to connect to SQL Server Database where QAD BI DWD is running. This should be the QAD BI DWD owner account.
- **BI Portal database User / pwd.** User name and password to connect to SQL Server database where BI Portal repository is installed.

Installation Notes

Note that the QAD BI Portal cannot be installed in the same Tomcat instance as QXtend. QXtend is forcing the use of a specific DOM parser library that is incompatible with the BI application usage. Install the two applications in separate Tomcat instances.

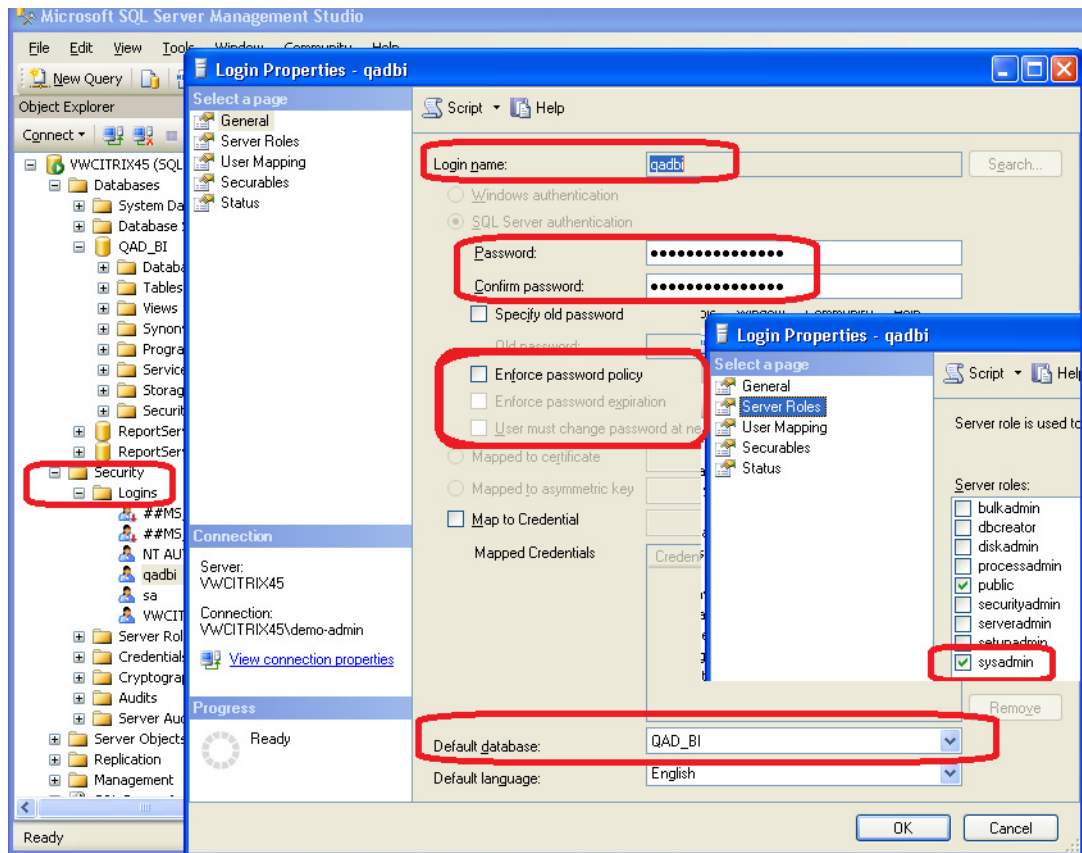
Installation Steps

Follow the steps below to install and configure a new BI Portal instance:

- 1 As part of the Portal Installation, set up a new user in the database. Go to the SQL Server Management Studio.
- 2 Select Security\Logins. Right-click Logins and select New Login.
- 3 When the Login - New screen comes up, enter the new login name; for example, qadbi.
- 4 Select SQL Server authentication and make sure that the Enforce password policy box is not selected.

- 5 Change the database from master to the database you are working with.
- 6 Then select the Server Roles, on the upper left-hand side of the window, and make sure that the Server roles public and sysadmin are both selected.

Fig. 2.1
Login Properties



Install BI Portal Web Application

- 7 Create a subdirectory for the Web application below the `<tomcat_dir>\webapps\` directory; for example, qadbi.
- 8 With Tomcat not running, unzip qadbi.zip into the Tomcat webapps sub-directory created above. To shut down Tomcat, in Windows, go to Start/Administrative Tools/Services and find Apache Tomcat in the list of running services. Right-click on it and choose Stop.
- 9 Configure Tomcat's `server.xml` file, which can be found in the `/conf` directory of Tomcat. Add the `maxPostSize` directive, which is case sensitive, and set it to 0:

```
<Connector port= 8888
protocol= HTTP/1.1
connectionTimeout= 20000
redirectPort= 8443
maxPostSize= 0 />
```

Configure BI Portal

- 10 Copy the XML file `<tomcat_dir>\webapps\qadbi\WEB-INF\config\server-config.xml.default` to a file named `server-config.xml` in the same directory.
- 11 Edit the `server-config.xml` file to match the values for the configuration parameters to the appropriate values for your environment, as described below.

Contents are as follows:

- repository Element:
 - **dbType.** Database platform. Value should be SQL for SQL Server.
 - **driver.** JDBC driver to be loaded at runtime for business intelligence server database. For SQL Server, the value is `com.microsoft.sqlserver.jdbc.SQLServerDriver`.
 - **jdbcUrl.** JDBC URL to connect to Business Intelligence Portal database
 - **user.** User name for the database
 - **pwd.** Password for the database
 - **logCheckInterval.** How often, in minutes, the portal should perform its maintenance tasks; for instance, log the system state, clean up logs, and so on. Default is 15 minutes.
 - **logDaysToKeepEvents.** Number of days to store system events in the database; `aw_evt_system` and `aw_evt_request` tables. Default is 14 days.
- log Element:
 - **watch.** Whether to watch the log file for changes and update log4j settings dynamically; true or false.
 - **configFile.** The name of the log4j configuration file.
- Example contents of the `server-config.xml` file:

```
<root>
<repository
dbType=SQL driver=com.microsoft.sqlserver.jdbc.SQLServerDriver jdbcUrl=
jdbc:sqlserver://myhost:1433;database=QAD_PORTAL user=myuser password=mypwd
logCheckInterval=15
logDaysToKeepEvents=14 />
<log watch=true configFile=LoggerConfig.properties />
</root>
```

Configure log4j in LoggerConfig File

- 12 Copy the file `<tomcat_dir>\webapps\qadbi\WEB-INF\config\LoggerConfig.properties.default` to a file named `LoggerConfig.properties` in the same directory. Edit the Log4j configuration file to specify a path for the log file that is the R1 appender defined in the properties file. Note that if you are running Tomcat as a Windows Service, the log file name must be a full path, and the backslashes must be escaped; for example,

```
log4j.appender.R1.File=D:\\apache-tomcat-6.0.20\\logs\\qadbi.log
```

Install BI Portal Repository

- 13 Connect to the SQL Server database where BI Portal Repository is to reside, using the SQL tool of choice.

- 14 Locate the BI DDL script in the `<tomcat_dir>\webapps\qadbi\install\sqlserver` file.

- 15 Run the `bi3_ddl.sql` SQL script to create required repository tables.

Note This script drops the tables first if they already exist and re-creates them. Make sure you are running this in the correct database.

- 16 Edit the seed script `seed.sql` to set any variables to the specific values necessary for your installation.

- a Set the e-mail address for the initial root user to the e-mail address of the application administrator. If required, customize the authorization roles, which are the values being entered into `aw_auth_role` table and into `aw_user_auth_role` for the initial root user. Do not modify any of the `aw_app_role` descriptions. The `aw_data_role` table, as well as the `data_role` column in `aw_user`, are no longer used.

- b Set the different `aw_sys_config` entries as follows:

- **DATASERVICE.** Configure with the information required to connect to the DWD data warehouse. This includes the `dbType` (SQL), `whSchema` (the name of the database owner—typically `dbo` for SQL Server), the `jdbcDriver` and `jdbcURL`, `jdbcUser`, and `jdbcPassword` for accessing the database, and the `metadataAdapter` (`com.awbi.server.data.adapter.WhereScapeAdapter`). For example:

```
<dataConfig dbType= SQL whSchema= dbo
jdbcDriver=
com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbcURL= jdbc:sqlserver://myhost:1433;database=QAD_DW jdbcUser= myuser
jdbcPassword= mypwd
metadataAdapter= com.awbi.server.data.adapter.WhereScapeAdapter />
```

- **WELCOME_EMAIL.** Customize the text that is contained in the Welcome E-mail sent to new users. Make sure that the link to the BI Portal Web application is correct for your environment.
- **SMTP and MAILER.** Configure this with your specific mail server parameters.
- **USERSERVICE.** Customize the text that is used in the Subject of the Welcome and Password Change emails as desired.
- **PWD_RESET_EMAIL.** Customize the text that is contained in the Password Reset e-mails sent to users who request a new password.
- **SECSERVICE.** Do not edit; security models are built in the Administration UI.
- **AUTHSERVICE.** If a custom authentication class is used, set the custom `authClass` to the customized authorization class and uncomment this section. Otherwise, leave this section commented out.

- 17 Run the modified `seed.sql` script against your database to load the data into your portal database.

Install Portal Constraint Tables in Data Warehouse

- 18 Connect to the database in which the data warehouse is to reside. This is the database specified in the DATASERVICE key, above. Locate the `bi3_wh_ddl.sql` script in the `<tomcat_dir>\webapps\qadbi\install\sqlserver` directory. Run the `bi3_wh_ddl.sql` SQL script to create the required portal tables in the data warehouse database.

Technical Documentation for the Data Warehouse

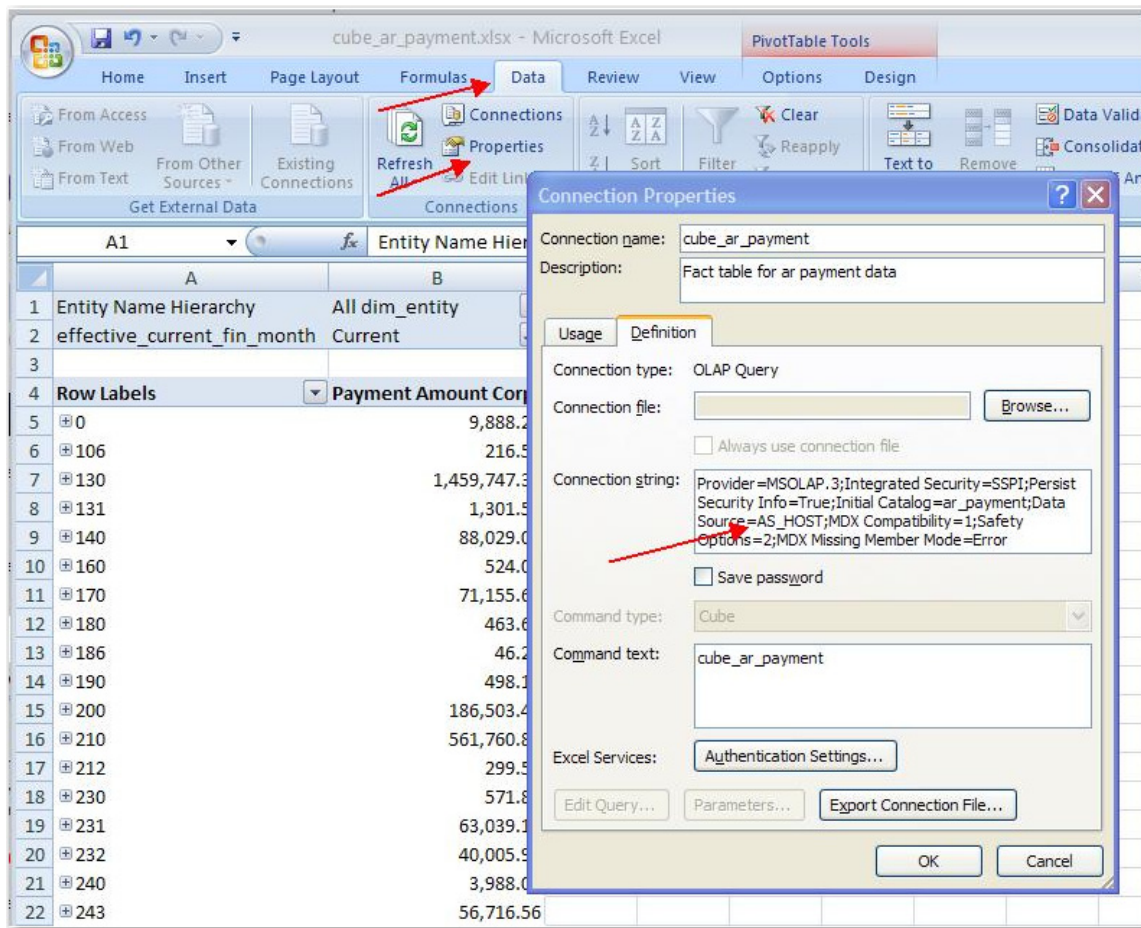
- 19 The Standard Dashboards include links to the technical documentation for the Data Warehouse. This documentation is shipped with the Common analytical modules in the file `qadbi-documentation.zip`. The Standard Dashboards expect them to be located in a directory named `documentation` within the web application. To deploy the documentation, therefore, create a directory `<tomcat_home>/webapps/<application name>/documentation` and extract the `qadbi-documentation.zip` file into that directory.

Installing Excel Workbooks

- 20 Some of the standard dashboards for some analytical models include Excel workbooks that connect to an Analysis Services system for ad-hoc analysis. Create a new Web application directory, `<tomcat_home>/webapps/<application name>/published`. Extract the file `<module_name>_adhoc_excel_files.zip` into that directory. These files are referenced by the standard dashboards loaded in the step “Install Standard Dashboards and Reports” on page 34.
- 21 Each workbook uses a data source connection to the Analysis Services system. The connections used are embedded in the worksheets but must be modified to connect to the local Analysis Services host. To do this, open the workbook in Excel 2007, press Data, then Properties. On the Connection Properties tab select Definition and change the text that says `DataSource=AS_HOST` to give the name of your Analysis Services host machine.
- 22 Full documentation on this process is included in the following document from Microsoft:

<http://office.microsoft.com/en-us/excel/HP101672991033.aspx>

Fig. 2.2
Excel Workbooks



Customize Report Logo

23 The report wizard allows you to put a logo on your generated reports. The file it uses is stored in two places:

- `<tomcat_dir>\webapps\qadbi\images\logo.gif` - used for standard HTML reports.
- `<tomcat_dir>\webapps\qadbi\WEB-INF\classes\images\logo.gif` - used for PDF and Excel exports of reports.

If you want to have your company logo print on reports, replace the default `logo.gif` in the Web application's images subdirectories with a GIF file of your logo. The logo file can be different for the two classes of report types.

24 Restart Apache Tomcat through the Services window.

Note on Password Encryption in the BI Portal

You can encrypt the SQL Server password in both the `server-config.xml` file, which contains the connection information for the BI Portal database, and in the `aw_sys_config` table, which contains the connection information for the BI Data Warehouse. The encryption uses the Blowfish algorithm.

A command-line tool to encrypt the password named `encpwd.bat` is located in `TOMCAT_HOME\webapps\your_bi_appname\cmdline`. To run this, open a DOS window, change to the `cmdline` directory, and run:

```
encpwd.bat mypassword
```

The encrypted password is printed on the screen. Copy this value and use it in the `server-config.xml` file as the value for the `pwd`. Also make sure that `pwdEncrypted=true`. If you change back to using an unencrypted password value, make sure you set `pwdEncrypted=false`.

Similarly, if you want to encrypt the password used by the Portal to connect to the Data Warehouse, use the `encpwd.bat` script to generate the encrypted password. Edit the value for the `jdbcPassword` in the `aw_sys_config` table in the `DATASERVICE` row and then set it to encrypted value. Make sure you add `pwdEncrypted=true`.

If you change either password, you must stop and restart Tomcat for the change to take effect.

Customizing the BI Portal Logo

By default, the BI Portal displays the QAD logo. To change this logo, copy a similarly sized GIF image to `TOMCAT_HOME\webapps\your_bi_appname\assets\logos\_logo.gif`.

The directory `TOMCAT_HOME\webapps\your_bi_appname\assets\logos\` contains a Precision logo for use by TMS customers. To display this instead of the QAD logo, copy `TOMCAT_HOME\webapps\your_bi_appname\assets\logos\tms_precision.gif` to `TOMCAT_HOME\webapps\your_bi_appname\assets\logos\_logo.gif`.

Getting Started in the BI Portal Web Application

The initial login is root, with a password of root. You are required to change the password immediately.

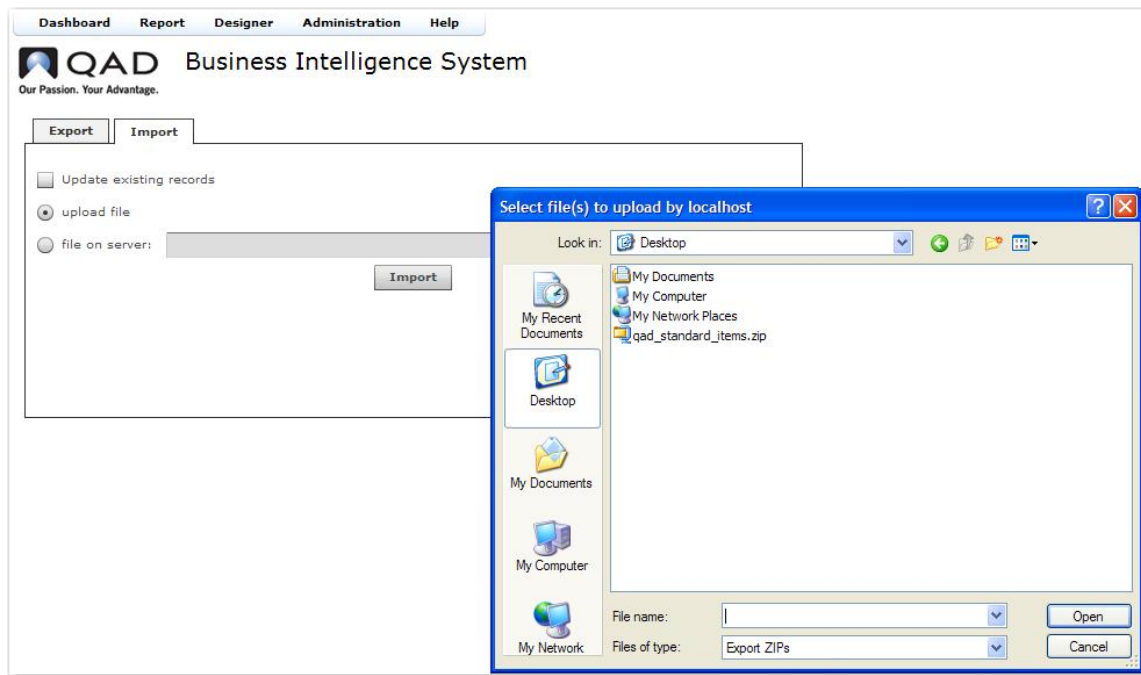
You will need to build your model before you can do any analysis. Select Administration|Model Administration and then click Rebuild Master.

Install Standard Dashboards and Reports

Standard dashboard, reports, visual items, and queries are shipped with the QAD analytical modules. The contents of the zip files `qad_<module_name>_dashboards.zip` and `qad_<module_name>_reports.zip` are portal documents used to interact with the standard modules.

Find the `qad_common_dashboards.zip` file and place it somewhere for easy access. Using the portal, go to the menu option Administration|Data Migration. Select the Import tab, select the upload file option, and then click Import. Select the `qad_common_dashboards.zip` file and click OK.

Fig. 2.3
QAD Standard Items Zip



The items upload into the system. On completion, a message indicating that the items were loaded is displayed.

Repeat the import process for your module-specific `qad_<module_name>_dashboards.zip` and `qad_<module_name>reports.zip` files.

After importing the standard objects, you must log out and log in again to see them.

Installing QAD BI Mobile App on the iPad

Requirements

- Apple iPad
- QAD BI Portal installed and configured, and Tomcat running
- User IDs created and passwords set up using the BI portal for all users who will install and use the app on their iPads

Installation and Configuration

Follow these steps to download the QAD Mobile BI app:

- 1 On your iPad, go to the Apple App Store.
- 2 In the search window, search for QAD Mobile.
- 3 Choose and install the QAD Mobile BI app.
- 4 Connect to your office network via VPN.
- 5 Double-click on the QAD BI icon. When you launch the BI Mobile App for the first time, it prompts you for the following information:
 - **User.** Enter the user id that you use to access BI portal.
 - **Password.** Enter the password that you use to access BI portal.
 - **Server URL.** The same URL that you use to access BI portal in the form of `http://<biportalhostname>:<tomcatportnumber>/<webappname>/`. For example:
`http://company:8080/qadbi/`.

If you would like to save these credentials for future use, tap on the Save Password On/Off button and turn it on.
- 6 Click on the check mark. You should now be able to log in to BI from your iPad.

QAD BI Portal CFO Dashboard

This chapter provides information about configuring the QAD BI Portal CFO Dashboard application.

It discusses the following topics:

QAD Standard CFO Dashboard 38

Describes how to configure the CFO Dashboard.

QAD Standard CFO Dashboard

The QAD CFO Dashboard is one of the standard elements that ships with the QAD Business Intelligence Portal. The CFO Dashboard is designed to collect in one easy-to-view place the important financial metrics for your organization so that the financial health of the organization can be quickly understood.

Since GL account structures vary widely from company to company, the data elements of the CFO dashboard require initial configuration. The Dashboard's visual elements are already set up in the Portal, and are ready to display graphs and charts once the data configuration is complete.

Configuring the CFO Dashboard

Metrics on the CFO dashboard are generated from report lines defined in the QAD BI database. Report lines are like the lines of a balance sheet or an income statement, representing a range, or ranges, of GL accounts. Once defined, report lines can be rolled up into groups and computations can be performed on the groups to generate the various metrics and data that appear on the CFO dashboard.

The CFO Dashboard report lines are defined and maintained with the help of an Excel workbook, which creates a set of text files that are used in the nightly QAD BI data load. Using Excel allows finance managers and analysts to maintain the financial metrics themselves and gives them the most freedom in creating and defining new metrics and financial reports.

The CFO Dashboard workbook, `cfo_dashboard_template.xls`, is located in the metadata installation package. This workbook consists of four tabs, all of which are used to build the configuration data for the CFO dashboard and its charts:

- **Reports.** This tab contains the very highest level of information to be displayed. This page lists the name of the report sets that the CFO dashboard describes. One set is given by default: CFO_METRICS. Multiple reports, or versions of reports, can be included in the same spreadsheet.
- **Report Lines.** This tab defines the lines of each report. Report lines can be labels (which just display text), data lines, which are rollups of GL accounts, or calculations, which are rollups of data lines. The CFO_METRICS report in the default configuration is broken down on the Report Lines page into 16 different lines, such as Inventory and Net Revenue.
- **Report Line Ranges.** This tab maps the Report Lines to specific accounts in the General Ledger.
Important metrics for each report may not simply be summaries of accounts but more complicated combinations of the report lines.
- **Report Calculations.** Allows user to combine the lines of the report arbitrarily to create derived metrics.

These tabs are more fully described below. Before configuring the CFO dashboard, however, you should start with a specific goal in mind to properly set up the configuration elements.

Metric Example

Consider the example of an organization that wants to see a metric like debtor days, which is the average number of days that a payment takes. The debtor days metric can be computed on any account by taking the total amount in accounts receivable, divided by net revenue received, and then multiplied by the number of days in the fiscal year. Furthermore, this should be done by region so you can see the measures for the North American and Pacific Rim regions.

To do this, first set up the accounts that you want to total. Specify the AR accounts in both North America and the Pacific Rim, and then set up the revenue received in those accounts.

After these account lines are set up, you can create the computation to display on the final dashboard.

Reports Tab

For this example, the reports tab will not have to change. This report can be collected in the set for the standard CFO_METRICS.

If the system calls for a lot of reports that need to be divided at the highest level, you can add another element to the reports tab for that highest level grouping. That new element is then carried over to the Report Lines tab.

ReportLines Tab

This tab holds the high level groupings on which the computations in the reports are performed. In this specific example, four lines must be defined: Accounts Receivable for both North America and the Pacific Rim, and revenue received for both regions.

These accounts are created by setting the report_name column to the name given on the first tab. In this example, this is CFO_METRICS.

The line_number column must be unique for each entry on this page, and it describes to the subsequent tabs how to reference this account line.

Line_type can have the value label, calc or data. A label type is the text placed into the visual item on the CFO dashboard for this line. A row can be marked either calc for a computed column as described in the upcoming Calculations tab, or data to indicate that this line's data is used in a computation.

Line_text is a description of the account. Indent_level should be set to 0 for a label, and 1 otherwise. Items that should be visible to the end user should be marked as Y in the visible column.

All computations on a report line are done on the value of the report. Since the report computations described below are additive, and sometimes you must subtract values of one line from another, you can specify that a line should be a negative value in the report lines tab by placing a Y in the change_sign column.

In this specific example, the line of net revenues must be subtracted from the total accounts receivable for a region. When you specify the received revenue in the report lines tab, change its sign to a negative by indicating a Y in the change_sign column for both the North American and Pacific Rim Net Received lines.

ReportLineRanges Tab

The report line ranges tab allows the user to group the accounts required for each report line.

Accounts in the GL are listed on the report line ranges tab sequentially to generate the higher level reporting accounts. The report line name (CFO_METRICS) must be given in the first column. The line_number column corresponds to the report line in the second tab. Range_type can take the value of include or exclude.

The remaining columns specify the characteristics range of accounts to include in the summarized report line. The following is a description of each column:

- **Source_system_code.** Name of the source system in the QAD BI database. Most implementations have a single source system.
- **From_entity / to_entity.** Entity codes to be processed, starting at from value and going until to value.
- **From_account / to_account.** Account codes to process.
- **From_subaccount / to_subaccount.** Sub accounts to process.
- **From_department / to_department.** Departments to process.
- **From_project / to_project.** Project indicators to process.

In this specific example, the accounts for the four elements of the computation are defined. Report Line 2, North American AR (East) is defined as 4000 to 4100, excluding accounts 4003 to 4005. Report Line 3, North American AR (West) is defined as 5010 to 5020. The source system for both is QAD_US. Line 4, Pacific Rim AR, is defined as accounts 1600 to 1700 for source system for both is QAD_PACRIM. The example tab continues to define the account ranges and source systems for all five data lines of the CFO_METRIC report.

ReportLineCalculations Tab

This tab defines the computations to be performed on the report lines as defined on the Report Lines tab. By combining the report lines, the higher order metrics desired can be determined.

Each line in the ReportLineCalculations tab specifies part of a computation. The data into which the computation is placed is specified in the line_number of type calc specified on the ReportLines tab. The computation steps are processed one at a time, in an order specified by each line's calc_seq. For each specific report_name and line_number, the calc_seq must be unique.

```
A step
Report_name - CFO_METRIC
Line_number - refers to the ReportLines tab, line_number. Must be a calc type.
Calc_seq - order in which computations should be performed
Operator - add or multiply
Source_line - refers to a data line in ReportLines
```

Providing the Workbook Data to the Data Warehouse

Once the data has been entered into the workbook as described, it must be translated into a format that the data warehouse processes can operate on to build the required elements for the CFO dashboard.

Each sheet in the workbook contains a yellow section of cells that is built automatically as the data is filled out. This section of cells must be placed into a text file and that text file moved to a location specified in the data warehouse.

Find the location in which the text files must be placed by double-clicking on the load tables in QAD BI DWD and noting the directory location in the File Path field. By default, this is set to `c:\work\QADReport\`. This can be changed to match the locations required by any site.

Fig. 3.1
File Path

The screenshot shows a dialog box titled "load_gen_led_report_line table definition" with a close button (X) in the top right corner. It has four tabs: "Properties", "File Attributes", "Storage", and "Notes". The "File Attributes" tab is selected. The fields and their values are as follows:

- File Path:** `C:\work\QADReport\`
- File Name:** `ReportLines.txt`
- File Column Delimiter:** `|`. A tooltip explains: "CHAR(nn) inserts an ascii character (e.g. CHAR(9) = tab). No column delimiter indicates a width based record."
- File has a header line:** ☒
- File Record Delimiter:** (empty). A tooltip explains: "FIX nnn indicates a fixed length record of length nnn. No record delimiter indicates a line feed delimited record."
- BULK INSERT options:** (empty). A tooltip explains: "See online help for further details. (e.g. TABLOCK, MAXERRORS=0)"

The files to be used, by default, are

- Reports tab - `Reports.txt`
- Report Lines tab - `ReportLines.txt`
- Report Line Ranges tab - `ReportLineRanges.txt`
- Report Line Calculations tab - `ReportLineCalculations.txt`

Once the files are in place and the QAD Warehouse daily jobs run, the new computations are reflected into the data warehouse and the CFO dashboard contents are displayed. The daily jobs are usually run overnight.

Reference

This chapter provides reference information:

Jobs ***44***

Reference Tables ***47***

Jobs

Jobs can be roughly categorized into the following types, in the order in which the types run in Chained job processing. The descriptions include the naming conventions associated with each type.

Truncate

The purpose of a Truncate job is to truncate—delete all data from—load tables prior to the beginning of a new load. This ensures that load jobs are starting with a “clean slate.” There is one such job per analytical module, named `PREPROCESS_[MODULE]_TRUNCATE`.

Load

The purpose of a Load job is to extract data from the ERP sources and load the data into the Data Warehouse for processing. The Load job for an analytical module is named `PREPROCESS_[MODULE]_LOAD`. However, if there are specific tables that are relevant to history loads only, or to daily loads only, those are loaded in a job named `PREPROCESS_[MODULE]_HISTONLY_LOADS` or `PREPROCESS_[MODULE]_DAILYONLY_LOADS`, respectively. Also, there might be further distinction when a module requires separate job streams; for example, there are some tables processed for EE sources only, and some for SE sources only. In this case the EE or SE distinction is included in the job name; for example, `PREPROCESS_FIN_EE_LOADS` is a load job for the Financials analytical module that runs for EE sources only. COMMON module load tables with SE- or EE-specific tables will include `SEONLY` or `EEONLY` in their job names; for example, `PREPROCESS_COMMON_EEONLY_LOADS` or `PREPROCESS_COMMON_SEONLY_LOADS`.

Perm

These jobs create the “perm” tables, which are permanent data stores, where the BI system stores data extracted from the ERP sources that needs to be maintained for long periods of time. There are two such jobs for each module:

- **History loads.** The History job expects a large amount of data to be processed at once as the Data Warehouse is loaded for the first time.
- **Daily loads.** Daily loads are usually processing a smaller amount of data—data that has changed in the sources since the last data load.

These jobs are named `[HIST_or_DAILY]_[MODULE]_PERM_CHAINED`. For Financials, there may be an `_EE` or `_SE` after the `[MODULE]` if the job applies to one type of ERP source only.

Extract

This set of jobs is to build extract and work tables from the perm tables. Not all of the data in the perm tables is typically extracted at one time, as much of the data in the perm tables has been already processed and has not changed. So usually a subset of data, covering a certain period and/or specifically targeting changed data, is extracted from the permanent data stores for further processing into the Fact tables. There are two such jobs for each module—one for History loads and one for Daily loads. These jobs are named `[HIST_or_DAILY]_[MODULE]_PERM_EXTRACT`. For Financials, there may be an `_EE` or `_SE`

after the [MODULE] if the job applies to one type of ERP source only. Note that Extract jobs do not end in CHAINED; they are not an explicit part of the job chain, but rather are always called from a Perm job.

Process

This set of jobs is to take extracted and loaded data through various stages to build the dimensions and facts. There are no summaries, aggregates, snapshots, or cubes. These jobs are named [HIST_or_DAILY]_[MODULE]_PROCESS_CHAINED. For Financials, there may be an _EE or _SE after the [MODULE] if the job applies to one type of ERP source only.

Additionally, some modules may have a main job containing the module name, and then additional jobs for processing related to a specific submodule. These jobs are named [HIST_or_DAILY]_[MODULE]_[SUB_MODULE_ABBREVIATION]_PROCESS_CHAINED and run after the primary job for the module. For example, the Financials module has its primary Daily job, DAILY_FIN_PROCESS_CHAINED, plus separate jobs for processing Accounts Payable (AP), Accounts Receivable (AR), and General Ledger (GL). The submodule jobs are DAILY_FIN_[EE_or_SE]_AP_PROCESS_CHAINED, DAILY_FIN_[EE_or_SE]_AR_PROCESS_CHAINED, and DAILY_FIN_[EE_or_SE]_GL_PROCESS_CHAINED.

Rollup (aggregate/rollup/snapshot)

This set of jobs is to create aggregate/rollup/snapshot tables, which mainly rely on the relevant facts/dimensions to be populated already. There are two such jobs for each submodule—one for History and one for Daily loads. They are typically named POSTPROCESS_[MODULE]_SUMMARY. However, if the specific post-processing is to history only, or to daily only, the job name is POSTPROCESS_[MODULE]_HISTONLY_SUMMARY or POSTPROCESS_[MODULE]_DAILYONLY_SUMMARY, respectively. Additionally, if the post-processing is specific to a submodule of a module, the submodule's abbreviation will appear in the name; for example, POSTPROCESS_OP_PO_DAILYONLY_SUMMARY.

Cubes

Cubes jobs process analytical cubes so that they include the most recent data from the fact and dimension tables. There is one job per module. The naming convention is POSTPROCESS_[MODULE]_CUBES. For Financials, there might be an SE or EE after the MODULE name if the cubes are specific to one source type; for example, POSTPROCESS_FIN_SE_CUBES. While not installed in the job chain automatically, these jobs can be added to the Daily job chain if desired.

Notes on ERP Transaction Load History

TR_HIST_LOAD_MAXSIZE is the maximum number of rows to load in an iteration loading tr_hist.

The Transaction History load (HIST_LOAD_TR_HIST) processes data in chunks, rather than trying to extract and load all data at once, since the tables can contain large amounts of data. The History Load processing runs this job many times, processing some subset of the data with each pass. The parameters that are used by the TR_HIST jobs are listed below.

Important Do not update them manually; the HIST_LOAD jobs use these to keep track of what data has already been processed, so that if the load job is interrupted, it can pick up where it left off.

- TR_HIST_CONNECTION_NAME
- TR_HIST_DOMAIN
- TR_HIST_DOMAIN_SXX
- TR_HIST_LOOP_NO
- TR_HIST_ROW_COUNT
- TR_HIST_SOURCE_SXX
- TR_HIST_TRAN_MAX
- TR_HIST_TRAN_MAX_SXX
- TR_HIST_TRAN_MIN

Reference Tables

The following tables include installation and configuration reference information in tabular form.

Common Module

Table 4.1

Parameters for the Common Module (Page 1 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
ALTERNATE_EXCHANGE_DOMAIN		USA	Alternate exchange rate domain name	Used to calculate exchange rates if the domain in which the calculation is occurring does not have an exchange rate defined	COM	M
AR_PAYMENT_PROCESS_DAYS	60	60	Default value should be 60 days to search for any backdated payments.		COM	M
AR_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the AR process job		COM	P
COM_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the common process job		COM	P
COM_PROCESS_DAYS	750		The number of days for Dimension tables to extract and process from load tables based on last update date.		COM	M

Table 4.1
Parameters for the Common Module (Page 2 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/Programmatic
COM_PROCESS_RUNNING	N	N	Indicates if a Common process is currently running (Y,N)	Used to prevent multiple copies of the COMMON job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from to "N" from "Y".	COM	P
COM_PROCESS_RUNNING_JOB_NAME		DAILY_PERM_2010SE	Indicates the current or last run job name for COMMON processing.		COM	P
CORP_CALENDAR_DOMAIN	<Enter domain here for corporate calendar setup>	10USA	The Domain Code used for the corporate financial calendar	Not used currently	COM	M
CORP_CALENDAR_SOURCE	<Enter data source name here for corporate calendar setup>	QMI	The Source System Code used for the corporate financial calendar	Not used currently	COM	M
CORP_CURRENCY_CODE	USD	USD	Corporate currency code used for high level reporting		COM	M
DAILY_DATE_ROLL_OVERRIDE		2010-01-01	YYYY-MM-DD Leave blank to keep dim_date_qad based on today's date; Enter Date to make DAILY job reset the current calendar date to override date.	Added in 3.7. Used to override current date in the daily_date_roll_qad (which resets the dim_date_qad dimension's flags for current day, week, month, period, etc.) LEAVE BLANK FOR NORMAL PROCESSING.	COM	M

Table 4.1

Parameters for the Common Module (Page 3 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
DAILY_HISTORY_DATE_SET_CURRENT	Y	Y	Default Y indicates jobs will always set the SALES and PO HISTORY DATE to the current date. If there is a need to run loads using a date prior to today, set the Value to N and change the values of the history dates to the date needed.	3.4.1 - Added to determine if the SALES_ORDER_HISTORY_DATES and PO_ORDER_HISTORY_DATES that are set by their respective step jobs should increment by one (better for testing purposes) or be set to current date (presuming that current date minus process days is greater than the last _HISTORY_DATE to avoid data gaps).	COM	P
DAILY_LOAD_JOB_CURNO	0	21	Current Daily Load Job Number		COM	P
DAILY_LOAD_JOB0001		SET_CONNEC TION_QMI	Inserted via WsParameterWrite		COM	P
DAILY_LOAD_JOB0002 - DAILY_LOAD_JOBxxxx	Does not exist until INITIAL_JOB_S ETUP is run; jobs and their order determined by INITIAL_JOB_S ETUP job; should be a complete set of jobs for each source, in order		Inserted via WsParameterWrite		COM	P
DAILY_LOAD_SOURCE_CURNO	0	0	Current Source System Number being run		COM	P
DAILY_LOAD_SOURCE_S01	<Enter data source name here for initial setup>	QMI	Name of the source system loaded as the 01 job.	The connection name of the first or primary source system.	COM	M

Table 4.1
Parameters for the Common Module (Page 4 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/Programmatic
DAILY_LOAD_SOURCE_S02 - DAILY_LOAD_SOURCE_Sxx	Does not exist by default; add if additional source(s)	2010SE		If needed, the connection name(s) of the second and subsequent source systems	COM	M
DAILY_LOAD_SOURCE_S99	Dummy Parameter	Dummy Parameter	Add more by just inserting more parameters	Add parameters with consecutive numbers for each additional source system.	COM	Do Not Edit
DAILY_MULTI_SOURCE_STAGING			'Y' - Run procedures for all sources; 'N' - Run procedure for source specified in %_PROCESS_CONNECTI ON_NAME	Added in 3.7; for future use	COM	M
DATE_EARLIEST_POSSIBLE	1900-01-01	1900-01-01	The latest date value possible used for unknown dates and setting history dates		COM	M
DATE_LATEST_POSSIBLE	2999-12-31	2999-12-31	The earliest date value possible used for unknown dates and setting history dates		COM	M
DEFAULT_ACCOUNT	00000	00000	The default value used for unknown account	Must not be a real account code	COM	M
DEFAULT_COST_CENTER	0000	0000	The default value used for unknown cost center	Must not be a real cost center code	COM	M
DEFAULT_DOMAIN	00000	Unknown Domain	The default value used for unknown domain	Must not be a real domain code	COM	M
DEFAULT_ENTITY	0000	0000	The default value used for unknown entity	Must not be a real entity code	COM	M
DEFAULT_PROJECT	00000	00000	The default value used for unknown project	Must not be a real project code	COM	M
DEFAULT_SOURCE		Unknown Source	The default value used for unknown source system	Must not be a real source system code	COM	M

Table 4.1

Parameters for the Common Module (Page 5 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
DEFAULT_SUB_ACCOUNT	0000000	0000000	The default value used for unknown sub-account	Must not be a real sub-account code	COM	M
FIN_FIRST_DAY_NAME	SUN	SUN	The short day name of the first day of the week. Used in financial calendar derivation.		COM	M
HIST_LOAD_JOB_CURNO	0	21	Current History Load Job Number		COM	P
HIST_LOAD_JOB_PREFIX	HIST_LOAD_	HIST_LOAD_	Base name of the History Load job		COM	P
HIST_LOAD_JOB0001		SET_CONNEC TION_QMI	Inserted via WsParameterWrite		COM	P
HIST_LOAD_JOB0002 - HIST_LOAD_JOBxxxx	Does not exist until INITIAL_JOB_S ETUP is run; jobs and their order determined by INITIAL_JOB_S ETUP job; should be a complete set of jobs for each source, in order		Inserted via WsParameterWrite		COM	P
HIST_LOAD_SOURCE_CURNO	1	1	Current Source System Number being run		COM	P
HIST_LOAD_SOURCE_S01	<Enter data source name here for initial setup>	QMI	Name of the source system loaded as the 01 job.		COM	M
HIST_LOAD_SOURCE_S02 - HIST_LOAD_SOURCE_Sxx	Does not exist by default; add if additional source	2010SE			COM	M

Table 4.1
Parameters for the Common Module (Page 6 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
HIST_LOAD_SOURCE_S99	Add more by just inserting more parameters	Add more by just inserting more parameters	Dummy Parameter		COM	Do Not Edit
HIST_LOAD_TR_HIST_RUNNING	N	C	Indicates a tr_hist history load processes is underway. C for completed, Y for yes or N for no.		COM	P
INITIAL_JOB_SETUP_CONNECTION_01	<Enter data source name here for initial setup>	QMI	Name of the connection 01 used as a source system for cloning jobs.		COM	M
INITIAL_JOB_SETUP_CONNECTION_01_RUN	N	Y	This determines if the Initial Job Setup has been run against this data source yet or not. Any new data source should have an N. Value is set to C when historic load has completed.	Used by INITIAL_JOB_SETUP job to ensure that existing data sources are NOT re-run as part of the historic load when new sources are added.	COM	M/P
INITIAL_JOB_SETUP_CONNECTION_01_TYPE	<Enter source type - SE, EE, EAM>	EE	Connection Type - SE for ERP with Standard Edition Financials, EE for ERP with Enterprise Edition Financials, and EAM for Enterprise Asset Management. Defaults to SE	Used to determine whether to set up EE Financial or SE Financial jobs for this data source, or whether to use the EAM job chain instead of the ERP job chain.	COM	M
INITIAL_JOB_SETUP_CONNECTION_01_DAILY_MODULES	COM, OM, OP, FIN, EAM	COM, FIN	Example: 'COM, FIN' - List Modules to be part of DAILY job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM	M
INITIAL_JOB_SETUP_CONNECTION_01_HIST_MODULES	COM, OM, OP, FIN, EAM	COM, FIN	Example: 'COM, FIN' - List Modules to be part of HIST job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM	M

Table 4.1

Parameters for the Common Module (Page 7 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INITIAL_JOB_SETUP_CONNECTION_02		2010SE	Name of the connection 02 used as a source system for cloning jobs.		COM	M
INITIAL_JOB_SETUP_CONNECTION_02_RUN		Y	This determines if the Initial Job Setup has been run against this data source yet or not. Any new data source should have an N.	Added in 3.4.1 – Used by INITIAL_JOB_SETUP job to ensure that existing data sources are NOT re-run as part of the historic load when new sources are added.	COM	M/P
INITIAL_JOB_SETUP_CONNECTION_02_TYPE		SE	Connection Type - SE for ERP with Standard Edition Financials, EE for ERP with Enterprise Edition Financials, and EAM for Enterprise Asset Management. Defaults to SEConnection		COM	M
INITIAL_JOB_SETUP_CONNECTION_02_DAILY_MODULES	COM, OM, OP, FIN, EAM	COM, FIN	Example: 'COM, FIN' - List Modules to be part of DAILY job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM	M
INITIAL_JOB_SETUP_CONNECTION_02_HIST_MODULES	COM, OM, OP, FIN, EAM	COM, FIN	Example: 'COM, FIN' - List Modules to be part of HIST job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM	M
INITIAL_JOB_SETUP_CONNECTION_99	Add more by just inserting more parameters	Add more by just inserting more parameters	Dummy Parameter		COM	

Table 4.1
Parameters for the Common Module (Page 8 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INITIAL_JOB_SETUP_DATE	<Set to today's date before starting the initial history load>	Set to today's date (current date) so INITIAL_JOB_SETUP uses this date to populate various other parameters. YYYY-MM-DD	Added in 3.4.1. INITIAL_JOB_SETUP uses this date to populate the majority of the previously manually populated date fields for job setup.		COM	M
INITIAL_JOB_SETUP_ENABLED	Y	N	Indicates the job cloning functionality is unlocked. This is a Y or N flag - initially Y then change to N when job cloning has been completed.	The INITIAL_JOB_SETUP job will not do anything if this is set to "N". If set to "Y", running the INITIAL_JOB_SETUP job will create the jobs for each source defined in the DAILY and HIST source parameters, then set the value back to "N" so that the initial job cannot be accidentally run again.	COM	M/P
INITIAL_JOB_SETUP_JOBINJOB_01	DAILY_XXXX XXXX	DAILY_XXXX XXXX	Name of the 01 job that contains other jobs.		COM	Do Not Edit
INITIAL_JOB_SETUP_JOBINJOB_02	HIST_LOAD_X XXXXXXX	HIST_LOAD_ XXXXXXXXX	Name of the 02 job that contains other jobs.		COM	Do Not Edit
INITIAL_JOB_SETUP_JOBINJOB_03			Name of the 03 job that contains other jobs.		COM	Do Not Edit
INITIAL_JOB_SETUP_JOBINJOB_04			Name of the 04 job that contains other jobs.		COM	Do Not Edit
INITIAL_JOB_SETUP_JOBINJOB_99	Add more by just inserting more parameters	Add more by just inserting more parameters	Dummy Parameter		COM	Do Not Edit
JOB_CHAINING_ENABLED	Y	Y	Y if a job starts the next job in the list automatically as it ends, N otherwise		COM	M

Table 4.1

Parameters for the Common Module (Page 9 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
JOB_COM_PROCESS_BATCH_ID	0	748	The batch id for the current DAILY_COMMON_PROCESS job.		COM	P
load_tr_hist_tr_qty_cn_adj	0	tr_hist.tr_qty_cn_adj	Internal use only. Specifies the column name in tr_hist for consignment adjustments, or 0 if not supported on this version of the source system.		COM	P
NOT_APPLICABLE_FLAG_VALUE	N	N	The value assigned for a flag field that is not applicable.		COM	M
NOT_APPLICABLE_NUM_VALUE	0	0	The value assigned for a numeric field that is not applicable.		COM	M
NOT_APPLICABLE_TEXT_VALUE	Not Applicable	Not Applicable	The value assigned for a field that is not applicable.		COM	M
PERM_HIST_DATE	1900-01-01	1900-01-01	History Date for Permanent Staging Tables		COM	P
PERM_HIST_PREFIX	HIST	HIST	History load job name prefix		COM	Do Not Edit
QAD_MASTER_CONNECTION_NAME	QAD_MASTER	QAD_MASTE R	The name of the master generic connection used for all load tables		COM	Do Not Edit
TR_HIST_CONNECTION_NAME		2010SE	Inserted via WsParameterWrite		COM	P
TR_HIST_DATE_MAX		2012-08-08	DON'T UPDATE MANUALLY. Maximum tr_hist Date (YYYY-MM-DD) from last run for this source.	Copied from TR_HIST_DATE_MAX_SXX for the source currently being processed.	COM	P

Table 4.1
Parameters for the Common Module (Page 10 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TR_HIST_DATE_MAX_SXX		2012-08-08	DON'T UPDATE MANUALLY. Maximum tr_hist Date (YYYY-MM-DD) from last run for this source.	Added in 3.6.1. Used with TR_HIST_PROCESS_DAYS and tr_trnbr to ensure that most recent transactions are retrieved, even after tr_trnbr rollover.	COM	P
TR_HIST_DOMAIN		'AUTO','QP','QPAU','QPCAN','QPMEX','train'	DON'T UPDATE MANUALLY. Domain to extract this iteration		COM	P
TR_HIST_DOMAIN_SXX	Does not exist until after a load of source_XX's data	'10USA','11CAN','12MEX','20FRA','21NL','22UK','30CHN','31AUS','CAN','FRA','QMSEMEA','QMSUS','Train','UK','USA'	Inserted via WsParameterWrite	List of domains in the source; set programatically.	COM	P
TR_HIST_LOAD_MAXSIZE	1000000	1000000	The maximum number of rows to load in an iteration loading tr_hist		COM	P
TR_HIST_LOOP_NO	0	0	DON'T UPDATE MANUALLY. The iteration number loading tr_hist.		COM	P
TR_HIST_MAX_VALUE	99999999	99999999	The rollover number for the tr_hist.tr_trnbr field.	Added in 3.6.1. This is the maximum number allowed for the tr_hist.tr_trnbr field, or in other words, the number after which the tr_trnbr will roll over and start at the first available low number. The default value is 99999999. If the tr_sq01 sequence is set to a different maximum value, enter that here.	COM	M

Table 4.1

Parameters for the Common Module (Page 11 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TR_HIST_PCT_OF_MAX_REF	50	30	Percent of the TR_HIST_MAX_VALUE to use as a reference point to look at smaller or bigger numbers when dealing with the rollover.	Added in 3.6.1. This is the percent of the TR_HIST_MAX_VALUE to use as a reference point to look for new low-numbered transactions when dealing with the rollover. The default value is 50. When this value is 50, you will search the transactions with numbers less than 50% of the TR_HIST_MAX_VALUE for the highest number with a transaction date within the processing window.	COM	M
TR_HIST_PRE_ROLLOVER_COUNT	5000	7000	Number of records that would typically be in three days worth of data to ensure readiness for tr_hist.tr_trnbr rollover.	Added in 3.6.1. This is the number of records that would typically be in three days worth of data to ensure readiness for tr_hist.tr_trnbr rollover. Once the tr_trnbr for a source is detected to be within TR_HIST_PRE_ROLLOVER_COUNT of TR_HIST_MAX_VALUE, the BI system sets the TR_HIST_TRAN_MAX_Sxx value for the source to 0. This will force the tr_hist extraction to scan the whole table for records that have been processed within the last TR_HIST_PROCESS_DAYS. The default value is 5000.	COM	M

Table 4.1
Parameters for the Common Module (Page 12 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TR_HIST_PROCESS_DAYS	3	6	Number of look back days when extracting data from tr_hist. The records being extracted are tr_trnbr records that are > than the highest transaction_number records already extracted for that source except for tr_nbr rollover where dates come into play.	Added in 3.6.1. This is the number of look-back days when extracting data from tr_hist. The records being extracted are records where tr_trnbr is greater than the highest transaction_number records already extracted for that source, except after tr_trnbr rollover, when newer transactions may have smaller trnbrs than older transactions. The BI DW must also take into account the transaction date in order to find the lower-numbered transactions that are actually new. This parameter says to extract from the ERP transactions entered within the last X days. If the number of days since the TR_HIST_DATE_MAX_Sxx is greater than the TR_HIST_PROCESS_DAYS parameter, a task that checks this during the run will fail and indicate this number needs to be increased. The default value is 3.	COM	M
TR_HIST_ROW_COUNT	0	24186	DON'T UPDATE MANUALLY. Number of rows loaded into tr_hist.		COM	P
TR_HIST_SOURCE_SXX	Does not exist until after a load of source's data	QMI	Inserted via WsParameterWrite		COM	P
TR_HIST_TRAN_MAX		2134	DON'T UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		COM	P

Table 4.1

Parameters for the Common Module (Page 13 of 13)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TR_HIST_TRAN_MAX_SXX	Does not exist until after a load of source's data	27971	DON'T UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		COM	P
TR_HIST_TRAN_MIN	-1	-1	DON'T UPDATE MANUALLY. Minimum Transaction Id to extract this iteration.		COM	P
TR_QTY_CN_ADJ	tr_hist.tr_qty_cn_adj	0	Internal use only. Specifies the column name in tr_hist for consignment adjustments, or 0 if not supported on this version of the source system.	Is set internally; do not set manually. Used by load_tr_hist and load_tr_hist_history	COM	P
UNKNOWN_NUM_VALUE	9999	9999	Default value to insert for a column containing NULLs. Used for Primary Key Columns.		COM	M
UNKNOWN_VALUE	Unknown	Unknown	Default value to insert for a column containing NULLs. Used for Primary Key Columns.		COM	M

Table 4.2

Parameters for Various Module

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
load_XXXXXX_0	Does not exist until after a load	1301	Rowcount from native load at 31/01/11 04:36:45	One row per load table; value is number of rows loaded the last time the specified table was loaded; comment indicates date/time of last load	Various	P

Financials Module

Table 4.3
Parameters for the Financials Module (Page 1 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
AP_CHECK_PROCESS_DAYS	60	60	Used to find any checks to be processed which covers up to 60 days backdated.		FIN	M
AP_EE_HISTORY_DATE		20081231	The date the AP aging table is to be refreshed from. Three years back is common. For example, 20081231	Set from AP_EE_HISTORY_DATE_MIN when initial job setup is run.	FIN	P
AP_EE_HISTORY_DATE_MIN	<customer defined - usually three years back. YYYYMMDD>	20081231	The date the AP aging table is to be refreshed from. Three years back is common. For example, 20081231	Added this in 3.4.1 to handle historic loads from multiple EE sources.	FIN	M
AP_EE_HISTORY_DATE_Sxx		20081231	Inserted via WsParameterWrite	Created by procedures. - 3.4.1, using the AP_EE_HISTORY_DATE. Sxx references which EE source number this date is for.	FIN	P
AP_EE_SNAPSHOT_FREQUENCY	M	M	The frequency of snapshots in the AP aging snapshot tables. Possible values are: D for Daily, W for Weekly, M for Monthly, B for both weekly and monthly.	Note that Daily will result in 7 times the amount of data as Weekly; Weekly is approximately 4 times the amount of data as Monthly. Both Weekly and Monthly means that you will get a record for each end-of-week date as well as a record for each end-of-month date. Need to determine frequency based upon desired view level (i.e., do you want to be able to compare things on a monthly, weekly, or daily basis), number of days of data to view (i.e., how far back should the data go in the fact table), and amount of physical disk space available	FIN	M

Table 4.3

Parameters for the Financials Module (Page 2 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
AP_EE_SNAPSHOT_HIST_START_DATE	2008-12-31	2008-12-31	The first date in the AP aging snapshot table in the format YYYY-MM-DD for the first run	Date used for start of AP snapshot in History load. Used for History load only; Daily loads use AP_EE_SNAPSHOT_NUM_DAYS	FIN	M
AP_EE_SNAPSHOT_NUM_DAYS	31	31	The number of days of snapshots to load back each run.	Number of days to include in snapshot. Used to determine AP_EE_SNAPSHOT_START_DATE, which is calculated as TODAY - this value.	FIN	M
AP_EE_SNAPSHOT_REPLACE_FLAG	N	N	Indicates snapshots are replaced or not. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to just add new records and not update previously calculated records. Default value is N.	FIN	M
AP_EE_SNAPSHOT_START_DATE	1996-11-12	1997-05-24	The first date in the AP aging snapshot table in the format YYYY-MM-DD	Calculated as TODAY minus AP_EE_SNAPSHOT_START_DATE	FIN	P
AP_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the AP process job	Connection name used by the currently running AP process, or if no AP job is running, the last-used source connection	FIN	P
AP_PROCESS_DAYS	10	10	The number of days of AP to extract and process from permanent staging tables.		FIN	M
AP_PROCESS_RUNNING	N	N	Indicates if an AP process is currently running (Y,N)	Used to prevent multiple copies of the AP job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from to "N" from "Y".	FIN	P

Table 4.3
Parameters for the Financials Module (Page 3 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
AP_PROCESS_RUNNING_JOB_NAME		DAILY_EE_A P_CHAINED	Indicates the current or last run job name for AP processing.		FIN	P
AP_REVALUE_PERIODS	30	30	The number of periods to revalue in invoice history		FIN	M
AP_SE_SNAPSHOT_FREQUENCY	M	M	The frequency of snapshots in the AP aging snapshot tables. Possible values are: D for Daily, W for Weekly, M for Monthly, B for both weekly and monthly.		FIN	M
AP_SE_SNAPSHOT_HIST_START_DATE	2009-01-01	2009-01-01	The first date in the AP aging snapshot table in the format YYYY-MM-DD for the first run	Date used for start of AP snapshot in History load. Used for History load only; Daily loads use AP_SE_SNAPSHOT_NUM_DAYS	FIN	M
AP_SE_SNAPSHOT_NUM_DAYS	31	31	The number of days of snapshots to load back each run.	Number of days to include in snapshot. Used to determine AP_SE_SNAPSHOT_START_DATE, which is calculated as TODAY - this value.	FIN	M
AP_SE_SNAPSHOT_REPLACE_FLAG	N	N	Indicates snapshots are replaced or not. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to just add new records and not update previously calculated records. Default value is N.	FIN	M
AP_SE_SNAPSHOT_START_DATE	2009-01-01	2009-01-01	The first date in the AP aging snapshot table in the format YYYY-MM-DD to be loaded this run.	Calculated as TODAY minus AP_SE_SNAPSHOT_START_DATE	FIN	P
AR_EE_HISTORY_DATE		20081231	The date the AR aging table is to be refreshed from. Three years back is common. For example, 20081231		FIN	P

Table 4.3

Parameters for the Financials Module (Page 4 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
AR_EE_HISTORY_DATE_MIN	<customer defined - usually three years back. YYYYMMDD>	20081231	The date the AP aging table is to be refreshed from. Three years back is common. For example, 20081231	Added this in 3.4.1 to handle historic loads from multiple EE sources.	FIN	M
AR_EE_HISTORY_DATE_Sxx		20081231	Inserted via WsParameterWrite	Created by procedures. - 3.4.1, using the AP_EE_HISTORY_DATE. Sxx references which EE source number this date is for.	FIN	P
AR_EE_SNAPSHOT_FREQUENCY	M	M	The frequency of snapshots in the AR aging snapshot tables. Possible values are: D for Daily, W for Weekly, M for Monthly, B for both weekly and monthly.		FIN	M
AR_EE_SNAPSHOT_HIST_START_DATE	2008-12-31	2008-12-31	The first date in the AR aging snapshot table in the format YYYY-MM-DD for the first run	Date used for start of AR snapshot in History load. Used for History load only; Daily loads use AR_EE_SNAPSHOT_NUM_DAYS	FIN	M
AR_EE_SNAPSHOT_NUM_DAYS	31	31	The number of days of snapshots to load back each run.	Number of days to include in snapshot. Used to determine AR_EE_SNAPSHOT_START_DATE, which is calculated as TODAY - this value.	FIN	M
AR_EE_SNAPSHOT_REPLACE_FLAG	N	N	Indicates snapshots are replaced or not. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to just add new records and not update previously calculated records. Default value is N.	FIN	M
AR_EE_SNAPSHOT_START_DATE	1996-11-12	1997-05-24	The first date in the AR aging snapshot table in the format YYYY-MM-DD	Calculated as TODAY minus AR_EE_SNAPSHOT_START_DATE	FIN	P

Table 4.3
Parameters for the Financials Module (Page 5 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
AR_PROCESS_DAYS	10	10	The number of days of AR to extract and process from permanent staging tables.		FIN	M
AR_PROCESS_RUNNING	N	N	Indicates if a AR process is currently running (Y,N)	Used to prevent multiple copies of the AR job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from "N" from "Y".	FIN	P
AR_PROCESS_RUNNING_JOB_NAME		DAILY_EE_AR_CHAINED	Indicates the current or last run job name for AR processing.		FIN	P
AR_REVALUE_PERIODS	30	30	The number of periods to revalue in invoice history		FIN	M
AR_SE_SNAPSHOT_FREQUENCY	M	M	The frequency of snapshots in the AR aging snapshot tables. Possible values are: D for Daily, W for Weekly, M for Monthly, B for both weekly and monthly.		FIN	M
AR_SE_SNAPSHOT_HIST_START_DATE	2009-01-01	2009-01-01	The first date in the AR aging snapshot table in the format YYYY-MM-DD for the first run	Date used for start of AR snapshot in History load. Used for History load only; Daily loads use AR_SE_SNAPSHOT_NUM_DAYS	FIN	M
AR_SE_SNAPSHOT_NUM_DAYS	31	31	The number of days of snapshots to load back each run.	Number of days to include in snapshot. Used to determine AS_EE_SNAPSHOT_START_DATE, which is calculated as TODAY - this value.	FIN	M

Table 4.3

Parameters for the Financials Module (Page 6 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
AR_SE_SNAPSHOT_REPLACE_FLAG	N	N	Indicates snapshots are replaced or not. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to just add new records and not update previously calculated records. Default value is N.	FIN	M
AR_SE_SNAPSHOT_START_DATE	2009-01-01	1997-05-24	The first date in the AR aging snapshot table in the format YYYY-MM-DD to be loaded this run.	Calculated as TODAY minus AR_SE_SNAPSHOT_START_DATE	FIN	P
BUDGET_ACTUAL_ROW	ACTUAL	ACTUAL	The source data for the actual row in the budget dimension		FIN	M
GL_EE_POST_MAX	0	0	0		FIN	P
GL_EE_POST_MAX_Sxx	Does not exist until first EE GL load	441563	DON'T UPDATE MANUALLY. Maximum Posting Id to extract this iteration.		FIN	P
GL_EE_POSTLINE_MAX	0	0	0		FIN	P
GL_EE_POSTLINE_MAX_Sxx	Does not exist until first EE GL load	441569	DON'T UPDATE MANUALLY. Maximum Posting Line Id to extract this iteration.		FIN	P
GL_EE_SOURCE_Sxx	Does not exist until first EE GL load	QMI	Inserted via WsParameterWrite		FIN	P
GL_HISTORY_LOAD_REQUIRED	Y	N	Controls initial processing of loading of historical GL data. Always 'N' unless an initial history load is required		FIN	P

Table 4.3
Parameters for the Financials Module (Page 7 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
GL_LOAD_DAYS	152	52	The number of days to load from the look back days forward for GL related transactional tables	NOTE: Set to a high value for History loads to load all data desired for historical purposes. Then set to a lower value for Daily loads. Should always be larger than GL_LOAD_LOOKBACK_DAYS.	FIN	M
GL_LOAD_LOOKBACK_DAYS	150	50	The number of days to look back for changes in QAD in a table for GL related transactional tables	NOTE: Set to a high value for History loads to load all data desired for historical purposes. Then set to a lower value for Daily loads. Should always be smaller than GL_LOAD_DAYS.	FIN	M
GL_LOAD_PERIOD_FROM	199301	199705	The starting period for changes in QAD in a table for GL related transactional tables. Calculated using parameter GL_LOAD_LOOKBACK_DAYS.		FIN	P
GL_LOAD_PERIOD_TO	201002	201012	The finishing period for changes in QAD in a table for GL related transactional tables. Calculated using parameters GL_LOAD_LOOKBACK_DAYS and GL_LOAD_DAYS.		FIN	P
GL_PERIOD1_FISCAL_ACCOUNT_ZERO_TYPES		I,E	Account types to set to zero in fact_gl_balances when a new fiscal period starts. If null, none are set. The proper values in this parameter should be a list of account codes such that the string would be appropriate in an "IN" clause.		FIN	M

Table 4.3

Parameters for the Financials Module (Page 8 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
GL_POST_LOOP_NO	0	0			FIN	P
GL_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the GL process job		FIN	P
GL_PROCESS_DAYS	10	10	The number of days of GL to extract and process from permanent staging tables.		FIN	M
GL_PROCESS_RUNNING	N	N	Indicates if a GL process is currently running (Y,N)	Used to prevent multiple copies of the GL job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from "N" from "Y".	FIN	P
GL_PROCESS_RUNNING_JOB_NAME		DAILY_EE_GL_CHAINED	Indicates the GL process that is currently running or the GL process that last ran.		FIN	P
GL_YEAR_END_ACCOUNT	121100	121100	The default account for the year end carry over amount	Not used currently	FIN	M
GLTR_HIST_CONNECTION_NAME		2010SE	Inserted via WsParameterWrite		FIN	P
GLTR_HIST_DOMAIN		'QAD','QP','QPAU','train'	Inserted via WsParameterWrite		FIN	P
GLTR_HIST_LOAD_MAXSIZE	500000	500000	The maximum number of rows to load in an iteration loading gltr_hist		FIN	P
GLTR_HIST_LOOP_NO	0	0	DON'T UPDATE MANUALLY. The iteration number loading tr_hist.		FIN	P

Table 4.3
Parameters for the Financials Module (Page 9 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
GLTR_HIST_ROW_COUNT	0	4919	DON'T UPDATE MANUALLY. Number of rows loaded into gltr_hist.		FIN	P
GLTR_HIST_TRAN_MAX	0	2007041100041 48000.00	DON'T UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		FIN	P
GLTR_HIST_TRAN_MIN	-1	-1	DON'T UPDATE MANUALLY. Minimum Transaction Id to extract this iteration.		FIN	P
HIST_LOAD_EE_AP_RUNNING	N	C	Indicates the EXTRA_DAY_EE_AP_HI STORY load processes is underway. C for completed, Y for yes or N for no.	Added in 3.4.1 – Used by the HIST_EE_AP_AR_FINISH_CHECK job to determine if it can let the next set of jobs run.	FIN	P
HIST_LOAD_EE_AR_RUNNING	N	C	Indicates the EXTRA_DAY_EE_AR_HI STORY load processes is underway. C for completed, Y for yes or N for no.	Added in 3.4.1 – Used by the HIST_EE_AP_AR_FINISH_CHECK job to determine if it can let the next set of jobs run.	FIN	P
HIST_LOAD_GLTR_HIST_RUNNING	N	C	Indicates a gltr_hist history load processes is underway. C for completed, Y for yes or N for no.		FIN	P
JOB_AP_PROCESS_BATCH_ID	0	749	The batch id for the current DAILY_AP_PROCESS job.		FIN	P

Table 4.3

Parameters for the Financials Module (Page 10 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
JOB_AR_PROCESS_BATCH_ID	0	750	The batch id for the current DAILY_AR_PROCESS job.		FIN	P
JOB_GL_PROCESS_BATCH_ID	0	751	The batch id for the current DAILY_GL_PROCESS job.		FIN	P

Order Management Module

Table 4.4

Parameters for the Order Management Module (Page 1 of 5)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
JOB_SALES_PROCESS_BATCH_ID	0	753	The batch id for the current DAILY_SALES_PROCESS job.		OM	P
load_ih_hist_ih_invoicetype	0	ih_hist.ih_invoicetype	Internal use only. Specifies the column name in ih_hist for invoice_type, or 0 if not supported on this version of the source system. There are two invoice type columns in the EE version of ih_hist. This allows us to capture both.		OM	P
load_sod_det_sod_compl_stat	N/A'	sod_det.sod_compl_stat	Internal use only. Specifies the column name in sod_det for complete state or N/A if not supported on this version of the source system. (EE vs SE)		OM	P

Table 4.4
Parameters for the Order Management Module (Page 2 of 5)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
SALES_BOOKING_DATE	<Set to today's date before starting the initial history load>	2011-01-31	The processing date used for SALES bookings processing. Format is YYYY-MM-DD		OM	P
SALES_BOOKING_FIRST_DATE	<Set to today's date before starting the initial history load>	2011-01-28	The first processing date for loading bookings. Format is YYYY-MM-DD		OM	P
SALES_EARLY_TOLERANCE_DELAY	0	2	Amount of days an item can be early and still considered On Time.		OM	M
SALES_LATE_TOLERANCE_DELAY	0	2	Amount of days an item can be late and still considered On Time.		OM	M
SALES_LOAD_DAYS	3000	5000	The number of days to load from the look back days forward for SALES related transactional tables	Obsolete	OM	M
SALES_LOAD_LOOKBACK_DAYS	2000	5000	The number of days to look back for changes in QAD in a table for SALES related transactional tables	Obsolete	OM	M
SALES_LOOKBACK_DAYS	60	60	The number of days to look back in the ih_hist and idh_hist tables to pick up changed records.		OM	M
SALES_ORDER_HISTORY_DATE	<Set to today's date before starting the initial history load>	2011-01-30	The processing date used for SALES orders history calculations Format is YYYY-MM-DD		OM	M
SALES_ORDER_PERF_DATE		2011-01-28	The processing date used for SALES order performance table Format is YYYY-MM-DD		OM	P

Table 4.4

Parameters for the Order Management Module (Page 3 of 5)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
SALES_ORDER_REVALUE_DAYS	10	10	The number of days after the end of the period to wait for revalue exchange rates to be entered before revaluing order history		OM	M
SALES_ORDER_REVALUE_PERIODS	1	1	The number of periods to revalue in order history		OM	M
SALES_PERCENT_QTY_OVER	0	2	Percentage over the ordered amount that can be delivered and the order still considered delivered in full.	Added in 3.7	OM	M
SALES_PERCENT_QTY_UNDER	0	3	Percentage under the ordered amount that can be delivered and the order still considered delivered in full.	Added in 3.7	OM	M
SALES_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the SALES process job		OM	P
SALES_PROCESS_DAYS	10	20	The number of days of sales to extract and process from permanent staging tables.		OM	M
SALES_PROCESS_RUNNING	N	N	Indicates if a SALES process is currently running (Y,N)	Used to prevent multiple copies of the Order Management/Sales job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from "N" from "Y".	OM	P

Table 4.4
Parameters for the Order Management Module (Page 4 of 5)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
SALES_PROCESS_RUNNING_JOB_NAME		DAILY_SALE S_PROCESS_2 010SE	Inserted via WsParameterWrite		OM	P
SALES_SHIPMENT_FILTER	N	N	Filter the shipment data to remove consignments		OM	M
SALES_SNAPSHOT_DAYS	31	31	Number of days to look back when processing sales order snapshot		OM	M
SALES_SNAPSHOT_DOW	1	1	Day of Week Number of the Sales Snapshot Extract. Use % for all days.		OM	M
SALES_SNAPSHOT_FREQUENCY	M	M	The frequency of snapshots in the Sales order snapshot table. Possible values are: W for Weekly, M for Monthly, B for both weekly and monthly.		OM	M
SALES_SNAPSHOT_HIST_START_DATE	<Set to today's date before starting the initial history load>	2011-05-24	Used by the HIST_SALES_ROLLUP job to determine the date at which to start taking Sales order snapshots. Set to today's date by the INITIAL_SETUP_JOB, but can be set to an earlier date by the user if more history in the snapshot is desired.		OM	P

Table 4.4

Parameters for the Order Management Module (Page 5 of 5)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
SALES_SNAPSHOT_REPLACE_FLAG	N	N	Indicates snapshots are replaced or not. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to just add new records and not update previously calculated records. Default value is N.	OM	M
SALES_SNAPSHOT_START_DATE		2011-05-24	Internal use only. The setup procedure for the Sales snapshot table updates this parameter to determine the date at which to start by subtracting the SALES_SNAPSHOT_DAYS value from the current date.		OM	P

Operations Module

Table 4.5

Parameters for the Operations Module (Page 1 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
IH_INVOICETYPE	0	ih_hist.ih_invoicetype	Internal use only. Specifies the column name in ih_hist for invoice_type, or 0 if not supported on this version of the source system. There are two invoice type columns in the EE version of ih_hist. This allows us to capture both.	When extracting from EE sources, invoice type needs to reference a new column in the ih_hist table.	OPS	P
INV_BALANCE_HIST_START_DATE	2008-01-01	2008-01-01	The START date used for Inventory Transactions history loading		OPS	M

Table 4.5
Parameters for the Operations Module (Page 2 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INV_HISTORICAL_PERIOD_COUNT	-24	-24	Number of periods to use to determine Average daily issued quantity historical.		OPS	M
INV_HISTORY_DATE	<Set to today's date before starting the initial history load>	2012-08-28	The first processing date for loading inventory. Format is YYYY-MM-DD. Set Programmatically by INITIAL_JOB_SETUP		OPS	P
INV_LAST_IN_MSTR_EXTRACT		40574.53221	Last Extract Timestamp of data currently in load_in_mstr		OPS	P
INV_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the Inventory process job		OPS	P
INV_PROCESS_DAYS	10	10	The number of days of Inventory to extract and process from permanent Data Store tables		OPS	M
INV_PROCESS_RUNNING	N	N	Indicates if an Inventory process is currently running (Y,N)	Used to prevent multiple copies of the Inventory job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from to "N" from "Y".	OPS	P
INV_PROCESS_RUNNING_JOB_NAME		DAILY_INV_PROCESS_CHANGED	Indicates the current or last run job name for INVENTORY processing.		OPS	P

Table 4.5

Parameters for the Operations Module (Page 3 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INV_PROJECTED_DAYS_ON_HAND_MAX	999	99999	If there is no mrp_demand to calculate projected days on hand, the number to display to represent an infinite amount of days.		OPS	M
INV_REVALUE_PERIODS	100	10	The number of periods to revalue in inventory balances		OPS	M

Table 4.5
Parameters for the Operations Module (Page 4 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INV_TRANS_TYPES	'CST-ADJ','CST-TR','CYC-CNT','CYC-ERR','CYC-RCNT','ISS-CHL','ISS-COR','ISS-DO','ISS-FAS','ISS-GIT','ISS-POIT','ISS-PRV','ISS-RV','ISS-SCRIP','ISS-SO','ISS-TR','ISS-UNP','ISS-WO','PLCG-ADJ','RCT-CHL','RCT-DO','RCT-FAS','RCT-GIT','RCT-PO','RCT-POIT','RCT-RS','RCT-SOR','RCT-TR','RCT-UNP','RCT-WO','RJCT-WO','TAG-CNT','CN-CNT'		Transaction types (tr_type) that are related to inventory changes		OPS	Do Not Edit
INV_TRANS_TYPES_CONS	'CN-SHIP','CN-USE','CN-ADJ','CN-RCT','CN-ISS'	'CN-SHIP','CN-USE','CN-ADJ','CN-RCT','CN-ISS'	Transaction types (tr_type) that are related to consignment inventory movements		OPS	Do Not Edit

Table 4.5

Parameters for the Operations Module (Page 5 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INV_TRANS_TYPES_COUNT_ERR	'CYC-ERR'	'CYC-ERR'	Transaction types (tr_type) that are related to count errors		OPS	Do Not Edit
INV_TRANS_TYPES_COUNT_TR	'CYC-CNT'	'CYC-CNT'	Transaction types (tr_type) that are related to count transactions		OPS	Do Not Edit
INV_TRANS_TYPES_CUST	'ISS-SO','RCT-SOR','ISS-COR'	'ISS-SO','RCT-SOR','ISS-COR'	Transaction types (tr_type) that are related to customers		OPS	Do Not Edit
INV_TRANS_TYPES_GIT	'ISS-GIT','ISS-POIT','RCT-GIT','RCT-POIT'	'ISS-GIT','ISS-POIT','RCT-GIT','RCT-POIT'	Transaction types (tr_type) that are related to goods in transit (GIT).		OPS	Do Not Edit
INV_TRANS_TYPES_ISSUED	'ISS_SO','ISS-COR','ISS-WO','ISS-FAS','ISS-DO','ISS-RMA'	'ISS_SO','ISS-COR','ISS-WO','ISS-FAS','ISS-DO','ISS-RMA'	Transaction types (tr_type) that are related to inventory issued		OPS	Do Not Edit
INV_TRANS_TYPES_RECEIVED	'RCT-FAS','RCT-PO','RCT-WO','RCT-DO'	'RCT-FAS','RCT-PO','RCT-WO','RCT-DO'	Transaction types (tr_type) that are related to inventory received		OPS	Do Not Edit
INV_TRANS_TYPES_RECOUNT	'CYC-RCNT'	'CYC-RCNT'	Transaction types (tr_type) that are related to re-count transactions		OPS	Do Not Edit
INV_TRANS_TYPES_SALES	'ISS-SO','ISS-COR'	'ISS-SO','ISS-COR'	Transaction types (tr_type) that are related to sales orders		OPS	Do Not Edit
INV_TRANS_TYPES_SUPPLIER	'ISS-PRV','RCT-PO'	'ISS-PRV','RCT-PO'	Transaction types (tr_type) that are related to suppliers		OPS	Do Not Edit
JOB_INV_PROCESS_BATCH_ID	0	754	The batch id for the current DAILY_INV_PROCESS_JOB job.		OPS	P

Table 4.5
Parameters for the Operations Module (Page 6 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
JOB_OP_PROCESS_BATCH_ID	0	755	The batch id for the current DAILY_OP_PROCESS_JOB job.		OPS	P
JOB_PO_PROCESS_BATCH_ID	0	752	The batch id for the current DAILY_PO_PROCESS_JOB job.		OPS	P
OP_HIST_LOAD_MAXSIZE	5000000	5000000	The maximum number of rows to load in an iteration loading op_hist		OPS	P
OP_HIST_LOAD_RUNNING	C	C	Indicates if a load of op_hist is running (Y/N/C)		OPS	P
OP_HIST_TRAN_MAX		19975	DON'T UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		OPS	P
OP_HIST_TRAN_MAX_Sxx		433	DON'T UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		OPS	P
OP_LOAD_LOOKBACK_DAYS	20	20	The number of days to look back in QAD when extracting OP data	Used to limit the amount of data extracted from wo_mstr, wod_det, and wr_route to just those open or closed between today and (today - lookback days)	OPS	M
OP_MOVING_AVG_WEEKS	13	13	The Number of Weeks to base Moving Average on.		OPS	M
OP_OEE_SNAPSHOT_FREQUENCY	B	B	The frequency of snapshots in the OP OEE snapshot tables. Possible values are: W for Weekly, M for Monthly, B for both weekly and monthly.		OPS	M

Table 4.5

Parameters for the Operations Module (Page 7 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
OP_PROCESS_CONNECTION_NAME		QMI	Indicates the connection name used by the OP process job.		OPS	P
OP_PROCESS_DAYS	5	5	The number of days of OP to extract and process from permanent staging tables.		OPS	M
OP_PROCESS_RUNNING	N	N	Indicates if a OP process is currently running (Y,N)		OPS	P
OP_PROCESS_RUNNING_JOB_NAME		DAILY_OP_PROCESS_CHAINED	Indicates the current or last run job name for module processing.		OPS	P
PO_BOOKING_DATE	<Set to today's date before starting the initial history load>	2011-01-28	The processing date used for PO bookings processing. Format is YYYY-MM-DD		OPS	M/P
PO_BOOKING_FIRST_DATE	<Set to today's date before starting the initial history load>	2011-01-28	The first processing date for loading bookings. Format is YYYY-MM-DD		OPS	M/P
PO_EARLY_TOLERANCE_DELAY	2	4	Amount of days an order can be early and still be considered as "on-time".		OPS	M
PO_LATE_TOLERANCE_DELAY	2	1	Number of days a PO can be late and still considered as on time.		OPS	M
PO_LOOKBACK_DAYS	30	60	Set to the number of days the user is able to backdate the receipt date. This is used by the load table load_prh_hist.		OPS	M

Table 4.5
Parameters for the Operations Module (Page 8 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
PO_ORDER_HISTORY_DATE	<Set to today's date before starting the initial history load>	2011-01-28	The processing date used for PO orders history calculations. Format is YYYY-MM-DD		OPS	P
PO_PERCENT_QTY_OVER	10	10	Percentage of Receipt Quantity over Order Quantity for a Purchase Order to still be considered In Full		OPS	M
PO_PERCENT_QTY_UNDER	20	20	Percentage of Receipt Quantity under Order Quantity for a Purchase Order to still be considered In Full		OPS	M
PO_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name used by the PO process job.		OPS	P
PO_PROCESS_DAYS	10	10	The number of days of PO to extract and process from permanent staging tables.		OPS	M
PO_PROCESS_RUNNING	N	N	Indicates if a PO process is currently running (Y,N)	Used to prevent multiple copies of the Purchasing job from running at the same time. Under normal conditions you should not update this manually. However, if a job fails and is deleted, you may need to reset this to "N" to run the job again. But use caution and ensure that there really is no copy of the job running before changing from "N" from "Y".	OPS	P
PO_PROCESS_RUNNING_JOB_NAME		DAILY_PO_PROCESS_2010SE	Indicates the current or last run job name for PO processing.		OPS	P

Table 4.5

Parameters for the Operations Module (Page 9 of 9)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
PO_RET_QTY_TOLERANCE	15	15	Percentage of Returns Quantity used for calculating Defect Rate.		OPS	M
PO_SNAPSHOT_FREQUENCY	B	M	The frequency of snapshots in the PO order snapshot table. Possible values are: W for Weekly, M for Monthly, B for both weekly and monthly.		OPS	M
PO_SNAPSHOT_HIST_START_DATE	<Set to today's date before starting the initial history load>	2011-01-28	The first date in the PO order snapshot table in the format YYYY-MM-DD for the first run		OPS	P
PO_SNAPSHOT_NUM_DAYS	31	31	The number of days of snapshots to load back each run.		OPS	M
PO_SNAPSHOT_REPLACE_FLAG	N	N	Indicates snapshots are replaced or not. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to just add new records and not update previously calculated records. Default value is N.	OPS	M
PO_SNAPSHOT_START_DATE		1997-05-24	Internal use only. The setup procedure for the PO snapshot tables updates this parameter to determine the date at which to start.		OPS	P

Enterprise Asset Management Module

Table 4.6

Parameters for the Enterprise Asset Management Module (Page 1 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
EAM_CORP_CALENDAR_DOMAIN	<If EAM module installed - enter domain code for EAM Corporate financial calendar>	QP	The Domain Code used for the EAM corporate financial calendar.		EAM	M
EAM_CORP_CALENDAR_SOURCE	<If EAM module installed - Enter data source name here for EAM calendar setup>	QMI	The Source System Code used for the EAM corporate financial calendar.		EAM	M
EAM_DEFAULT_DOMAIN	0000	0000	Default Domain to use when loading the EAM Data, if no valid domain exists.		EAM	M
EAM_DEFAULT_ENTITY	000000	000000	Default entity to use when loading the EAM Data, if no valid entity exists.		EAM	M
EAM_DEFAULT_SOURCE		Unknown Source	The default value used for unknown EAM source system, if no valid source system exists.		EAM	M
EAM_PROCESS_CONNECTION_NAME		EAM QPS	EAM_QPSInternal Use Only. Indicates the connection name last used by the EAM process job.		EAM	P
EAM_PROCESS_DAYS	3	3	Internal Use Only.. Set to either EAM_PROCESS_DAYS_DAILY or EAM_PROCESS_DAYS_HIST, depending on type of run.		EAM	P

Table 4.6

Parameters for the Enterprise Asset Management Module (Page 2 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
EAM_PROCESS_RUNNING_JOB_NAME		DAILY_EAM_PROCESS_C HAINED	Internal Use Only. Indicates the current or last run job name for module processing.		EAM	P
EAM_WO_ORDER_HISTORY_DATE	2012-08-05	2012-08-05	Generally internal use only. Set to current date, unless DAILY_HISTORY_DATE_SET_CURRENT = N.		EAM	P
EAM_MAINT_COST_SNAP_DAYS	31	31	Internal Use Only. Number of days to look back when processing EAM work order snapshot.		EAM	P
EAM_MAINT_COST_SNAP_DAYS_DAILY	31	31	Change as needed. Number of days to look back when processing EAM work order snapshot during a DAILY run.		EAM	M
EAM_MAINT_COST_SNAP_DAYS_HIST	3650	1825	Change before initial loads, as needed. Number of days to look back when processing EAM work order snapshot during a HISTORY run.		EAM	M
EAM_MAINT_COST_SNAP_FREQUENCY	M	M	Change before initial loads, as needed. The frequency of snapshots in the EAM maintenance cost snapshot table. Possible values are: W for Weekly, M for Calendar Monthly, B for both weekly and monthly, F for Fiscal Period.		EAM	M

Table 4.6
Parameters for the Enterprise Asset Management Module (Page 3 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
EAM_PROCESS_DAYS_DAILY	10	3	Change as needed. The number of days of EAM to extract and process from source system files and permanent staging tables for the DAILY jobs.		EAM	M
EAM_PROCESS_DAYS_HIST	3650	1825	Change before initial loads, as needed. The number of days of EAM to extract and process from source system files and permanent staging tables for the HIST jobs.		EAM	M
EAM_PROCESS_RUNNING	N	N	Internal Use Only. Indicates if a EAM process is currently running (Y,N).		EAM	P
EAM_WO_SNAP_DAYS	31	31	Internal Use Only. Number of days to look back when processing EAM work order snapshot.		EAM	P
EAM_WO_SNAP_DAYS_DAILY	31	31	Change as needed. Number of days to look back when processing EAM work order snapshot during a DAILY run.		EAM	M

Table 4.6

Parameters for the Enterprise Asset Management Module (Page 4 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
EAM_WO_SNAP_DAYS_HIST	3650	1825	Change before initial loads, as needed. Number of days to look back when processing EAM work order snapshot during a HISTORY run.		EAM	M
EAM_WO_SNAP_FREQUENCY	M	M	Change before initial loads, as needed. The frequency of snapshots in the EAM Work Order snapshot table. Possible values are: W for Weekly, M for Calendar Monthly, B for both weekly and monthly, F for Fiscal Period.		EAM	M

Transportation Management Module

Table 4.7

Parameters for the Transportation Management Module (Page 1 of 3)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
JOB_TM_PROCESS_BATCH_ID	1506		The batch id for the current DAILY_TM_PROCESS_JOB job.		TMS	P
TM_EXCHANGE_RATE_TYPE	FIXED		Determines what Exchange Rate Type to use for currency conversion (FIXED, SPOT, etc).		TMS	M
TM_JOB_CHAINING_ENABLED	N		Y if a job starts the next TMS job in the list automatically as it ends, N otherwise.		TMS	M

Table 4.7
Parameters for the Transportation Management Module (Page 2 of 3)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TM_LOAD_JOB00XX			TM Job number 00XX	TM modules jobs in the proper order. Do not modify; add any custom jobs to the end of the job chain.	TMS	M
TM_LOAD_LOOKBACK_DAYS	428		The number of days to look back into TMS when extracting data.		TMS	M
TM_LOAD_SINGLE_SOURCE	N		Flag to identify whether to load only a single source or all defined sources (Y/N).		TMS	M
TM_LOAD_SOURCE_S01			Name of the source system loaded as the 01 job.		TMS	M
TM_LOAD_TRANS_TYPES	'S', 'T'				TMS	M
TM_LOAD_TRANS_TYPES_S01	'S', 'T'				TMS	M
TM_PROCESS_CONNECTION_NAME			Indicates the connection name used by the TM process job.		TMS	P
TM_PROCESS_DAYS	100		The number of days of TM to extract and process from permanent staging tables.		TMS	M
TM_PROCESS_RUNNING	N		Flag indicating if TM Process is running. Y/N		TMS	P
TM_TOLERANCE_DAYS	3		Grace period (in days) after which package is deemed to be failed delivery		TMS	M
TM_WEIGHT_CAT_MAX	150		MAX Weight Value to individually categorize		TMS	M
TM_WEIGHT_CAT_UNITS	1		Units to increment each weight category. Value MUST be greater than 0		TMS	M
TM_WEIGHT_CAT_UOM	LB		Unit of Measure for Weight Category Buckets (LB or KG)		TMS	M

Table 4.7

Parameters for the Transportation Management Module (Page 3 of 3)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TM_SourceConvert_SchemaORA	p2009dev		The name of the Oracle schema. Not necessary for Progress source because it always uses PUB.		TMS	M
TM_SourceConvert_TargetSource	oracle		Tells the conversion procedure what the type of db the load tables will source from. Values either: oracle or progress		TMS	M
TM_SourceConvert_TMSMSTRConnectionName	TMS_MASTER		The name of the connection the TMS load tables are using. ***All load tables using this connection are updated.		TMS	M

Table 4.8
System Information

Related to:	Information Needed
Progress Database	Have they already set up ODBC connections to Progress? If not, please answer the following Progress database questions for each data source being implemented.
Progress Database	ERP Version
Progress Database	Progress Version (must be 10.1B03 or above)
Progress Database	Preferred connection name.
Progress Database	Host Name
Progress Database	Port Number
Progress Database	Database Name
Progress Database	User ID
Progress Database	Password
Tomcat	URL
Tomcat	Port Number
SQL Server	Host Name
SQL Server	Port Number
Email for Portal	Host Name
Email for Portal	Port Number
Email for Portal	Authorization? Y or N

Table 4.9

Parameter Values that Need to Be Set Before History Load (Page 1 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
AP_EE_HISTORY_DATE_MIN	<customer defined - usually three years back. YYYYMMDD>	The date the AP aging table is to be refreshed from. Three years back is common. For example, 20081231		FIN
AP_EE_SNAPSHOT_HIST_START_DATE	2008-12-31	The first date in the AP aging snapshot table in the format YYYY-MM-DD for the first run	Date used for start of AP snapshot in History load. Used for History load only; Daily loads use AP_EE_SNAPSHOT_NUM_DAYS	FIN
AP_SE_SNAPSHOT_HIST_START_DATE	2009-01-01	The first date in the AP aging snapshot table in the format YYYY-MM-DD for the first run		FIN
AR_EE_HISTORY_DATE_MIN	<customer defined - usually three years back. YYYYMMDD>	The date the AP aging table is to be refreshed from. Three years back is common. For example, 20081231		FIN
AR_EE_SNAPSHOT_HIST_START_DATE	2008-12-31	The first date in the AR aging snapshot table in the format YYYY-MM-DD for the first run		FIN

Table 4.9
Parameter Values that Need to Be Set Before History Load (Page 2 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
AR_SE_SNAPSHOT_HIST_START_DATE	2009-01-01	The first date in the AR aging snapshot table in the format YYYY-MM-DD for the first run		FIN
CORP_CURRENCY_CODE	USD	Corporate currency code used for high level reporting	The currency code for corporate amounts calculations. Same for entire warehouse.	COM
DAILY_LOAD_SOURCE_S01	<Enter data source name here for initial setup>	Name of the source system loaded as the 01 job.	The connection name of the first or primary source system.	COM
DAILY_LOAD_SOURCE_S02 - DAILY_LOAD_SOURCE_Sxx	Does not exist by default; add if additional source(s)		If needed, the connection name(s) of the second and subsequent source systems	COM
DEFAULT_ACCOUNT	00000	The default value used for unknown account	Must not be a real account code	COM
DEFAULT_COST_CENTER	0000	The default value used for unknown cost center	Must not be a real cost center code	COM
DEFAULT_DOMAIN	00000	The default value used for unknown domain	Must not be a real domain code	COM
DEFAULT_ENTITY	0000	The default value used for unknown entity	Must not be a real entity code	COM
DEFAULT_PROJECT	00000	The default value used for unknown project	Must not be a real project code	COM
DEFAULT_SOURCE		The default value used for unknown source system	Must not be a real source system code	COM

Table 4.9

Parameter Values that Need to Be Set Before History Load (Page 3 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
DEFAULT_SUB_ACCOUNT	0000000	The default value used for unknown sub-account	Must not be a real sub-account code	COM
GL_LOAD_DAYS	152	The number of days to load from the look back days forward for GL related transactional tables	NOTE: Set to a high value for History loads to load all data desired for historical purposes. Then set to a lower value for Daily loads. Should always be larger than GL_LOAD_LOOKBACK_DAYS.	FIN
GL_LOAD_LOOKBACK_DAYS	150	The number of days to look back for changes in QAD in a table for GL related transactional tables.	NOTE: Set to a high value for History loads to load all data desired for historical purposes. Then set to a lower value for Daily loads. Should always be smaller than GL_LOAD_DAYS.	FIN
GL_PERIOD1_FISCAL_ACCOUNT_ZERO_TYPES		Account types to set to zero in fact_gl_balances when a new fiscal period starts. If null, none are set. The proper values in this parameter should be a list of account codes such that the string would be appropriate in an "IN" clause.		FIN
HIST_LOAD_SOURCE_S01	<Enter data source name here for initial setup>	QMI	Name of the source system loaded as the 01 job.	COM
HIST_LOAD_SOURCE_S02 - HIST_LOAD_SOURCE_Sxx	Does not exist by default; add if additional source	2010SE		COM
INITIAL_JOB_SETUP_CONNECTION_01	<Enter data source name here for initial setup>	Name of the connection 01 used as a source system for cloning jobs.		COM

Table 4.9
Parameter Values that Need to Be Set Before History Load (Page 4 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
INITIAL_JOB_SETUP_CONNECTION_01_RUN	N	This determines if the Initial Job Setup has been run against this data source yet or not. Any new data source should have an N. Value is set to C when historic load has completed.	Used by INITIAL_JOB_SETUP job to ensure that existing data sources are NOT re-run as part of the historic load when new sources are added.	COM
INITIAL_JOB_SETUP_CONNECTION_01_TYPE	<Enter source type - SE, EE, or EAM>	Connection Type - SE for Standard Edition Financials, EE for Enterprise Edition Financials, and EAM for Enterprise Asset Management. Defaults to SE	Used to determine whether to set up EE Financial or SE Financial jobs for this data source.	COM
INITIAL_JOB_SETUP_CONNECTION_02_DAILY_MODULES	COM, OM, OP, FIN, EAM	Example: 'COM, FIN' - List Modules to be part of DAILY job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM
INITIAL_JOB_SETUP_CONNECTION_02_HIST_MODULES	COM, OM, OP, FIN, EAM	Example: 'COM, FIN' - List Modules to be part of HIST job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM
INITIAL_JOB_SETUP_CONNECTION_02		Name of the connection 02 used as a source system for cloning jobs.		COM
INITIAL_JOB_SETUP_CONNECTION_02_RUN		This determines if the Initial Job Setup has been run against this data source yet or not. Any new data source should have an N.	Added in 3.4.1 – Used by INITIAL_JOB_SETUP job to ensure that existing data sources are NOT re-run as part of the historic load when new sources are added.	COM

Table 4.9

Parameter Values that Need to Be Set Before History Load (Page 5 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
INITIAL_JOB_SETUP_CONNECTION_02_TYPE	<Enter source type - SE, EE, or EAM>	Connection Type - SE for Standard Edition Financials, EE for Enterprise Edition Financials, EAM for Enterprise Asset Management. Defaults to SE.		COM
INITIAL_JOB_SETUP_CONNECTION_02_DAILY_MODULES	COM, OM, OP, FIN, EAM	Example: 'COM, FIN' - List Modules to be part of DAILY job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM
INITIAL_JOB_SETUP_CONNECTION_02_HIST_MODULES	COM, OM, OP, FIN, EAM	Example: 'COM, FIN' - List Modules to be part of HIST job chain.	Added in 3.7 - Set to list of modules appropriate to the source. Default is fine for most installations as any uninstalled modules are skipped.	COM
INITIAL_JOB_SETUP_DATE	<Set to today's date before starting the initial history load>	Added in 3.4.1. INITIAL_JOB_SETUP uses this date to populate the majority of the previously manually populated date fields for job setup.		COM

QAD BI Upgrade Instructions

This chapter includes instructions for upgrading your QAD BI environment to the current version.

Upgrade Information 96

Upgrade to QAD BI 3.7 from QAD BI 3.6.x 96

Upgrade to QAD BI 3.6 from QAD BI 3.5 105

Upgrade to QAD BI 3.5 from QAD BI 3.4.1 113

Upgrade Information

To upgrade to the latest version of QAD BI, you must upgrade one version at a time until you reach the current released version. For example, if you are using QAD BI 3.4.1 and are upgrading to QAD BI 3.7, first upgrade from BI 3.4.1 to 3.5, then upgrade from BI 3.5 to 3.6, and then upgrade from BI 3.6 to 3.7.

Upgrade to QAD BI 3.7 from QAD BI 3.6.x

Preparation

- 1 Plan for a couple of hours of down time.
- 2 Ensure that there are no loads currently in progress. All jobs must be completed.
- 3 Put all scheduled jobs On Hold until the upgrade is complete. Shut down the scheduler.
- 4 Take a backup of the current warehouse database and the current portal database.
- 5 Before you begin, make note of the following:

- Note any custom DAILY_LOAD_JOBxxx jobs that appear after the final DAILY_MODULE_LOADS_FINISH jobs: _____
- Note any custom HIST_LOAD_JOBxxx jobs that appear after the final POSTPROCESS_module_SUMMARY job: _____

When INITIAL_JOB_SETUP is run during the upgrade procedure to rebuild the history and daily job chains, any custom jobs should be placed at the end of the chains, but you should note your custom jobs here and ensure that all jobs appear in the rebuilt job chain.

Important This upgrade will replace some load and stage tables. If there are any customizations to these tables, ensure that they are ready to be made again.

Upgrade Steps for QAD BI Data Warehouse Designer

- 1 Install the new QAD BI Data Warehouse Designer, version 6.6.2, by running the `QAD_66200.msi` installer application.
- 2 Run the new version 6.6.2 Setup Administrator. Choose Metadata Repository -> Validate Metadata Repository.
- 3 Select the ODBC DSN of the DWD you want to upgrade and click OK.
- 4 Verify that the screen says “Version is 6.5. Current version is 6.6.” When the system prompts you to validate all Meta Tables, click OK.
- 5 A dialog box says “Do you wish to upgrade?” Click OK.
- 6 A list of update commands is displayed. Click OK. At the “Define FileGroups” dialog click OK.

- 7 Launch the new version 6.6.2 BI Data Warehouse Designer and verify that you can connect to your upgraded Data Warehouse.

BI Portal Upgrade Procedure

Preparation

- Shut down Tomcat.
- Back up the BI Portal database.

Upgrade Steps for QAD BI Portal

- 1 Copy the `TOMCAT_HOME\webapps\your_bi_appname` directory to `TOMCAT_HOME\your_bi_appname.save36` so that you have a backup of your BI 3.6 web application.
- 2 Unzip the new `BI 3.7 Portal data.zip` file into `TOMCAT_HOME\webapps\your_bi_appname`, overwriting any existing files with the newer version from the `data.zip` file. Your configuration files will not be overwritten.
- 3 In SQL Server Management Studio, run the upgrade script:
`TOMCAT_HOME\webapps\your_bi_appname\install\sqlserver\v36tov37.sql`.
- 4 Start Tomcat.

Note Users may need to clear their browser cache prior to reconnecting to the BI Portal to ensure that they are running the latest version of the application. Go to the Portal URL in the browser and click Refresh to make sure that the latest version of the application is loaded.

Upgrade Steps for QAD BI Metadata

- 5 Unzip the metadata for the Common module and any other modules you have installed.
Note Because the metadata files are now segregated by module, it is only necessary to install the modules that are needed.
 - Perform the Common upgrade first. The other upgrades can be done in any order.
 - All users need to run the Common metadata upgrade application.
 - If you process any of the Order Management (Sales) `fact_om_*` tables, install the Order Management (OM) upgrade application.
 - If you process Inventory and/or Purchase Order tables, install the Operations module upgrade application.
 - If you process any Financial tables, install the Financials module upgrade application.
- 6 Log in to Setup Administrator.
- 7 Choose Tools|Start Logging. Select a name and location for your log file and save.

8 Load the Common Module Upgrade Application as follows:

- a** Set your application directory to the new *metadata/common* folder.
- b** Load application *upgradeCom* by right-clicking on the file and choosing Install Application.
- c** From the drop-down list, select the ODBC DSN for your DWD instance.
- d** The system prompts you to proceed. Click OK.

Note You can cancel the installation here to review the list of new and modified objects. To continue with the load, right-click in the window and choose Proceed with Load Application.

- e** The next dialog box is Application Load Properties. Verify that the default values are correct. Make any changes if necessary:
 - On the left, select Dimension. On the right, click on the box next to *Existing Dimension objects will be* and select Altered.
 - On the left, select Dimension View. On the right, click on the box next to *Existing Dimension View objects will be* and select Recreated.
 - On the left, select Stage Table. On the right, click on the box next to *Existing Stage table objects will be* and select Recreated.
 - On the left, select Permanent Stage Table. On the right, click on the box next to *Existing Permanent Stage table objects will be* and select Altered.
 - On the left, select Fact Table. On the right, click on the box next to *Existing Fact table objects will be* and select Altered.
 - Click OK.

9 Repeat for any other modules that must be upgraded.

- **Order Management.** Change the Application Directory to the *metadata/order_management* subdirectory and install application *upgradeOM*.
- **Operations.** Change the Application Directory to the *metadata/operations* subdirectory and install application *upgradeOps*.
- **Financials.** Change the Application Directory to the *metadata/financials* subdirectory and install application *upgradeFin*.

Note The upgrade application installations will take some time as the load process alters and re-creates various objects.

10 In SQL Server Management Studio, run the *misc.sql* script in your Data Warehouse database:

<Location of unpacked Common module>\sqlserver\misc.sql

This will enable an extra “Tech Notes” tab in your Data Warehouse Designer.

11 Start the Scheduler. Make sure all jobs are still suspended.**12** If the upgrade affected any customized tables, reapply your customizations. For a list of new and modified tables, see “Data Warehouse Tables Changed” on page 101.

- 13** If there were any customizations previously added to any of the modified jobs, update the job to include those extra parts. You can double check what was in the job by looking at the newly renamed `<jobname>_1`. The jobs that changed in this release are:

- CUSTOMER_PROCESS
- DAILY_JOB_SETUP
- HIST_JOB_SETUP
- DAILY_COMMON_PROCESS_CHAINED
- PREPROCESS_COMMON_LOADS
- PREPROCESS_COMMON_TRUNCATE
- CLEANUP_COMMON_TABLES
- HIST_COMMON_PROCESS_CHAINED
- HIST_MODULES_LOAD_FINISH
- INITIAL_JOB_SETUP
- PREPROCESS_COMMON_EEONLY_LOADS
- CLEANUP_OPERATIONS_TABLES
- PREPROCESS_OP_LOAD
- PREPROCESS_OP_TRUNCATE
- DAILY_OP_PERM_EXTRACT
- HIST_OP_PERM_EXTRACT
- DAILY_OP_INV_PROCESS_CHAINED
- HIST_OP_INV_PROCESS_CHAINED
- DAILY_OP_PO_PROCESS_CHAINED
- HIST_OP_PO_PROCESS_CHAINED
- POSTPROCESS_OP_CUBES
- CLEANUP_OM_TABLES
- DAILY_OM_PROCESS_CHAINED
- HIST_OM_PROCESS_CHAINED
- POSTPROCESS_OM_CUBES
- CLEANUP_FIN_TABLES
- DAILY_FIN_EE_AP_PROCESS_CHAINED
- DAILY_FIN_EE_AR_PROCESS_CHAINED
- DAILY_FIN_EE_GL_PROCESS_CHAINED
- DAILY_FIN_EE_PERM_CHAINED
- DAILY_FIN_SE_PERM_CHAINED
- HIST_FIN_EE_AP_PROCESS_CHAINED
- HIST_FIN_EE_FINISH_CHECK
(was HIST_EE_AP_AR_FINISH_CHECK)
- HIST_FIN_EE_GL_PROCESS_CHAINED
- HIST_FIN_EE_PERM_CHAINED
- HIST_FIN_SE_PERM_CHAINED
- POSTPROCESS_FIN_SE_AP_SUMMARY
- POSTPROCESS_FIN_SE_CUBES
- PREPROCESS_FIN_EE_LOADS
- PREPROCESS_FIN_TRUNCATE

- 14** There are several new parameters (as of BI 3.6.1) to control how the BI system handles the rollover of the `tr_trnbr` – ensure that these are set appropriately for your system:

- **TR_HIST_PROCESS_DAYS.** This is the number of look-back days when extracting data from `tr_hist`. The records being extracted are records where `tr_trnbr` is greater than the highest `transaction_number` records already extracted for that source, except after `tr_trnbr` rollover, when newer transactions may have smaller `trnbrs` than older transactions. The BI DW must also take into account the transaction date in order to find the lower-numbered transactions that are actually new. This parameter says to extract from the ERP transactions entered within the last X days. If the number of days since the `TR_HIST_DATE_MAX_Sxx` is greater than the `TR_HIST_PROCESS_DAYS` parameter, a task that checks this during the run will fail and indicate this number needs to be increased. The default value is 3.
- **TR_HIST_MAX_VALUE.** This is the maximum number allowed for the `tr_hist.tr_trnbr` field, or in other words, the number after which the `tr_trnbr` will roll over and start at the first available low number. The default value is 99999999. If the `tr_sq01` sequence is set to a different maximum value, enter that here.
- **TR_HIST_PRE_ROLLOVER_COUNT.** This is the number of records that would typically be in three days' worth of data to ensure readiness for `tr_hist.tr_trnbr` rollover. Once the `tr_trnbr` for a source is detected to be within `TR_HIST_PRE_ROLLOVER_COUNT` of `TR_HIST_MAX_VALUE`, the BI system sets

the TR_HIST_TRAN_MAX_Sxx value for the source to 0. This will force the tr_hist extraction to scan the whole table for records that have been processed within the last TR_HIST_PROCESS_DAYS. The default value is 5000.

- **TR_HIST_PCT_OF_MAX_REF.** This is the percent of the TR_HIST_MAX_VALUE to use as a reference point to look for new low-numbered transactions when dealing with the rollover. The default value is 50. When this value is 50, you will search the transactions with numbers less than 50% of the TR_HIST_MAX_VALUE for the highest number with a transaction date within the processing window.

15 If you already installed the BI 3.6.1 or BI 3.6.2 patch, skip ahead to Step 16.

If you are upgrading directly from BI 3.6, run the Upgrade jobs for the tr_hist.tr_trnbr roll-over issue for each module you have installed:

- **Common Module.** Run the UPGRADE_36_to_361_QBIS_25_COMMON job to update the permsup_transaction_hist table to include the transaction date and to seed the TR_HIST_DATE_MAX values for each source.
- **Order Management.** Run the UPGRADE_36_to_361_QBIS_25_OM job to update the fact_om_booking table to include the transaction date.
- **Operations.** Run the UPGRADE_36_to_361_QBIS_25_OP job to update the fact_op_transaction table to include the transaction date.

16 Run the upgrade jobs for BI 3.7:

- a Set the JOB_CHAINING_ENABLED parameter to N.
- b Run the UPGRADE_36_TO_37_QBI_2121 job.
- c If you use the Order Management module, run the UPGRADE_36_to_37_OM job.
- d If you use the Operations module, run the UPGRADE_36_to_37_PO job.

17 There are two new parameters for each connection:

- INITIAL_JOB_SETUP_CONNECTION_xx_DAILY_MODULES
- INITIAL_JOB_SETUP_CONNECTION_xx_HIST_MODULES.

The upgrade will install copies of this parameter for connections 01 and 02. If you have more than two connections, add additional copies of this parameter for each additional connection. These parameters are used by the INITIAL_JOB_SETUP job in determining the Daily and History job chains, respectively. By default the parameters for the 01 and 02 connections contain all ERP modules (COM, OM, OPS, FIN). Remove any modules that you do not use from this list.

18 Set INITIAL_JOB_SETUP_ENABLED to Y and run INITIAL_JOB_SETUP to rebuild the daily and historical job chains.

19 If you use the Operations module, truncate the fact_inv_mth_balance table. It is re-created in the next step. Enter the following command in SQL Server: truncate table fact_inv_mth_balance.

20 Set the INV_PROCESS_DAYS to 50000 to ensure that all data is reprocessed. Set the new INV_HISTORY_DATE parameter to today's date.

Note The daily load will take longer than normal, as all inventory transactions are being reprocessed to correct errors, and the fact_inv_mth_balance table is being repopulated.

- 21** Run the Daily load. If you had set the INV_PROCESS_DAYS value to a large number above, after the daily load completes, set the INV_PROCESS_DAYS to your normal value, such as 2, and resume normal processing.

Note These steps reload the fact_inv_transaction table with data as of the current day. Depending on the amount of process days, when the DAILY update resumes, it will reference all records from that fact table while trying to update the fact_inv_mth_balances table. Longer processing times are expected until the current date, minus the number of process days, exceeds the dss_update_time timestamp value from the reload. Once that time period has passed, the run times should improve substantially. Please be aware that you may experience at least one day of long processing time on the daily load.

Data Warehouse Tables Changed

The following sections list the Data Warehouse tables that have been changed, added, or deleted. Note that those in the Modified list may have been changed structurally (for example, columns added or deleted), or they may have had indexes added or deleted, or they may have had changes made to business display names or visibility of particular columns in the BI Portal, or any combination of these modifications.

Tables Modified in Existing Modules

Common Module Modified Tables

- | | | |
|----------------------------|----------------------------|-------------------------------|
| • dim_account | • dim_shipment_date | • stage_credit_term |
| • dim_account_currency | • dim_site | • stage_currency |
| • dim_approver | • dim_sub_account | • stage_customer |
| • dim_base_currency | • dim_supplier | • stage_customer_ar_list |
| • dim_buyer | • dim_transaction_currency | • stage_customer_bill |
| • dim_channel | • dim_transaction_type | • stage_customer_bill1 |
| • dim_corporate_currency | • dim_unit_of_measure | • stage_customer_dim_bill |
| • dim_cost_center | • load_ct_mstr | • stage_customer_dim_ship |
| • dim_credit_term | • load_ctd_det | • stage_customer_dim_sold |
| • dim_currency | • load_cu_mstr | • stage_customer_invoice_list |
| • dim_customer | • load_ee_Address | • stage_customer_master |
| • dim_customer_bill_to | • load_ee_BudgetGroup | • stage_customer_sales_list |
| • dim_customer_ship_to | • load_ee_Company | • stage_customer_ship |
| • dim_customer_sold_to | • load_ee_CostCentre | • stage_customer_ship1 |
| • dim_discount_cost_center | • load_ee_Creditor | • stage_customer_sold |
| • dim_discount_sub_account | • load_ee_Currency | • stage_customer_sold1 |
| • dim_due_date | • load_ee_Debtor | • stage_date_qad |
| • dim_effective_date | • load_ee_Division | • stage_domain |
| • dim_entered_date | • load_ee_DomainProperty | • stage_eam_entity |
| • dim_entity | • load_ee_Domains | • stage_ee_account |
| • dim_item | • load_ee_DomainSharedSet | • stage_ee_address |
| • dim_last_receipt_date | • load_ee_PaymentCondition | • stage_ee_ic_creditor |

- dim_location
- dim_order_date
- dim_payment_due_date
- dim_performance_date
- dim_project
- dim_receipt_date
- dim_requestor
- dim_required_date
- dim_service_type
- dim_ship_to_address
- load_ee_PaymentGroup
- load_ee_Project
- load_ee_SharedSet
- load_en_mstr
- load_pj_mstr
- load_tr_hist
- perm_transaction_hist
- permsup_transaction_hist
- stage_business_relation
- stage_cost_center
- stage_ee_ic_debtor
- stage_ee_project_status
- stage_entity
- stage_project
- stage_site
- stage_sub_account
- stage_supplier
- stage_tr_hist_list
- stage_unit_of_measure1

Financials Module Modified Tables

- dim_ap_transaction_type
- dim_ar_transaction_type
- dim_daybook
- dim_layer
- extract_ap_master
- extract_ap_voucher_master
- fact_ap_invoice_snapshot
- fact_ap_voucher
- fact_ap_voucher_history
- fact_ar_invoice
- fact_ar_invoice_history
- fact_ar_invoice_snapshot
- fact_ar_payment
- fact_gl_balance
- fact_gl_transaction
- load_ap_transaction_type
- load_ar_transaction_type
- load_ee_CompanySharedSet
- load_ee_Journal
- load_ee_Layer
- load_ee_Period
- load_vo_mstr
- perm_ap_master
- perm_ee_posting
- perm_ee_posting_line
- stage_ap_transaction_type
- stage_ar_transaction_type
- stage_ee_credit_invoice
- stage_ee_credit_invoice_move
- stage_ee_posting
- stage_gl_balance_close03
- stage_gl_balance_close04
- stage_gl_balance_ptd
- stage_gl_balance_qtd
- stage_gl_balance_ytd
- stage_gl_balance2
- stage_gl_transaction_budget
- stage_gl_transaction1
- stage_layer
- stage_se_ap_snap4
- work_ap_change_list1

Operations Module Modified Tables

- dim_po_line_status
- dim_po_line_type
- extract_inv_trans_hist_cons
- extract_inv_transaction_hist
- extract_op_transaction_hist
- extract_po_transaction_hist
- fact_inv_mth_balance
- fact_inv_transaction
- fact_op_transaction
- fact_po_order_history
- load_prh_hist
- load_wr_route
- perm_op_hist
- stage_inv_bal_item_list
- stage_inv_bal_mth_current
- stage_inv_bal_mth_trans1
- stage_inv_bal_mth_trans2
- stage_inv_bal_revalue2
- stage_inv_mth_balance
- stage_inv_mth_balance_hist_calc
- stage_inv_mth_balance1
- stage_inv_mth_end_cst
- stage_inv_mth_end_cst1
- stage_inv_transaction_con_cus3
- stage_inv_transaction_con_cus4
- stage_inv_transaction1
- stage_inv_trx_chg_mths
- stage_inv_trx_chg_mths1
- stage_op_tr_hist_rjct_wo
- stage_op_tr_hist_rjct_wo1
- stage_op_transaction2
- stage_po_order_hist
- stage_po_order_hist_initial
- stage_po_order_hist_initial1
- stage_po_order_hist_initial3
- stage_po_order_hist_initial4
- stage_po_order_hist_initial5
- stage_po_order_hist_initial6
- stage_po_order_hist_initial7
- stage_po_order_hist1
- stage_po_order_hist2
- stage_po_order_hist3
- stage_po_order_hist4
- stage_po_order_hist5
- stage_po_order_snap1

- stage_inv_bal_mth_current4
- stage_inv_bal_mth_date
- stage_inv_bal_mth_date1
- stage_inv_bal_mth_date2
- stage_inv_bal_mth_list
- stage_inv_bal_mth_trans
- stage_op_trans_scrapi
- stage_op_transaction
- stage_op_transaction_merge
- stage_op_transaction0
- stage_op_transaction1
- stage_po_order_snap3
- stage_po_order_snapshot
- work_inv_change_hist_list
- work_inv_change_list
- work_inv_change_list1

Order Management Module Modified Tables

- extract_om_transaction_hist
- fact_om_booking
- fact_om_shipment
- load_idh_hist
- load_ih_hist
- stage_om_ar_master_extract
- stage_om_booking
- stage_om_booking_previous
- stage_om_booking_transaction
- stage_om_booking_transaction1
- stage_om_booking1
- stage_om_invoice_line_list
- stage_om_invoice_line_ranged
- stage_om_invoice_transaction
- stage_om_invoice_transaction1
- stage_om_invoice1
- stage_om_order
- stage_om_order_inv_line
- stage_om_order_inv_line1
- stage_om_order_line
- stage_om_order_line_all
- stage_om_order_snap
- stage_om_order_snap1
- stage_om_order_snap3
- stage_om_order1
- stage_om_order2
- stage_om_order3
- stage_om_shipment1

Tables Added to Existing Modules

Common Module Added Tables

- dim_cal_int_end_date
- dim_cal_int_start_date
- dim_calendar_interval
- dim_statutory_currency
- load_ee_Address
- load_ee_AddressType
- load_ee_CompanyProperty
- load_ee_Contact
- load_ee_CorporateGroup
- load_ee_Country
- load_ee_County
- load_ee_CreditorType
- load_ee_DebtorCreditRating
- load_ee_DebtorType
- load_ee_Lng
- load_ee_PaymentConditionStaged
- load_ee_ProjectGroup
- load_ee_ProjectStatus
- load_ee_PurchaseType
- load_ee_Reason
- stage_business_relation
- stage_calendar_interval
- stage_calendar_interval_financial
- stage_calendar_interval_month
- stage_calendar_interval_week
- stage_date_qad5
- stage_eam_date_qad4
- stage_eam_domain
- stage_eam_entity
- stage_eam_site
- stage_ee_address
- stage_ee_budget_group
- stage_ee_contact
- stage_ee_cost_center
- stage_ee_credit_term
- stage_ee_currency
- stage_ee_customer
- stage_ee_customer_credit_rating
- stage_ee_customer_type
- stage_ee_entity
- stage_ee_payment_group
- stage_ee_project
- stage_ee_project_group
- stage_ee_project_status
- stage_ee_purchase_type
- stage_ee_sub_account
- stage_ee_supplier
- stage_ee_supplier_type
- stage_entity1
- stage_se_cost_center
- stage_se_credit_term
- stage_se_currency
- stage_se_customer
- stage_se_ee_cost_center
- stage_se_ee_entity
- stage_se_ee_sub_account
- stage_se_entity
- stage_se_project
- stage_se_sub_account
- stage_se_supplier

- load_ee_State
- load_ee_Usr
- stage_ee_invoice_status
- stage_site1

Financials Module Added Tables

- dim_accounting_period
- dim_ee_invoice_status
- load_ee_JournalGroup
- stage_accounting_period
- stage_ee_accounting_period

Operations Module Added Tables

- stage_inv_mth_end_cst
- stage_inv_mth_end_cst1
- stage_inv_trx_chg_mths
- stage_inv_trx_chg_mths1

Tables Deleted

No tables were deleted for the BI 3.7 release.

Upgrade to QAD BI 3.6 from QAD BI 3.5

BI Metadata Upgrade Procedure

Preparation

- 1 Plan for a couple of hours of down time.
- 2 Ensure none of the loads are currently in progress. **All jobs must be completed.**
- 3 Put all scheduled jobs On Hold until the upgrade is complete. Shut down scheduler.
- 4 Take a backup of the current warehouse database.
- 5 Before you begin, make note of the following:
 - a Note any custom DAILY_LOAD_JOBxxx jobs that appear after the final ROLLUP jobs (for example, DAILY_PO_ROLLUP, or DAILY_SALES_ROLLUP if you do not have the Purchase Orders module installed; if you have neither Sales nor Purchase Order processing, note any custom jobs that might be after the final SNAPSHOT jobs):

 - b Note any custom HIST_LOAD_JOBxxx jobs that appear after the standard ROLLUP or SNAPSHOT jobs:

Warning This upgrade will replace some load and stage tables, so if there are any customizations to these tables, ensure they are ready to be made again.

Upgrade Steps for QAD BI Metadata

- 1 Unzip the metadata for the Common module, plus any other modules you have installed, into a directory. Note that the Metadata files are now segregated by module, so you only install the modules you need.
 - All users need to run the Common metadata upgrade application.
 - If you process any of the Order Management (Sales) fact_om_* tables, also install the Order Management (OM) upgrade application.
 - If you process any of the SE and/or EE financial tables, also install the Financials upgrade application.
 - Finally, if you process Inventory and/or Purchase Order tables, install the Operations module upgrade application.

The Common upgrade should be performed first; the other upgrades can be done in any order.

- 2 Log in to Setup Administrator.
- 3 Choose Tools -> Start Logging. Select a name and location for your log file and Save.

4 Load the Common Module Upgrade Application as follows:

- a Set your application directory to the new metadata/common folder.
- b Load Application **upgradeCom v36** by right-clicking on it and choosing Install Application.
- c Select the ODBC DSN for your DWD instance from the drop-down list.
- d A dialog box will prompt you "Do you want to proceed now?"; click OK.

Note You can always Cancel here to review the list of new/modified objects; to continue with the load, right-click in the window and choose Proceed with Load Application.)

- e The next dialog box is for "Application Load Properties":
 - Select Dimension on the left - on the right, click on the box next to "Existing Dimension objects will be" and select Altered.
 - Select Dimension View on the left - on the right, click on the box next to "Existing Dimension View objects will be" and select Recreated.
 - Select Stage Table on the left - on the right, click on the box next to "Existing Stage table objects will be" and select Recreated.
 - Select Permanent Stage Table on the left - on the right, click on the box next to "Existing Permanent Stage table objects will be" and select Altered.
 - Select Fact Table on the left - on the right, click on the box next to "Existing Fact table objects will be" and select Altered.
 - Click OK.

5 The upgrade application output may have errors that some indexes could not be dropped; this is OK as they are new indexes that do not yet exist in the database (the upgrade script tries to drop each index prior to re-creating it; the drop will fail if the index is new). Any other errors in the upgrade output other than those index errors listed above should be noted and Support should be consulted prior to continuing.

6 Repeat for any other modules that need to be upgraded.

- a For Order Management, change the Application Directory to the metadata/order_management subdirectory, and install application **upgradeOM v36**.
- b For Operations, change the Application Directory to the metadata/operations subdirectory, and install application **upgradeOps v36**. Note that you may see an error here about the inability to create index dim_po_line_status_idx_A; this is corrected when the index is rebuilt after running the Upgrade jobs as described below.
- c For Financials (both EE and SE are in the same metadata package), change the Application Directory to the metadata/financials subdirectory, and install application **upgradeFin v36**.

7 The upgrade application installations will take some time as the load process alters/recreates various objects. When you have installed all pertinent upgrade applications, if any of the tables that were modified had included customizations, re-apply those customizations to your upgraded tables. Verify that all other customizations are not affected.

8 Start the Scheduler, but make sure all jobs are still suspended.

- 9 You must populate some new parameters and possibly modify some existing values, as follows:
- a Modify AR_PAYMENT_PROCESS_DAYS and COM_PROCESS_DAYS from the default values if desired:
 - AR_PAYMENT_PROCESS_DAYS: Used to find backdated payments in ar_mstr and ard_det tables. To save daily load time, not all records are loaded daily from ar_mstr and ard_det - only open records, plus closed records whose ar_date or ar_paid_date is between Today and Today minus AR_PAYMENT_PROCESS_DAYS. Set this to a value that will safely allow you to extract backdated payments; for example, if payments are sometimes entered with a date that is 3 months in the past, set this to approx. 90 days. However, if backdating is never more than 45 days, enter 45. Default value is 60.
 - COM_PROCESS_DAYS: Used to reduce load time by limiting the amount of data extracted from the pt_mstr, sct_det, and exr_rate tables to only those rows that have been modified between Today and Today minus COM_PROCESS_DAYS. Default value is 750.
 - b Financial module only: Modify AP_CHECK_PROCESS_DAYS from the default value if desired:
 - AP_CHECK_PROCESS_DAYS: Used to find backdated checks or vouchers in ap_mstr table. To save daily load time, not all records are loaded daily from ap_mstr - only open records, plus closed records such as checks with a back-dated void date or clear date, or vouchers with a back-dated paid date, between Today and Today minus AP_CHECK_PROCESS_DAYS. Set this to a value that will safely allow you to extract backdated rows; for example, if transactions are sometimes entered with a date that is 3 months in the past, set this to approx. 90 days. However, if backdating is never more than 45 days, enter 45. Default value is 60.
 - c Operations module only: Modify OP_LOAD_LOOKBACK_DAYS, OP_MOVING_AVG_WEEKS, OP_OEE_SNAPSHOT_FREQUENCY, and OP_PROCESS_DAYS from the default values if desired:
 - OP_LOAD_LOOKBACK_DAYS: Used to limit the amount of data extracted from wo_mstr, wod_det, and wr_route to just those open or closed between Today and Today minus OP_LOAD_LOOKBACK_DAYS. Default value is 20.
 - OP_MOVING_AVG_WEEKS: The number of weeks on which to base the Moving Average calculations. Default value is 13.
 - OP_OEE_SNAPSHOT_FREQUENCY: The frequency of snapshots in the OP OEE snapshot tables. Possible values are: W for Weekly, M for Monthly, B for both weekly and monthly. Default value is B for both.
 - OP_PROCESS_DAYS: The number of days of OP to extract and process from permanent staging tables. Default value is 5.
- 10 *****Very important***** Check the PO_ORDER_HISTORY_DATE and SALES_ORDER_HISTORY_DATE parameters. If the current date minus the PO_PROCESS_DAYS is newer than the PO_ORDER_HISTORY_DATE, or the current date minus the SALES_PROCESS_DAYS is newer than the SALES_ORDER_HISTORY_DATE, increase the process days value so that the current date minus that number of days will produce a date that is before the related ORDER_HISTORY_DATE (PO or SALES).

- 11 Delete all of your existing SET_CONNECTION_<source> jobs. (Note: do NOT delete the SET_CONNECTION_XXXXXXXX job; that is the template that is used to create new source-specific SET_CONNECTION jobs.)
- 12 Set INITIAL_JOB_SETUP_DATE to the current date.
- 13 Change INITIAL_JOB_SETUP_ENABLED to Y.
- 14 In the Scheduler, run the INITIAL_JOB_SETUP job.
- 15 The list of History and Daily jobs will have been re-written by the INITIAL_JOB_SETUP job. Add to the end of the DAILY_LOAD_JOBxxxx and HIST_LOAD_JOBxxxx load lists any jobs that occurred after the snapshot jobs (which you noted in the Preparation section).
- 16 Run the Upgrade jobs for each module you have installed:
 - a Run the UPGRADE_35_TO_36_COMMON job.
 - b If you installed the Financials module upgrade package, run the UPGRADE_35_TO_36_FINANCIALS job.
 - c If you installed the Operations module upgrade package, first set the Parameter QBI_924_PATCH_RUN to N, then run the UPGRADE_35_TO_36_OPERATIONS job. Note: the QBI_924_PATCH_RUN parameter is initially set to Y to guard against the accidental running of the patch upgrade task where it is not warranted. Set it to N just prior to running the Operations Upgrade job; after the job completes, the parameter is removed, ensuring the QBI-924 upgrade task is not allowed to run a second time.
 - d If you installed the Order Management (Sales) module upgrade package, run the UPGRADE_35_TO_36_ORDER_MANAGEMENT job.
- 17 If the upgrade affected any customized tables, re-apply your customizations (a list of new and changed tables can be found at the end of the Upgrade instructions.)
- 18 Resume normal processing.

To Add a New Source

If you want to add an additional data source, do the following:

- 1 Add the new source name to DAILY_LOAD_SOURCE_Sxx (where xx represents the next number in the sequence).
- 2 Add the new source name to HIST_LOAD_SOURCE_Sxx (where xx represents the next number in the sequence).
- 3 Add the new source name to INITIAL_JOB_SETUP_CONNECTION_xx (where xx represents the next number in the sequence).
- 4 Add a new INITIAL_JOB_SETUP_CONNECTION_xx_RUN (if it doesn't already exist) and set its value to N.
- 5 Add a new INITIAL_JOB_SETUP_CONNECTION_xx_TYPE (if it doesn't already exist) and set its value to either EE or SE depending on which type your financials system for that source is.

- 6 Make sure INITIAL_JOB_SETUP_DATE is set to the current date.
- 7 Set INITIAL_JOB_SETUP_ENABLED to Y.
- 8 Run INITIAL_JOB_SETUP.
- 9 Double-check the INITIAL_JOB_SETUP_CONNECTION_xx_RUN parameters to make sure that any sources that have been historically loaded have a Y next to their relative _xx_ number and that any new sources have an N.
- 10 Go to the Scheduler. Start the HIST_START job. Once it starts, double check to make sure it started running HIST_jobs only for your new data source.

BI Portal Upgrade Procedure

Preparation

- 1 Shut down Tomcat.
- 2 Back up the BI Portal database.

Upgrade Steps for QAD BI Portal

- 1 Copy the TOMCAT_HOME\webapps\your_bi_appname directory to TOMCAT_HOME\your_bi_appname.save35 so that you have a backup of your BI 3.5 web application.
- 2 Unzip the new BI 3.6 Portal data.zip file into TOMCAT_HOME\webapps\your_bi_appname, overwriting any existing files with the newer version from the data.zip file. Your configuration files will not be overwritten.
- 3 Start Tomcat.

Verification

- 1 Connect to your BI Portal. Click on the About link on the login screen and verify that the Version is **3.6** and the Build Number is **BI3-TRUNK-JOB1-1104**.

Note End users may need to clear their browser cache prior to re-connecting to the BI Portal in order to ensure they are running the latest version of the application.

Update Portal Model

- 1 Log into the Portal as an Administrator.
- 2 Go to the Administration menu and choose Model Administration. Click on Rebuild Model.
- 3 After the model has been rebuilt, if you have any custom data models, open each model and save it to ensure it picks up changes in the Master model.

Update Portal Documentation and Standard Objects

Update the metadata documentation:

- 1 Delete the existing TOMCAT_HOME\webapps\your_bi_appname\documentation\suite directory.
- 2 Unzip the latest qadbi-documentation.zip (found on the BI 3.6 Modules CD) into your TOMCAT_HOME\webapps\your_bi_appname\documentation directory.

To install the latest BI Portal standard objects, copy the files from the BI 3.6 Modules CD into your TOMCAT_HOME\webapps\your_bi_appname\export directory:

- portal\common\qad_common_dashboards.zip
- For each additional module you have installed, copy the portal\<module>\qad_<module>_reports.zip and qad_<module>_dashboards.zip files

Then, in the BI Portal:

- 1 Log in as an Administrator.
- 2 Go to the Administration -> Data Migration screen.
- 3 Click on the Import tab.
- 4 Check the "Update existing records" box.
- 5 Click the radio button in front of "File on server".
- 6 Enter in the box the file name qad_common_dashboards.zip and click on "Import".
- 7 Repeat for any other module reports and dashboards zip files that are required.

Data Warehouse Tables Changed

The following sections list the Data Warehouse tables that have been changed, added, or deleted. Note that those in the "Modified" list may have been changed structurally (for example, columns added or deleted), or they may have had indexes added or deleted, or they may have had changes made to business display names or visibility of particular columns in the BI Portal, or any combination of these modifications.

Tables Modified

Common Module Modified Tables

- | | | |
|------------------------|----------------------------|------------------------|
| • dim_account | • load_cu_mstr | • load_shop_cal |
| • dim_credit_term | • load_dom_mstr | • load_si_mstr |
| • dim_customer | • load_ee_BusinessRelation | • load_sp_mstr |
| • dim_customer_bill_to | • load_ee_Creditor | • load_SYSCOLUMNS |
| • dim_customer_ship_to | • load_ee_Currency | • load_tr_hist |
| • dim_customer_sold_to | • load_ee_Debtor | • load_tr_hist_history |
| • dim_date_qad | • load_ee_Domains | • load_tr_hist_list |

- dim_effective_date
- dim_item
- dim_sales_person
- dim_sales_person1
- dim_sales_person2
- dim_sales_person3
- dim_sales_person4
- load_ac_mstr
- load_ad_mstr
- load_ar_mstr
- load_budget_actual
- load_cc_mstr
- load_cm_mstr
- load_code_mstr
- load_consignment_type
- load_ct_mstr
- load_ctd_det
- load_ee_GL
- load_ee_SharedSet
- load_en_mstr
- load_exr_rate
- load_glc_cal
- load_hd_mstr
- load_icc_ctrl
- load_in_mstr
- load_loc_mstr
- load_pj_mstr
- load_pl_mstr
- load_pt_mstr
- load_ptp_det
- load_rnd_mstr
- load_sb_mstr
- load_sct_det
- load_service_type
- load_transaction_type
- load_um_mstr
- load_usr_mstr
- load_vd_mstr
- perm_item_master
- perm_shop_cal
- perm_transaction_hist
- stage_account
- stage_customer
- stage_customer_master
- stage_ee_account
- stage_ee_ic_debtor
- stage_item
- stage_item_costs
- stage_sales_person
- stage_shop_cal
- stage_transaction_hist

Financials Module Modified Tables

- extract_gl_transaction_hist
- fact_ap_invoice_snapshot
- fact_ap_voucher_history
- fact_ar_invoice_history
- fact_ar_invoice_snapshot
- fact_ee_ap_invoice_snapshot
- fact_ee_ar_invoice_snapshot
- fact_gl_transaction
- load_acd_det
- load_ap_mstr
- load_ap_transaction_type
- load_ar_transaction_type
- load_ard_det
- load_bg_mstr
- load_bdg_det
- load_ck_mstr
- load_ckd_det
- load_ee_ad_mstr
- load_ee_CInvoice
- load_ee_CInvoiceMovement
- load_ee_Company
- load_ee_CompanySharedSet
- load_ee_CostCenter
- load_ee_DInvoice
- load_ee_DInvoiceMovement
- load_ee_Division
- load_ee_DomainProperty
- load_ee_Journal
- load_ee_Layer
- load_ee_Period
- load_ee_Posting
- load_ee_PostingHist
- load_ee_PostingLine
- load_ee_Project
- load_ee_SysProperty
- load_gen_led_report
- load_gen_led_report_line
- load_gen_led_report_line_calcs
- load_gen_led_report_line_range
- load_gl_transaction_type
- load_gltr_hist
- load_gltr_hist_history
- load_gltr_hist_list
- load_vo_mstr
- stage_ee_gl_balance7
- stage_ee_gl_balance8
- stage_gl_balance1
- stage_gl_balance2
- stage_gl_transaction1

Operations Module Modified Tables

- dim_po_line_status
- dim_po_line_type
- dim_shipment_type
- extract_inv_trans_hist_cons
- extract_inv_transaction_hist
- load_inv_trans_type_sales
- load_inv_trans_type_supplier
- load_inv_transaction_type
- load_is_mstr
- load_ld_det
- stage_inv_transaction_con_cus3
- stage_inv_transaction_con_cus4
- stage_inv_transaction_con_cus5
- stage_inv_transaction_con_sup
- stage_inv_transaction_con_sup1

- fact_inv_transaction
- fact_po_order
- fact_po_order_history
- fact_po_order_performance
- fact_po_order_receipt
- load_inv_trans_type_cntrr
- load_inv_trans_type_cntrr
- load_inv_trans_type_cons
- load_inv_trans_type_cust
- load_inv_trans_type_git
- load_inv_trans_type_issued
- load_inv_trans_type_received
- load_inv_trans_type_recount
- load_mrp_det
- load_po_line_status
- load_po_line_type
- load_po_mstr
- load_pod_det
- load_prh_hist
- load_vep_mstr
- load_vepd_det
- stage_inv_bal_revalue1
- stage_inv_transaction
- stage_inv_transaction_con_cus
- stage_inv_transaction_con_cus1
- stage_inv_transaction_con_cus2
- stage_inv_transaction_con_sup2
- stage_inv_transaction_non_cons
- stage_inv_transaction1
- stage_inv_transaction2
- stage_inv_transaction3
- stage_inv_transaction4
- stage_po_order_hist
- stage_po_order_snap1
- stage_po_order_snap3
- stage_po_order_snapshot
- stage_po_receipt_perf
- work_po_perf_list

Order Management Module Modified Tables

- extract_om_invoice_line
- fact_om_order_history
- fact_om_order_performance
- fact_om_order_shipment
- load_cncu_mstr
- load_idh_hist
- load_ih_hist
- load_order_line_order_type
- load_so_mstr
- load_sod_det
- perm_om_invoice_line
- stage_om_invoice_line
- stage_om_invoice_order_line
- stage_om_invoice1
- stage_om_ord_perf
- stage_om_ord_perf_date1
- stage_om_ord_perf_full
- stage_om_ord_perf_ship
- stage_om_ord_perfl
- stage_om_order_inv_line
- stage_om_order_inv_line1
- stage_om_order_inv_line2
- stage_om_order_line
- stage_om_order_line_all
- stage_om_order1
- stage_om_order3
- stage_om_shipment
- stage_om_shipment_transaction

Tables Added

Common Module Added Tables

- dim_snapshot_period_type
- load_ee_BudgetGroup
- load_ee_CashGroup
- load_ee_DomainSharedSet
- load_ee_PaymentCondition
- load_ee_PaymentGroup
- load_ee_Unit
- load_snapshot_period_type
- stage_ee_account
- stage_se_account

Financials Module Added Tables

- work_gl_trans_change_list

Operations Module Added Tables

- dim_calendar_period
- dim_department
- dim_op_oeo_periods
- dim_production_line
- dim_release_date
- load_wr_route_hist
- perm_op_hist
- perm_work_center_master
- perm_work_order_detail
- perm_work_order_master
- stage_op_transaction
- stage_op_transaction_merge
- stage_op_transaction0
- stage_op_transaction1
- stage_op_transaction2

- | | | |
|-----------------------------------|------------------------------------|-------------------------------------|
| • dim_start_date | • perm_work_order_routing | • stage_op_work_center_period_hrs1 |
| • dim_work_center | • stage_calendar_period | • stage_op_work_order_detail_scrap |
| • dim_work_center_machine | • stage_exception_period_hrs | • stage_op_work_order_detail_scrap2 |
| • extract_op_oe_dates | • stage_op_equip_effect | • stage_op_work_order_lot |
| • extract_op_transaction_hist | • stage_op_equipment_effectiveness | • stage_op_work_order_lot_scrap |
| • fact_op_equipment_effectiveness | • stage_op_hist | • stage_prod_line_hrs |
| • fact_op_item_efficiency | • stage_op_item_efficiency | • stage_prod_line_period_hrs |
| • fact_op_transaction | • stage_op_item_efficiency1 | • stage_prod_line_period_hrs_ma |
| • fact_wo_order | • stage_op_item_efficiency2 | • stage_wo_mstr |
| • fact_wo_routing_transaction | • stage_op_item_moving_avg | • stage_wo_order |
| • load_cal_det | • stage_op_oe_periods | • stage_wo_order0 |
| • load_dpt_mstr | • stage_op_oe_periods_month | • stage_wo_routing_transaction |
| • load_ln_mstr | • stage_op_oe_periods_week | • stage_wo_routing_transaction0 |
| • load_op_hist | • stage_op_prod_line_period_hrs1 | • stage_wod_det |
| • load_shift_det | • stage_op_prod_line_period_hrs2 | • stage_work_center_period_hrs |
| • load_wc_mstr | • stage_op_prod_line_wc_period | • stage_work_center_period_hrs_ma |
| • load_wo_mstr | • stage_op_resource_avail_hrs | • stage_work_center_site |
| • load_wo_mstr_hist | • stage_op_resource_avail_keys | • stage_work_center_site_hrs |
| • load_wod_det | • stage_op_tr_hist_rjct_wo | • stage_wr_route |
| • load_wod_det_hist | • stage_op_tr_hist_rjct_wol | • work_op_wo_routing_costs |
| • load_wr_route | • stage_op_trans_scrapi | • work_op_wo_routing_costs_cum |

Order Management Module Added Tables

- dim_om_service_type

Tables Deleted

No tables were deleted for Release BI 3.6.

Upgrade to QAD BI 3.5 from QAD BI 3.4.1

BI Portal Upgrade Procedure

Preparation

- Shut down Tomcat.
- Back up the BI Portal database.

Upgrade Steps for QAD BI Portal

- 1 Copy the `TOMCAT_HOME\webapps\your_bi_appname` directory to `TOMCAT_HOME\your_bi_appname.save341` so that you have a backup of your BI 3.4.1 web application.
- 2 Unzip the new BI 3.5 Portal `data.zip` file into `TOMCAT_HOME\webapps\your_bi_appname`, overwriting any existing files with the newer version from the `data.zip` file. Your configuration files will not be overwritten.
- 3 Delete the file `TOMCAT_HOME\webapps\your_bi_appname\WEB-INF\lib\xstream-1.3.1.jar`.
- 4 Start Tomcat.

Note Users may need to clear their browser cache prior to reconnecting to the BI Portal to ensure that they are running the latest version of the application. Go to the Portal URL in the browser and click Refresh to make sure that the latest version of the application is loaded.

Verification

Connect to your BI Portal. Click on the About link on the login screen and verify that the Version is 3.5 and the Build Number is BI3-TRUNK-JOB1-1086.

BI Metadata Upgrade Procedure

Refer to the release notes for what has changed in this release.

Preparation

- 1 Plan for a couple of hours of down time.
- 2 Ensure that none of the loads are currently in progress. All jobs must be completed.
- 3 Put all scheduled jobs On Hold until the upgrade is complete. Shut down scheduler.
- 4 Take a backup of the current warehouse database.
- 5 Before you begin, make note of the following:
 - a Note any `DAILY_LOAD_SOURCE_Sxx` jobs that appear after the standard `_SNAPSHOT` tables.
 - b Note any `HIST_LOAD_SOURCE_Sxx` jobs that appear after the standard `_SNAPSHOT` tables.

Important This upgrade will replace some load and stage tables, so if there are any customizations to these tables, ensure that they are ready to be made again.

Upgrade Steps for QAD BI Metadata

- 1 Due to significant improvements in Purchase Order processing, the `fact_po_receipt`, `fact_po_order_history`, and `fact_po_order_performance` tables need to be reloaded completely. Corrections made to the SE Financials module require the re-processing of SE AP and AR

data. The fact_om_order_history table must also be reloaded. Truncate these tables now; after the upgrade application is loaded, jobs are run to reload these tables using the new processing. In SQL Server, truncate the tables as follows:

- truncate table fact_po_receipt
- truncate table fact_po_order_performance
- truncate table fact_om_order_history
- truncate table fact_ap_voucher_history
- truncate table fact_po_order_history

- 2 Log in to Setup Administrator.
- 3 Set your application directory to the new metadata folder. Load Application 35-up (new/modified tables, procedures, parameters, and jobs) by right-clicking on it and choosing Install.
 - a A dialog box will prompt you to proceed; click OK.
 - b The next dialog box is for Application Load Properties:
 - Select Dimension on the left; on the right, click on the box next to Existing Dimension objects will be and select Altered.
 - Select Dimension View on the left; on the right, click on the box next to Existing Dimension View objects will be and select Recreated.
 - Select Stage Table on the left; on the right, click on the box next to Existing Stage table objects will be and select Recreated.
 - Select Permanent Stage Table on the left; on the right, click on the box next to Existing Permanent Stage table objects will be and select Altered.
 - Select Fact Table on the left; on the right, click on the box next to Existing Fact table objects will be and select Altered.
 - Click OK.
- 4 The upgrade application installation will take some time as the load process alters/recreates various objects. Once the process completes, make any necessary customizations to your load tables. Verify that all other customizations are not affected.
- 5 The upgrade application output may have errors that some indexes could not be dropped; this is OK as they are new indexes that do not yet exist in the database (the upgrade script tries to drop each index prior to re-creating it; the drop will fail if the index is new). Any other errors in the upgrade output other than those index errors listed above should be noted; consult QAD Support before continuing.
- 6 Start the Scheduler, but make sure all jobs are still suspended.
- 7 You must populate some new parameters and possibly modify some existing values, as follows:
 - a Modify PO_RET_QTY_TOLERANCE, PO_PERCENT_QTY_UNDER, PO_PERCENT_QTY_OVER from the default values if desired:

- **PO_PERCENT_QTY_OVER**: Percentage of Receipt Quantity over Order Quantity for a Purchase Order to still be considered In Full. Default value is 10%. In other words, the actual quantity received can be up to 10% greater than the amount ordered and the order line will still be considered In Full. If the quantity received is greater than the quantity ordered plus 10%, the order line is not considered In Full.
- **PO_PERCENT_QTY_UNDER**: Percentage of Receipt Quantity under Order Quantity for a Purchase Order to still be considered In Full. Default value is 10%. In other words, the actual quantity received can be up to 10% less than the amount ordered and the order line will still be considered In Full. If the quantity received is less than the quantity ordered minus 10%, the order line is not considered In Full.
- **PO_RET_QTY_TOLERANCE**: This is the Percentage of Returns Quantity used for calculating the Defect Rate. Default value is 15%. In other words, if the return quantity is greater than 15% of the order quantity, the order line is considered a defect.

b Modify **SALES_SNAPSHOT_REPLACE_FLAG** and **SALES_SNAPSHOT_FREQUENCY** from the default values if desired:

- **SALES_SNAPSHOT_REPLACE_FLAG**: Y to replace the snapshot for recalculated periods when the **SALES_ROLLUP** job is run; N to just add new records and not update previously calculated records. Default value is N.
- **SALES_SNAPSHOT_FREQUENCY**: Set to M for monthly snapshots, W for weekly snapshots, or B for both monthly and weekly. Default value is M.

Important Check the **PO_ORDER_HISTORY_DATE** and **SALES_ORDER_HISTORY_DATE** parameters. If the current date minus the **PO_PROCESS_DAYS** is newer than the **PO_ORDER_HISTORY_DATE**, or the current date minus the **SALES_PROCESS_DAYS** is newer than the **SALES_ORDER_HISTORY_DATE**, increase the process days value so that the current date minus that number of days will produce a date that is before the related **ORDER_HISTORY_DATE** (PO or SALES).

8 Set **INITIAL_JOB_SETUP_DATE** to the current date.

9 Change **INITIAL_JOB_SETUP_ENABLED** to Y.

10 Check the following jobs for customizations. Note any custom changes that have been made; reapply them to the new job templates. Then delete the following jobs for each existing source (Note: do NOT delete the job name ending in **_XXXXXXXX**; that is the template from which the source-specific jobs is created):

- **DAILY_COMMON_PROCESS_sourcename**
- **DAILY_PERM_sourcename** (CAUTION: do NOT delete the job named **DAILY_PERM_EXTRACT**)
- **DAILY_PO_PROCESS_sourcename**
- **DAILY_SALES_PROCESS_sourcename**
- **HIST_COMMON_PROCESS_sourcename**
- **HIST_PERM_sourcename**
- **HIST_PO_PROCESS_sourcename**
- **HIST_SALES_PROCESS_sourcename**

11 In the Scheduler, run the **INITIAL_JOB_SETUP** job.

- 12 The list of History and Daily jobs will have been rewritten by the INITIAL_JOB_SETUP job. Make sure any customizations that were made to the new job templates are now in the source-specific copies of those jobs (or reapply any customizations as necessary). Also add to the end of the DAILY_ and HIST_ load lists any jobs that occurred after the snapshot jobs (which you noted in the Preparation section).
- 13 This might be a good time to review the Parameters and their definitions in the BI 3.5 Installation Guide. Check your parameter settings; you might want to consider changing some, such as various PROCESS_DAYS and SNAPSHOT_NUM_DAYS values, to make sure you are not reloading extra data unnecessarily each day. The module PROCESS_DAYS are typically set to 10 by default; SNAPSHOT_NUM_DAYS are set to 31 by default.

Fixing Existing Fact Tables

Now run the jobs to reload the fact tables you had truncated earlier. Note that you will need to run these for each source. Depending on the amount of history data in your system, these jobs may take a while to run.

- 1 Set parameter JOB_CHAINING_ENABLED to N.
- 2 Run the job SET_CONNECTION_XXXXXX, where XXXXXX = the source connection name.
- 3 Run HIST_PERM_EXTRACT.
- 4 Run the job HISTORY_RELOAD_OM_35_UPGRADE.
- 5 Run the job HISTORY_RELOAD_PO_35_UPGRADE.
- 6 Run the job UPGRADE_FACT_PO_RECEIPT_341_TO_35.
- 7 (SE sources only) Run the job RELOAD_SE_AP_AR_FROM_PERM.
- 8 Repeat steps 2-7 for each source connection (if you have multiple sources).
- 9 Run the job HISTORY_SALES_ROLLUP.
- 10 Run the job HISTORY_PO_ROLLUP.
- 11 When upgrade jobs have been run for all sources, make sure to set the parameter JOB_CHAINING_ENABLED back to Y.

Re-create Purchasing Cube

If you use the Purchasing cube, you must delete the measure group that referenced the fact_po_booking table:

- 1 Delete the cube in the Analysis Server.
- 2 In the DWD, right-click on the OLAP Purchasing cube and select Display Measure Groups.
- 3 Right-click on the fact_po_booking measure group and choose Delete Measure Group. Click OK to answer the verification questions about deleting the associated measures.
- 4 Right-click on the OLAP Purchasing cube and choose Create (Alter) Cube.

5 Process the cube.

Upgrade should be complete at this point. Resume normal processing.

To Add a new Source

If you want to add an additional data source, do the following:

- 1 Add the new source name to DAILY_LOAD_SOURCE_Sxx (where xx represents the next number in the sequence).
- 2 Add the new source name to HIST_LOAD_SOURCE_Sxx (where xx represents the next number in the sequence).
- 3 Add the new source name to INITIAL_JOB_SETUP_CONNECTION_xx (where xx represents the next number in the sequence).
- 4 Add a new INITIAL_JOB_SETUP_CONNECTION_xx_RUN (if it does not already exist) and set its value to N.
- 5 Add a new INITIAL_JOB_SETUP_CONNECTION_xx_TYPE (if it does not already exist) and set its value to either EE or SE depending on which type your financials system for that source is.
- 6 Make sure INITIAL_JOB_SETUP_DATE is set to the current date.
- 7 Double-check the INITIAL_JOB_SETUP_CONNECTION_xx_RUN parameters to make sure that any sources that have been historically loaded have a Y next to their relative _xx_ number and that any new sources have an N.
- 8 Go to the Scheduler. Start the HIST_START job. Once it starts, double-check to make sure it started running HIST_jobs only for your new data source.

Update Portal Model

Log in to the Portal as an Administrator. Go to the Administration menu and choose Model Administration. Click on Rebuild Model.

After the model has been rebuilt, if you have any custom data models, open each model and save it to ensure that it picks up changes in the Master model.

Update Portal Documentation and Standard Objects

Update the metadata documentation:

- Delete the existing
TOMCAT_HOME\webapps\your_bi_appname\documentation\suite directory.
- Unzip the latest qadbi-documentation.zip (found on the BI 3.5 Modules CD) into your
TOMCAT_HOME\webapps\your_bi_appname\documentation directory.

To install the latest BI Portal standard objects, copy these files from the BI 3.5 Modules CD into your TOMCAT_HOME\webapps\your_bi_appname\export directory:

- BI35_deprecated_items.zip

- qad_standard_items.zip

Then, in the BI Portal:

- 1 Log in as an Administrator.
- 2 Go to the Administration -> Data Migration screen.
- 3 Click on the Import tab.
- 4 Select the Update existing records box.
- 5 Click the radio button in front of File on server.
- 6 Enter in the box the file name BI35_deprecated_items.zip and click on Import.
- 7 Repeat for the qad_standard_items.zip file.

Sizing Guidelines

This document covers sizing for the server-side BI components. Topics include:

Business Intelligence Deployment Options and Terminology 123

CPU Sizing 126

RAM Sizing 127

Disk and IO Subsystem 128

Network Considerations 130

Example Sizing 131

Reference: Estimation of Initial Database Size 132

Reference: Network Sizing Details 135

Introduction

This document is a provisional and interim sizing guide for the appropriate sizing of hardware environments ready for the deployment of the QAD Business Intelligence product—QAD BI version 3.6 and above.

- **Provisional.** A formal sizing guide is normally produced after a detailed capacity planning analysis involving a series of formal internal sizing/scalability/performance testing. The sizing estimates in this document are provisional in the sense that they have been produced in the absence of a formal capacity planning process.
- **Interim.** This sizing guide can be considered an interim sizing guide until a formal capacity planning testing process can be carried out for QAD BI.
- **Practical Sizing.** However, at the time of publication, the sizing estimates in this guide are the most accurate available to QAD, and this document should be used for practically sizing BI deployments.
- **Caveats to BI Sizing.** Although the sizing estimates given here are the best available, the absence of a formal capacity planning process means that certain estimates in this document may need to be updated as more accurate information becomes available and hence, sizing should be carried out bearing in mind all the caveats listed under the following topic.

This document covers sizing for the server-side BI components. It does not comment on client-side BI Portal (browser-based) sizing requirements.

Important Restrictions and Caveats on BI Sizing

All restrictions and caveats related to sizing are covered in this section.

Note In the absence of a formal capacity planning testing process for QAD BI, this document presents best-estimate sizing guidelines for QAD BI.

At the time of publication, these estimates are the most accurate available, and should serve as an appropriate guideline for sizing QAD BI, taking the following caveats into account.

The following restrictions and caveats need to be borne in mind when estimating sizing from this document:

- **Caveat 1:** This product has not been sized using formal internal sizing/scalability/performance testing. It should therefore be understood that all sizing guidelines in this document are based on valid but incomplete information so guidelines are updated as more sizing tests and research are carried out.
- **Caveat 2:** All effort has been made to make these sizing estimates as accurate and reliable as possible, using standard capacity planning and sizing planning best-practice. However, as we are sizing based on incomplete information, some estimates may not be 100% accurate. Keep this in mind.

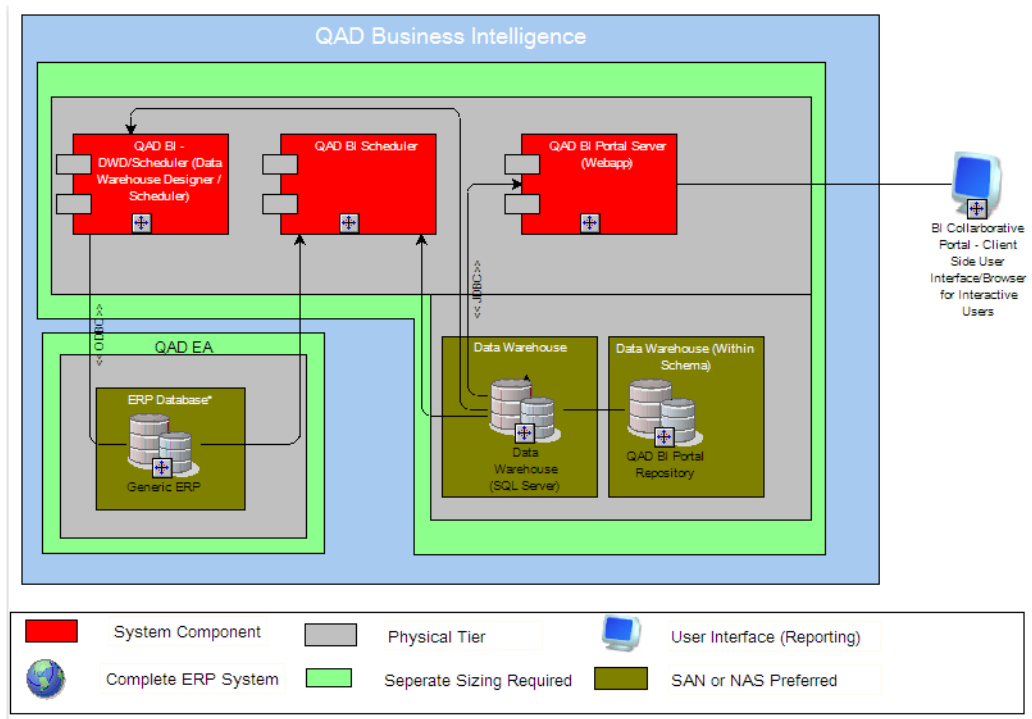
In summary, the current sizing estimates should be considered best-estimate sizing guidelines, safe to use in the field, but subject to future updates/correction and also subject to the caveats listed above.

Business Intelligence Deployment Options and Terminology

This section covers BI terminology and BI deployment options.

BI Terminology

A schematic picture of BI is provided in the following figure:



- **Data Warehouse Designer (DWD).** The application that allows viewing and modification of metadata that describes and builds the data warehouse.
- **Data Warehouse Designer Scheduler (DWD Scheduler).** The component of the DWD responsible for loading data. The DWD Scheduler schedules and runs jobs to extract data from remote ERP systems to build the data warehouse.
- **DWD/Scheduler.** In this document, DWD/Scheduler is used to refer to all the technical components composing the DWD and the DWD Scheduler.
- **QAD BI Collaborative Portal (Client-Side Browser).** This is the remote client-side user interface to QAD BI. It is a flash-based interface run on a remote Internet browser and provides for interactive dashboards, reports, collaboration and administration.
- **QAD BI Portal Server (Server-Side Webapp).** This is the server-side component of the QAD BI Portal. It is a java-based Web application (webapp), which receives requests from the client-side BI Portal. The webapp communicates requests as queries to SQL Server and coordinates the reply to the client-side browser.

- **Single Tier.** Single-tier deployment refers to the fact that all the primary QAD BI components —DWD/Scheduler, BI Portal (Webapp), and SQL-Server—are deployed on one single server machine. In this deployment model and all other deployment models, it is preferred and recommended that the BI data warehouse itself is placed on a separate SAN/NAS device.
- **Two Tier.** Two-tier deployment refers to having the QAD BI Portal (server-side webapp) on a separate machine (server) from the DWD/Scheduler and SQL Server. The DWD/Scheduler and SQL-Server remain on a the primary server machine.

BI Deployment Options

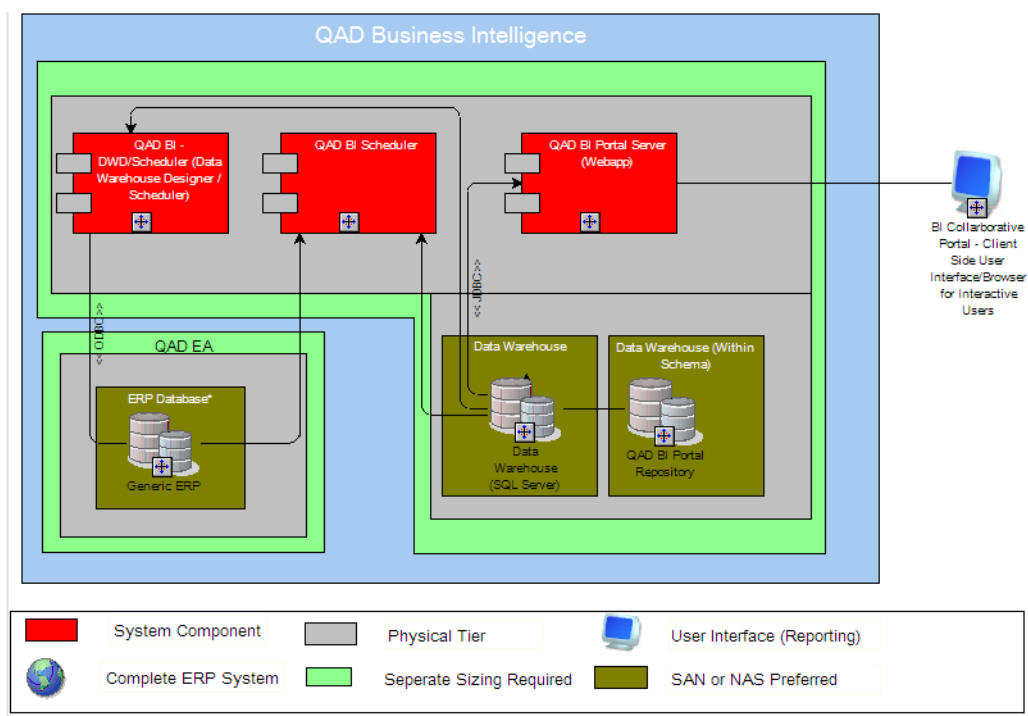
Business Intelligence involves several different components and processes. Before determining a final sizing estimate, the components and processes must be sized separately and appropriately, depending on the deployment option. There are two primary deployment options available:

- **Single tier.** All BI components on one server
- **Two tier.** The QAD BI Portal Server on a separate physical/virtual server to all other BI components

Using this breakdown of BI technical components allows you to size appropriately for either a single-tier or two-tier deployment.

Deployment Option 1 – Single Tier

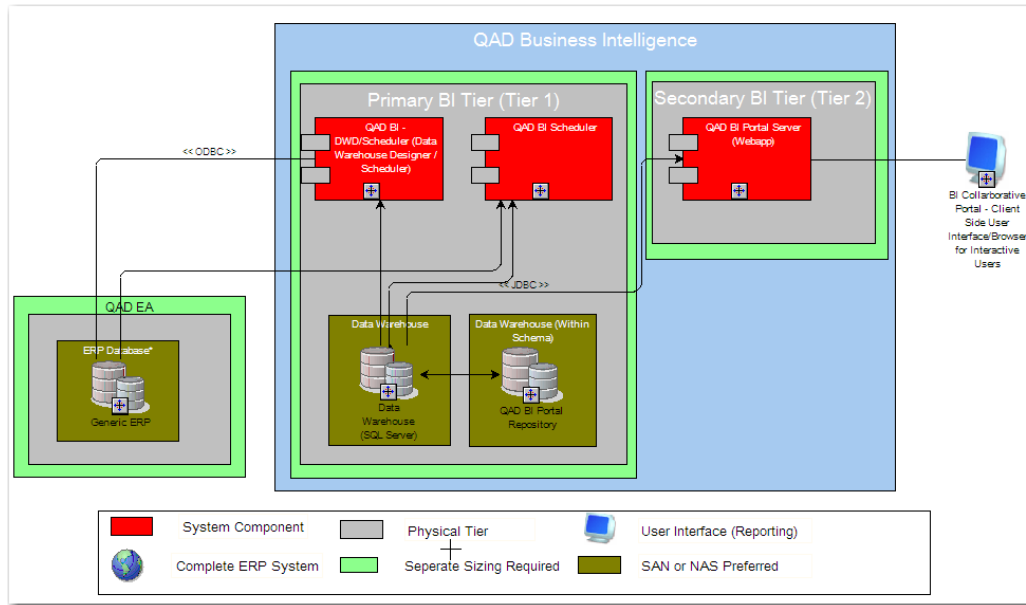
A single-tier BI deployment has all BI components on the same server. This is illustrated in “BI Terminology” on page 123 above and is repeated schematically here:



Deployment Option 2 – Two Tier

The two-tier BI deployment has the BI Portal webapp on a separate server from all other BI components. The deployment option is illustrated in the following figure.

Note The BI Portal has a much smaller resource requirement demand than the other BI components, so the size of a BI Portal server is smaller than the BI DWD/Scheduler server. This distinction is made throughout this document.



Sizing Terminology Used Throughout this Sizing Guide

Sizing Metrics

Sizing across the following four metrics is covered separately in this guide:

- CPU
- RAM
- IO Subsystem & Disk
- Network / WAN

Sizing Methodology for QAD BI

Note This document comments on sizing for the QAD BI server-side components only.

Two different workload types place different demands on BI components. Also, there are two deployment options for QAD BI. Because of this, each type of workload must be sized for separately. The methodology used for sizing BI divides each metric into the following areas:

- **Sizing for DWD/Scheduler Workload (Batch Jobs).** This refers to the workload placed on the server as a result of the DWD/Scheduler automated jobs. This workload needs to be sized for separately and is of a batch-type.

- **Sizing for Concurrent User Workload (Interactive Portal Users).** This is the workload placed on the DW server as a result of the concurrent interactive users using the BI Portal. This workload involves concurrent active end-users effectively running queries against the BI data warehouse through their remote BI Portal-client user interface. This workload can place significant read demand against the data warehouse and needs to be sized for separately.
- **Sizing for BI Portal Webapp (Server Side).** Because the BI Portal webapp can be placed on a separate tier from the rest of BI in the two-tier deployment model, record sizing requirements for this component separately for each relevant sizing metric. If sizing for a single-tier, combine the sizing for the BI Portal webapp with the other sizing metrics.

CPU Sizing

Note This product has not been sized using formal performance and sizing procedures. All sizing guidelines in this document are based on incomplete information and guidelines are updated as more sizing tests and research are carried out.

CPU Sizing for DWD/Scheduler Component

Based on current information, the DWD/Scheduler processing is mostly single-threaded, meaning that it rarely requires more than one CPU core.

Hence, current guidelines suggest that for DWD/Scheduler CPU sizing, at most two CPU cores are required.

Sizing Metric	Recommendation	Recommendation 2
CPU – DWD Scheduler	2 CPU Cores	Use the fastest CPU cores available. Daily/Nightly DWD Scheduler jobs are CPU-bound and processing time benefits greatly from <i>faster</i> CPU cores (not from <i>more</i> CPU cores).

CPU Sizing for BI Portal Webapp (Server Side)

Note that this is CPU sizing just for the server-side BI Portal java-based webapp. In this case, QAD estimates that a single CPU core is sufficient to run the BI Portal webapp.

Sizing Metric	Recommendation	Recommendation 2
CPU – BI Portal Component (java webapp)	1 CPU Core	-

CPU Sizing for Concurrent User Workload (Interactive Portal Users)

“Interactive portal users” use their remote BI Portal-client user interface to execute queries against the primary DW server. “Concurrent User Workload” therefore refers to the load placed on the SQL-Server tier by the remote concurrent BI Portal users.

Note that multiple concurrent BI Portal users executing requests can place a large load on the SQL-Server data server. Hence, in this section, we are sizing for the SQL-Server tier, independent of a one-tier or two-tier deployment.

Sizing Metric	Concurrent User Count	Recommendation
CPU – Concurrent BI Portal User Load	1-10	4 CPU cores
	10-20	4-6 CPU Cores
	20-40	6-8 CPU Cores
	40-60	8 CPU Cores

RAM Sizing

RAM Sizing for DWD/Scheduler Component

The amount of RAM needed is determined by:

- The size of the data warehouse
- The amount of data being manipulated by the DWD Scheduler and SQL-Server
- The complexity and degree of customizations within the BI modules

RAM Sizing for BI Portal Webapp (Server-Side)

Sizing Metric	RAM for OS (GB)	RAM for SQL-Server	Total Server RAM ¹
Small Load	4	6+	10+
Medium Load	4	8+	12+
Large Load	4	12+	16+

1. Total Server RAM assumes RAM sizing for the Windows Operating System and BI DWD Scheduler/SQL-Server only. If other components are on the server, you must size for these separately and allocate an appropriate amount of extra RAM.

Sizing Metric	Recommendation	Comment
RAM – BI Portal Technical Component	2 GB	Allocate 2GB to the BI Portal webapp. On 64-bit systems and for very large workloads (concurrent user counts), over 2 GB can be allocated.

RAM Sizing for Concurrent Users (Interactive Portal Users)

A separate server-side RAM-sizing for interactive portal users is not required in this case, because allowance has been made in the preceding subtopics for this RAM sizing. To explain:

Concurrent interactive portal user workload consists of multiple remote-client BI Portal users placing a load on the server-side BI Portal webapp and SQL-Server. Hence, “Sizing for Concurrent Users” sizes for the workload placed on the server-side components by the remote interactive portal users. Allowance for this had been made in the preceding DWD/Scheduler and BI Portal Webapp sizing discussions.

Disk and IO Subsystem

Note on Importance of Appropriate Disk and IO Subsystem Sizing

Because of the heavy IO demands of each aspect of the BI product, disk and IO subsystem capacity and sizing are vitally important for product performance and scalability.

IO Subsystem Sizing

The speed and throughput of the IO subsystem is critical to the overall performance of the QAD BI application. Note that this sizing refers to the data warehouse IO subsystem.

Sizing Metric	Recommendation (IO Subsystem)
Disk Space	Predicting disk sizing requirements for QAD BI is complex. See “Key IO Subsystem Sizing Notes” below and “Reference: Estimation of Initial Database Size” on page 132 for details. The following guidelines should be followed. Recommendations 1. <u>Minimum</u> - A minimum of 500 GB disk space is recommended, but see points 5 and 6. 2. <u>Ideal</u> - Ideally 1 TB of disk space is recommended, but see points 5 and 6. General Guidelines 3. <u>Allowance for database growth</u> - To allow for BI database growth, 2-3 times the size of the original BI database is recommended. See “Allowance for Future Database Growth” on page 134 for the exact details. 4. <u>Allowance for disk expansion</u> – If appropriate and to allow for future project growth, it is recommended to allow for disk expansion capabilities in the SAN/NAS device. 5. <u>Estimation</u> - “Reference: Estimation of Initial Database Size” on page 132 provides an approximate method of estimating the initial BI database size from source ERP tables. This method can be used along with point 3 above. To provide an estimate of the initial disk space requirements for a new implementation, multiply “estimated” database size by a factor of 2-3 to allow for growth. Final Disk Sizing – Specific Guidelines 6. If the estimate from point 5 returns over 500 GB or over 1 TB of disk space requirements, then obviously size accordingly.
SAN/NAS?	It is preferable to use external disk arrays or SAN/NAS devices where possible. Internal disk drives should be configured to hold the operating system only.
RAID Level	RAID 5 or RAID 10 preferred.
Physical Disks	Fastest disks available: 15 k rpm SAS (or fiber channel) disks preferred. Note 1: <u>Due to the larger IO demands of BI systems, a <i>minimum</i> of 6 disks is recommended.</u> Double the amount of disks when using mirroring. Note 2: More disks are preferred, as this will significantly help performance and IO rates.
IO Controller	Ensure IO controller throughput does not limit transfer rates to/from disks. Where possible spread the IO across as many controllers as possible.

Key IO Subsystem Sizing Notes

- 1 In the absence of any estimates, 500 GB is the minimum recommendation and 1 TB is ideal. The final estimate of disk space needs to be based on best-estimates. See points 2 and 3 below.
- 2 Use “Reference: Estimation of Initial Database Size” on page 132 to estimate order-of-magnitude sizing for new BI databases.

- 3 To allow for BI database growth, size the total disk space for 2-5 times the size of the BI database. See “Allowance for Future Database Growth” on page 134 for details.
- 4 Fast disks – 15k SAS disks.
- 5 RAID Level spread across multiple disks, recommend 6 disks minimum.
- 6 Need the associated IO controllers to be capable of servicing maximum throughput from each disk or combination of disks in RAID configuration. Ensure the IO controllers used do not limit transfer rates per disk below the maximum rated physical-disk and RAID transfer rates, as this can act as a bottleneck at high IO demand rates.

Notes on Database Disk Sizing

Factors Affecting Disk Sizing Estimates

Currently there is no detailed information on how to scale the expected size of the BI data warehouse from a source ERP system. The reason for this is that the size of the BI data warehouse depends on a variety of factors:

- The number of modules implemented
- The size of the source ERP tables transformed
- The number of ERP sources
- The volume of daily and historical ERP data
- The degree of customization of the BI modules

Because these factors are difficult to predict beforehand, we currently give guidelines on expected data warehouse size based on existing customer data.

Estimation of Initial Database Size

If required, refer to “Reference: Estimation of Initial Database Size” on page 132 for an approximate method of estimating the initial BI database size. Note that this estimate is very approximate in nature, and should be used only as a means of getting an order-of-magnitude estimate on database size.

Allowance for Database Growth

Depending on the daily transaction volumes, the number of BI modules, and the customizations, the BI database grows on a daily basis; this depends on the daily scheduled load jobs. Allowance needs to be made up-front for this in terms of disk capacity.

The recommendation is to have a disk capacity of 2-3 times the estimated size of the BI database. The rule of thumb is that if the initial BI DW is based on converting 3 years of historical data and you assume the same rate of transactions are passing through the source ERP database, then expect the BI DW itself to grow by a similar volume to the initial size of the DW over the coming 3 years.

Also, daily or weekly monitoring of the BI database size provides ongoing measurement of growth rates, and adequate storage should be provided to manage this growth.

Allowance for Database Backup

Ensure a recommendation for a tape-drive backup or other appropriate backup media.

Network Considerations

Network sizing involves taking the following considerations into account. For network terminology, see “Reference: Network Sizing Details” on page 135:

- Bandwidth requirements of the product
- Effect of latency and packet loss on performance and responsiveness

There are different types of network communication demand within QAD BI, and each area requires separate consideration from a sizing point of view. To this end we divide network sizing into the following areas:

- 1 Network considerations between the ERP database and BI database
- 2 Network considerations between the remote client and BI Portal
- 3 WAN Considerations for both 1 and 2; for example, the impact of placing a WAN between the ERP database and the BI database server, and the impact of placing a WAN between a remote client and the BI Portal

See “Reference: Network Sizing Details” on page 135 for details of estimating the impact of network considerations in each of these areas.

Network Recommendations

Please be aware of both sizing notes from this table and review “Reference: Network Sizing Details” on page 135 before finalizing network recommendations for the customer.

Network Bandwidth	LAN	WAN (Effect of Latency)	Network Card	Comment
ERP database to BI database	10+ MB/s (80 Mbps)	4 MB/s ¹ (32 Mbps)	Ensure at least 1 Gbps card	The WAN figure is a guideline figure and depends on the actual latency involved (see “WAN Considerations” on page 137 for details).
Backup of BI database	1 Gbps	N/A	Ensure at least 1 Gbps card	
Remote Client to BI Portal	80-140 kB/s (640 – 1120 kbps) per user	50+ kB/s ² (400+ kbps) per user		Ensure these rates are <i>minimum bandwidth per user</i> for each remote client to BI Portal user

1. Increasing latency (on WANs) has a very significant effect on slowing down the transfer rate between the ERP database and the BI database. Very careful measurement should be made of the network quality (latency and packet loss, bandwidth) of any customer proposing a WAN-based separation between the ERP database and the BI database servers (see “Reference: Network Sizing Details” on page 135 for details)
2. Testing has shown that client response times are expected to slow down by up to 100% across a WAN with a latency of 300-400 ms. Responsiveness slows down across Wide Area Networks.

Example Sizing

This section provides an example of how to size a BI system.

Size a BI system with the following requirements:

- 20 expected concurrent BI Portal users
- A 50 GB ERP source database
- Initially, three BI Modules required

This example shows a single-tier deployment in which one server holds all BI components.

	Sizing Metric:	Recommendation	Note
A	CPU – DWD/Scheduler	2 cores	
B	CPU – Concurrent User Workload (Remote Portal Users)	20 concurrent users = 6 cores	
C	CPU – BI Portal Webapp	1 core	
D	CPU Total	6 + 1 = 7 cores	Max. (A,B) + C – Note that this is a <i>minimum requirement</i>
E	RAM - DWD/Scheduler	Assume “Medium” Load – 8+ GB	
F	RAM – BI Portal Webapp	2 GB	
G	RAM – Concurrent User Workload (Remote Portal Users)	Not relevant (as per “RAM Sizing for Concurrent Users (Interactive Portal Users)” on page 127)	
H	RAM - Total	8+ + 2 + 4 GB = 14 GB	As per “RAM Sizing for DWD/Scheduler Component” on page 127, the 4 GB extra is for the OS
I	Disk Sizing	Initial approximate sizing needs to be carried out to create an <i>approximate</i> estimate of the initial DW size based on ERP table data. See “Reference: Estimation of Initial Database Size”.	As a note, 500 GB is a recommended minimum but all new implementations should obtain <i>disk sizing estimates</i> using the referenced guidelines.
J	IO Subsystem	15 k rpm SAS disks RAID 10 on a SAN/NAS device Minimum of 6 disks (12 disks if using mirroring).	Again, depending on the estimated DW size, increase the number of disks in the RAID array.
K	Network Requirements	ERP-to-BI Server: Minimum of 3 MBps (Mega-bytes per second) bandwidth Client-to-BI Portal: Minimum of 100 kbps (kilo-bytes per second) bandwidth per user	Ensure 1 Gbps network card minimum on DW server. For two-tier deployment also ensure at least a 1 Gbps network connections between the DW server and the BI Portal webapp server.

Reference: Estimation of Initial Database Size

If the BI data warehouse does not exist, the following can be used to provide a very approximate guideline as to the approximate initial expected BI database size.

Dependence of Initial Database Size on BI Modules

Note that BI v3.6+ is fully “modularized,” with three primary modules:

- Financials
- Order Management
- Operations

Sizing for the resultant BI Data Warehouse depends on many factors, especially the table size within the source ERP database. As different BI modules use only restricted source ERP tables, the final size of the BI Data Warehouse sensitively depends on the modules and sub-modules used. This section describes how to create an estimate of the initial BI DW size based on the BI modules used.

Estimation of Initial BI Data Warehouse Size

The estimation of the initial data warehouse size involves a determination of the size of key ERP tables, and a subsequent calculation, which depends on the modularization, to convert these sizes into an estimated data warehouse size. Follow the calculation and description below to estimate the initial data warehouse size.

Note There is no guarantee that this estimate will give an accurate size for the initial data warehouse size, it is provided as an approximate method of generating an order-of-magnitude estimate on the initial data warehouse size.

Note SQL-Server uses a large amount of extra disk space for temporary manipulation of transformational data and allowance should also be made for this. The estimate below only considers an approximation for the BI database, excluding requirements for SQL-Server temporary files.

Determine the size of all the following tables in the source QAD ERP database and use these in combination with the “Size Factor” to calculate an estimate of the initial data warehouse size.

Table A.1 Initial Data Warehouse Sizing (Page 1 of 2)

Module	Table	Source	Size Factor**	Source ERP Table Size	(Size Factor) * (Source Size)
Financials	ap_mstr	SE	1		
	ar_mstr	SE	2/3*		
	ard_det	SE	1		
	ck_mstr	SE	1		
	ckd_det	SE	1		
	gltr_hist	SE	1		
	tr_hist	SE + EE	1/10*		
	ee_CInvoice	EE	1		
	ee_CInvoiceMovement	EE	1		
	ee_DInvoice	EE	1		
	ee_DInvoiceMovement	EE	1		
	ee_Posting	EE	1		
	ee_PostingHist	EE	1		
	ee_PostingLine	EE	1		
Order Management	so_mstr	SE + EE	1		
	sod_det	SE + EE	1		
	idh_hist	SE + EE	1		
	ih_hist	SE + EE	1		
	ar_mstr	SE + EE	1/3*		
	tr_hist	SE+EE	1/10*		
Operations (assuming all sub modules - Purchasing, Inventory, OEE)	wo_mstr	SE+EE	1		
	wod_det	SE+EE	1		
	tr_hist	SE+EE	4/5*		
	op_hist	SE+EE	1		
	po_mstr	SE + EE	1		
	pod_det	SE + EE	1		
	prh_hist	SE + EE	1		
Enterprise Asset Management (EAM)	work_order		1		
	equip_down		1		
	lab_deta		1		
	squaks		1		

Table A.1 Initial Data Warehouse Sizing (Page 2 of 2)

Module	Table	Source	Size Factor**	Source ERP Table Size	(Size Factor) * (Source Size)
Transportation Management (TM) - assuming SQL Server source	xmshdr0		1/10*** x 1/3**		
	xmsitm0		1/10*** x 1/3**		
	xmsovr0		1/10*** x 1/3**		
	xmsxpk0		1/10*** x 1/3**		
	xmsypk0		1/10*** x 1/3**		
	xmtpas0		1/10*** x 1/3**		
	xmtpes0		1/10*** x 1/3**		
	xmtpfsp0		1/10*** x 1/3**		
	xmtpssp0		1/10*** x 1/3**		
Total Source Size for Conversion					X
Estimated Initial BI DW Size					X*(100)

Estimated Size of Initial BI Database = (X)*100

You should allow for this amount of space at a minimum. Further, if this total comes to under 500 GB it is recommended that you set up at least a 500 GB disk allowance.

*** TABLE NOTE 1:** A “size factor” of less than one indicates that not all transactions in the specified table are used by the relevant BI module, and this needs to be taken into account in size estimations. For example, in the BI Financials Module, the ar_mstr table has a size factor of 2/3, indicating that only two-thirds of all transactions on the ar_mstr table are converted into aggregated BI data in the BI Data Warehouse.

**** TABLE NOTE 2:** The source tables might contain data for a large number of years (n years, say $n=10$ years). If the data warehouse is only going to be populated with for example only 3 years worth of data ($y=3$), then an additional size factor of y/n (in this case $3/10$) should be applied to the relevant source ERP tables. And this extra size factor should be included in the table above and in the calculations to determine the estimated initial size of the data warehouse.

***** TABLE NOTE 3:** TMS source can be a SQL Server database. Assuming that is the case, apply this factor. It is different for different types of source databases.

Allowance for Future Database Growth

To accommodate the growth of the data warehouse over time, use the following guidelines.

In general, if the initial data warehouse size is based on transactions over the last 3 years, expect a data warehouse size growth over the coming 3 years to be of the same size as the initial data warehouse size. If the expected transaction volumes for the customer increase, then you should make extra allowance.

Example

If the initial data warehouse is created based on the last 3 years of activity, and the initial size is 250 GB, then expect the data warehouse size to at least double over the coming 3 years.

Hence, estimate of Disk Space Allocation to allow for medium-term data warehouse growth = 2 * (initial database size estimate) = 2 * ((X)*100).

Reference: Network Sizing Details

Terminology

- **Bandwidth.** This is a measure of the data transfer rate. Usually measured in “bits” per second (bps) or “kilo-bits” per second (Kbps) or “Bytes” per second (Bps) and so on. Each physical network connection between two points has a maximum bandwidth rating, which cannot be exceeded.
- **Latency.** This is a measure of the “time delay” of transferring data from one point on a network to another point on the network. It is usually measures in milliseconds (ms) or seconds (s).
- **WAN (Wide Area Network).** A network where there is a large distance (or latency) between computers on the network. This is in contrast to a LAN (Local Area Network) where there all machines on the network are “local” in the sense that there is minimal latency between the network points.

The following sections discuss the impact of bandwidth, latency, and WAN on BI components.

Network Sizing Considerations - Database to BI Database Server

Communication between the BI database server and the ERP database is required during the nightly BI daily load.

The BI DWD Scheduler communicates requests to the ERP database server using an ODBC connection, and in return the ERP database server returns all required data to the BI database server. Outside of BI daily load times there should be no communication between the two servers. If the daily load fails, the administrator or business may require the load to be rerun at a different time.

- Bandwidth requirements, why this is important, and how to address this by sizing properly.
- Effects of latency/packet loss (WAN), why this is important, and how to address this by sizing appropriately.

Importance of Latency and Bandwidth

Bandwidth considerations are important for this aspect of BI for the following reasons:

- As a large volume of data may be transferred between the ERP system and the BI server—100's of MBs, multiple GBs, depending on database size, transaction volumes, and so on—bandwidth must be suitably sized so as not to act as a bottleneck to ETL processing.
- The BI daily load must complete within a limited business-imposed daily time-window. The transfer rate of large data transfers can be limited by both bandwidth and latency. For this reason, sizing and allocating appropriate bandwidth is vitally important.

Bandwidth Requirements

Figures quoted here the peak bandwidth requirements, but note that this is not a sustained rate. The sustained bandwidth transfer rate is much lower than the peak rate.

- **LAN Guideline.** No more than 10 MB/s (80 Mb/s).
Note From current internal testing as of November 2011, we see LAN transfer rates of no more than 8 MB/s (3 MB/s normally) with a setting of 10,000 for fetcharraysize.
- **WAN Guidelines.** Under 5 MB/s. We see degradation to:
 - 0.4 MB/s at 200 ms latency (peak value)
 - 0.4 MB/s at 400 ms latency (peak value)

Impact of Latency on Bandwidth Requirements

Example If you take a customer with 1 GB of data to transfer, you can estimate the minimum amount of time for that data to transfer from the ERP system to the BI database server:

LAN: $1000 \text{ MB} / (3 \text{ MB/s}) = 333 \text{ seconds} = 5 \text{ mins } 33 \text{ seconds}$

WAN, 200 ms: $1000 \text{ MB} / (0.4 \text{ MB/s}) = 2,500 \text{ seconds} = 41 \text{ mins } 36 \text{ seconds}$

In reality the peak rates are sustained only very briefly and average transfer rates are much lower—some average transfer rates may be 10 times lower. These figures are only provided here as an indication of how bandwidth and latency can influence data transfer times, and how these effects can impact daily load times.

Network Sizing Considerations - Remote Client to BI Portal

The remote client issues requests to the BI Portal to retrieve data and subsequently processed data is transferred from the Portal server to the client machine. This type of usage exhibits a “spiky” network pattern where periods of data retrieval may require 100's of kB/s transfer rates separated by periods of little network activity.

Because the network access pattern is “spiky,” specifying an “average bandwidth” requirement per user can be slightly misleading.

Recommendation

100 kB/s bandwidth minimum should be reserved per user, bearing in mind the caveats below.

- **Caveat 1 – Dashboard loading.** The initial dashboard load may result in most network traffic when the dashboard data is returned from the server. This can result in brief periods of up to 300+ kB/s bandwidth demand.

- **Caveat 2 – Report Usage.** Depending on the size of the reports and frequency at which reports are run, heavy report usage can result in increased bandwidth demand.

WAN Considerations

Latency is an unavoidable consequence of WANs.

On WANs, latency has the effect of:

- Extending the data transfer times from the ERP system to the BI database system
- Increasing the average client response times above the expected response times for LAN (Local Area Network) based client-server communication

Latency will result in a decreased network transfer rate, thereby extending communication time, and decreasing bandwidth requirements, at the cost of overall responsiveness.

Customization Guidelines

This chapter describes QAD-recommended best practices for QAD Business Intelligence customizations. Topics include:

Overview 140

Customization Challenges 140

QAD BI 3.x Customization Methodologies 143

Customization Needs for Additional Columns 145

Identifying Every Appearance of a Column in Existing Code 152

Naming Conventions for Custom Changes to Packaged QAD BI Modules 153

Guidelines for Creating Custom Path Custom Tables 153

Adding Additional Columns to a Load Table 155

Adding Additional Custom Columns to Pre-perm Stage Tables 157

Adding Additional Custom Columns to perm/dim/fact Tables 159

Adding Custom Path stage/extract Tables After the perm Tables 161

Additional Customization Topics 166

Overview

Most customers that purchase the QAD BI modules invariably want additional information that falls outside of the baseline code of the warehouses. This comprehensive guide to QAD-recommended best practices for BI 3.x customizations should help you get the additional information you need. The QAD BI Data Warehouse Designer tool (DWD) makes it easy to add in additional columns and tables to existing BI modules. Doing so, however, can create difficulties at later points when a bug patch or upgrade is required for tables or modules that have been customized. It is important that any customizations are created in a way that avoids having to redo them if patches or upgrades are applied.

Customization Challenges

The DWD tool makes the addition of additional columns or tables to a data warehouse a fairly straightforward exercise, and it is one of the primary selling points of its creator, Wherescape. One of the great challenges in data warehousing is that when a data warehouse is built, it is difficult to make changes to it. Wherescape's approach to this eliminates much of the difficulty with getting these changes right, but the original intent of the tool is for a single customer to be able to make changes to their data warehouse. As QAD is building out its BI solution to package warehouses into different modules that are implemented for many different customers, it is a much more challenging environment to release bug patches and upgrade code. Each customer may have a different set of customizations and installing patches or upgrades to tables and procedures that have customizations can potentially wipe out the customization work and require it to be done again.

As an example, look at a customized `dim_item` table that then has a patch without the customizations applied to it.

Here is the layout for the `dim_item` table from the 3.4.1 release. The last several columns are circled for reference purposes.

dim_due_date	promotion_group	Promotion Group	varchar(80)	stage_item	promotion_group	Y	N	N	Y	Y	
dim_effective_date	current_cost	Current Cost T...	numeric(38,10)	stage_item	current_cost	###.00;...	Y	Y	Y	N	Y
dim_end_of_month_date	standard_cost	Standard Cost ...	numeric(38,10)			###.00;...	Y	Y	Y	N	Y
dim_entered_date	latest_standard_cost	Latest Standar...	numeric(38,10)	stage_item	latest_standard_cost	###.00;...	Y	Y	N	N	Y
dim_entity	net_weight	Net Weight	numeric(38,10)	stage_item	net_weight	###0	Y	Y	Y	N	Y
dim_exchange_rate	net_weight_um	Net Weight Unit...	varchar(30)	stage_item	net_weight_um		Y	N	N	Y	Y
dim_expected_pay_date	latest_current_cost	Latest Current ...	numeric(38,10)	stage_item	latest_current_cost	###.00;...	Y	Y	N	N	Y
dim_gl_transaction_type	dss_extract_timestamp	Extract Timesta...	datetime	stage_item	dss_extract_timest...	MM/dd/yyyy...	Y	N	N	Y	Y
dim_invoice_date	dss_process_batch_id	Process Batch Id	numeric(18)	stage_item	dss_process_batch_id	###; -#...	Y	Y	Y	N	Y
dim_item	dss_update_time	Update Time	datetime		dss_update_time	MM/dd/yyyy...	Y	N	N	Y	N
dim_last due date											

Here is the layout for the `dim_item` table for a customer that has a 3.3 customized version of the modules that not only has the easily identifiable custom columns that begin with an `x_` prefix (QAD's preferred prefix for custom columns now is `xx_`), but also note that they have some other columns that do not match the 3.4.1 `dim_item` table. These had been added at time of installation with the belief that these same columns would be added to the code base. That ended up not yet happening, so an upgrade to 3.4.1 is likely to require even more work and investigation. Unless columns are already in the code base, treat any additions as custom.

dim_discount_sub_account	configuration	Configuration T...	varchar(30)	stage_item	configuration
dim_draft_discount_date	promotion_group	Promotion Group	varchar(80)	stage_item	promotion_group
dim_draft_submission_date	current_cost	Current Cost T...	numeric(38,10)	stage_item	current_cost ###0
dim_due_date	standard_cost	Standard Cost ...	numeric(38,10)	stage_item	standard_cost ###0
dim_effective_date	latest_standard_cost	Latest Standar...	numeric(38,10)	stage_item	latest_standard_cost ###0
dim_end_of_month_date	latest_current_cost	Latest Current ...	numeric(38,10)	stage_item	latest_current_cost ###0
dim_entered_date	promotion_group_desc...	Promotion Grou...	varchar(80)	stage_item	promotion_group_d...
dim_entity	supplier_description	Supplier Descri...	varchar(30)	stage_item	supplier_description
dim_exchange_rate	buyer_description	Buyer Description	varchar(80)	stage_item	buyer_description
dim_expected_pay_date	abc_class	ABC Class	varchar(1)	stage_item	x_abc_class
dim_gl_transaction_type	safety_time	safety time	numeric(20)	stage_item	safety_time ###0
dim_invoice_date	dss_extract_timestamp	extract timestamp	datetime	stage_item	dss_extract_timest...
dim_item	dss_process_batch_id	process batch id	numeric(18)	stage_item	dss_process_batch_id ###0
dim_last_due_date	dss_update_time	dss update time	datetime	stage_item	dss_update_time
dim_last_receipt_date	x_reporting_unit_of_mea...	reporting unit o...	varchar(64)	stage_item	x_reporting_unit_of...
dim_layer	x_ship_day_tolerance	ship day tolerance	integer	stage_item	x_ship_day_tolerance ###0
dim_location	x_ship_quantity_tolerance	ship quantity to...	decimal(13,2)	stage_item	x_ship_quantity_tol... ###0
dim_order_date					

In SQL Server you can see that the table matches the meta-data observed using the DWD.

configuration (varchar(30), null)
promotion_group (varchar(80), null)
current_cost (numeric(38,10), null)
standard_cost (numeric(38,10), null)
latest_standard_cost (numeric(38,10), null)
latest_current_cost (numeric(38,10), null)
promotion_group_description (varchar(80), null)
supplier_description (varchar(30), null)
buyer_description (varchar(80), null)
abc_class (varchar(1), null)
safety_time (numeric(20,0), null)
dss_extract_timestamp (datetime, null)
dss_process_batch_id (numeric(18,0), null)
dss_update_time (datetime, null)
x_reporting_unit_of_measure (varchar(64), null)
x_ship_day_tolerance (int, null)
x_ship_quantity_tolerance (decimal(13,2), null)

And if we open the affiliated procedure, in the DWD, we can see what columns are being inserted into and that they match.

current_cost	standard_cost
latest_standard_cost	latest_standard_cost
latest_current_cost	latest_current_cost
promotion_group_description	promotion_group_description
supplier_description	supplier_description
buyer_description	buyer_description
abc_class	abc_class
safety_time	safety_time
dss_extract_timestamp	dss_extract_timestamp
dss_process_batch_id	dss_process_batch_id
x_reporting_unit_of_measure	x_reporting_unit_of_measure
x_ship_day_tolerance	x_ship_day_tolerance
x_ship_quantity_tolerance	x_ship_quantity_tolerance

```

FROM stage_item
WHERE NOT EXISTS (
  SELECT 1 FROM dim_item
  WHERE ( stage_item.source_system_code = dim_item.source_system_code
        OR ( stage_item.source_system_code IS NULL AND dim_item.source_system_code IS NULL ))
  AND ( stage_item.domain_code = dim_item.domain_code
        OR ( stage_item.domain_code IS NULL AND dim_item.domain_code IS NULL ))
)

```

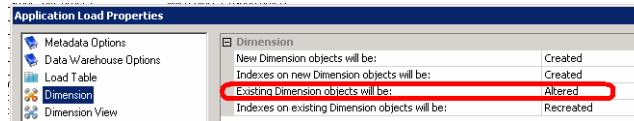
A patch is then applied using the Administrator tool to fix some issue with the table or to upgrade the table to the 3.4.1 version of the code. You can see that it initially says that existing objects are replaced.

Object	Status
Application load begins	D:\kurt\work\custom_testslapp_obj_Patch_Test_1.vst
set_dim_item_key	Will replace existing object
update_dim_item	Will replace existing object
update_stage_item	Will replace existing object
update_stage_item_site	Will replace existing object
update_stage_order3	Will replace existing object
update_stage_order2	Will replace existing object
update_stage_order1	Will replace existing object
custom_stage_order1	Will replace existing object
update_fact_order_history	Will replace existing object
custom_fact_order_history	Will replace existing object
dim_item	Will replace existing object
stage_item	Will replace existing object
stage_item_site	Will replace existing object
stage_order3	Will replace existing object
stage_order2	Will replace existing object
stage_order1	Will replace existing object
fact_order_history	Will replace existing object
dim_item_idx_0	Will replace existing object
dim_item_idx_A	Will replace existing object
stage_order1_idx_A	Will replace existing object
fact_order_history_idx_1	Will replace existing object
fact_order_history_idx_1n	Will replace existing object

Setup Administrator

47 Object/s will be replaced. You may cancel at this point and review the objects to be replaced and/or added.

However, the default setting for each object during application installation ensures that the dim, perm and fact tables are only altered by changes and the updates proceed successfully.



A refresh of the DWD Browser pane to review the changes to dim_item then reveals that the data in the tables looks exactly like the 3.4.1 version of dim_item. Any other columns that had existed for this table in the DWD's meta-data are no longer shown.

dim_due_date	promotion_group	Promotion Group	varchar(80)	stage_item	promotion_group	Y	N	N	Y	Y	
dim_effective_date	current_cost	Current Cost T...	numeric(38,10)	stage_item	current_cost	###.00;...	Y	Y	N	Y	
dim_end_of_month_date	standard_cost	Standard Cost ...	numeric(38,10)	stage_item		###.00;...	Y	Y	N	Y	
dim_entered_date	latest_standard_cost	Latest Standar...	numeric(38,10)	stage_item	latest_standard_cost	###.00;...	Y	Y	N	Y	
dim_entity	net_weight	Net Weight	numeric(38,10)	stage_item	net_weight	##0	Y	Y	N	Y	
dim_exchange_rate	net_weight_um	Net Weight Unit...	varchar(30)	stage_item	net_weight_um		Y	N	N	Y	
dim_expected_pay_date	latest_current_cost	Latest Current ...	numeric(38,10)	stage_item	latest_current_cost	###.00;...	Y	Y	N	Y	
dim_gl_transaction_type	dss_extract_timestamp	Extract Timesta...	datetime	stage_item	dss_extract_timest...	MM/dd/yyyy...	Y	N	N	Y	
dim_invoice_date	dss_process_batch_id	Process Batch Id	numeric(18)	stage_item	dss_process_batch_id	#####;...	Y	Y	Y	N	Y
dim_item	dss_update_time	Update Time	datetime	stage_item	dss_update_time	MM/dd/yyyy...	Y	N	N	Y	N
dim_last due date											

When the patch was applied, we had selected to not change existing tables, only alter them. Why are the other columns gone? Going to SQL Server reveals that they still exist in the database and that any new columns that were found when the patch was applied were placed at the end of the table column list. In this case, net_weight and net_weight_um were new columns in the 3.4.1 table.

configuration (nvarchar(30), null)
promotion_group (nvarchar(80), null)
current_cost (numeric(38,10), null)
standard_cost (numeric(38,10), null)
latest_standard_cost (numeric(38,10), null)
latest_current_cost (numeric(38,10), null)
promotion_group_description (varchar(80), null)
supplier_description (varchar(30), null)
buyer_description (varchar(80), null)
abc_class (varchar(1), null)
safety_time (numeric(20,0), null)
dss_extract_timestamp (datetime, null)
dss_process_batch_id (numeric(18,0), null)
dss_update_time (datetime, null)
x_reporting_unit_of_measure (varchar(64), null)
x_ship_day_tolerance (int, null)
x_ship_quantity_tolerance (decimal(13,2), null)
net_weight (numeric(38,10), null)
net_weight_um (nvarchar(30), null)

Reviewing the affiliated procedure in the DWD shows that all the logic referencing the custom columns has been removed and the new logic only reflects the columns that exist in the 3.4.1 version that was applied.

```

configuration      promotion_group
promotion_group     current_cost
current_cost        latest_standard_cost
latest_standard_cost
net_weight          net_weight_um
net_weight_um       latest_current_cost
latest_current_cost dss_extract_timestamp
dss_extract_timestamp
dss_process_batch_id
FROM
stage_item
WHERE NOT EXISTS (
SELECT 1 FROM dim_item
WHERE ( stage_item.source_system_code = dim_item.source_system_code
OR ( stage_item.source_system_code IS NULL AND dim_item.source_system_code IS NULL ))
AND ( stage_item.domain_code = dim_item.domain_code
...OR ( stage_item.domain_code IS NULL AND dim_item.domain_code IS NULL ))

```

Note Since the new columns were added to the table and the table was not dropped, the custom columns still would contain their data, but would no longer be updated by the new procedure.

As you can see, implementing patches and upgrades to tables that have any custom work done to them is not a straightforward process and likely requires a fair amount of followup work to get the customizations back in place in the DWD meta-data (the underlying SQL table can remain

unchanged). It is for this reason that **ideally** custom work will not be done directly on standard release tables in the data warehouse or at the very least, the footprint of the customized work will mostly fall outside of the baseline code.

QAD BI 3.x Customization Methodologies

With the potential problems affiliated with adding custom columns to our existing standard infrastructure, it is clear that whenever possible, an alternate way of doing custom work needs to be utilized to ensure easy patch and upgrade installations.

Custom Path Methodology (Preferred)

To avoid the pitfalls of making changes to our existing baseline tables at customer sites, it would be much simpler if the custom requirements could be met without ever touching the existing infrastructure. This is nearly a possibility! The approach to resolve this issue is to create a custom path for the custom work that has its own separate objects and is integrated into the baseline fact/dim by a custom procedure that updates the custom columns in the dim/fact table. This task is run after the regular tasks for populating/updating that table have completed.

Integration Points

The customized objects are integrated with the baseline objects using update procedures. If this option is chosen, any time a patch impacts the baseline objects then the custom column needs to be added back into the DWD but not the actual database. If cubes are used and the new customization desires them, the additional customization should be added as a new attributes to the existing OLAP dimension.

Note The grain of the update procedure should be the same as the core dimension table.

Processing Flow

Fig. B.1

DWD Processing Architecture with Customized Objects and Update Procedure for a Dimension

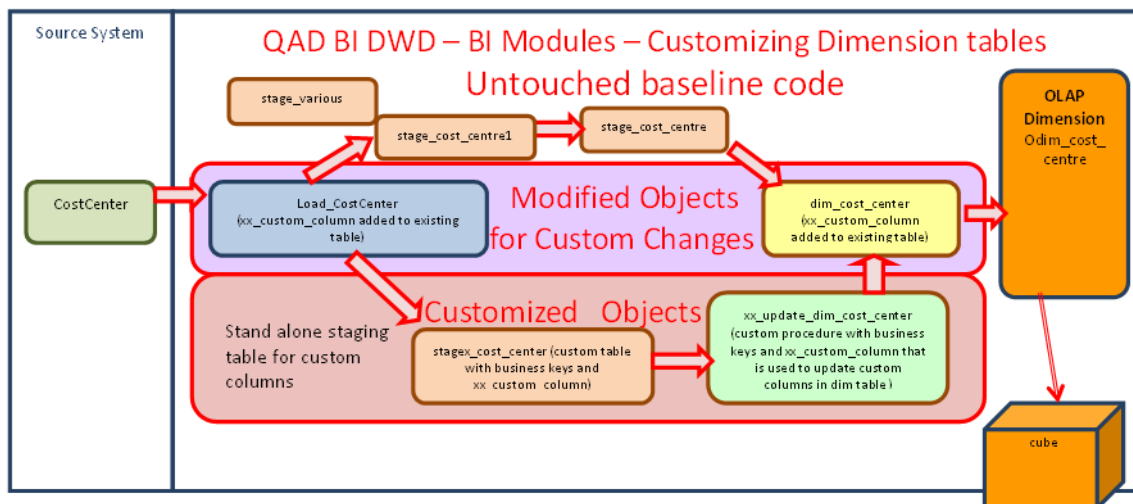
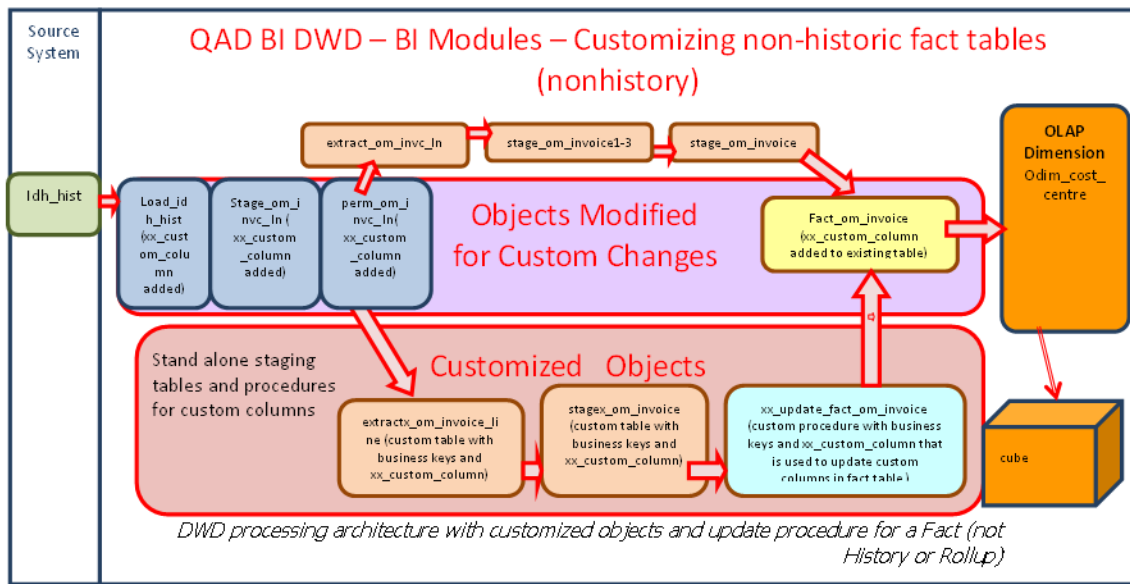
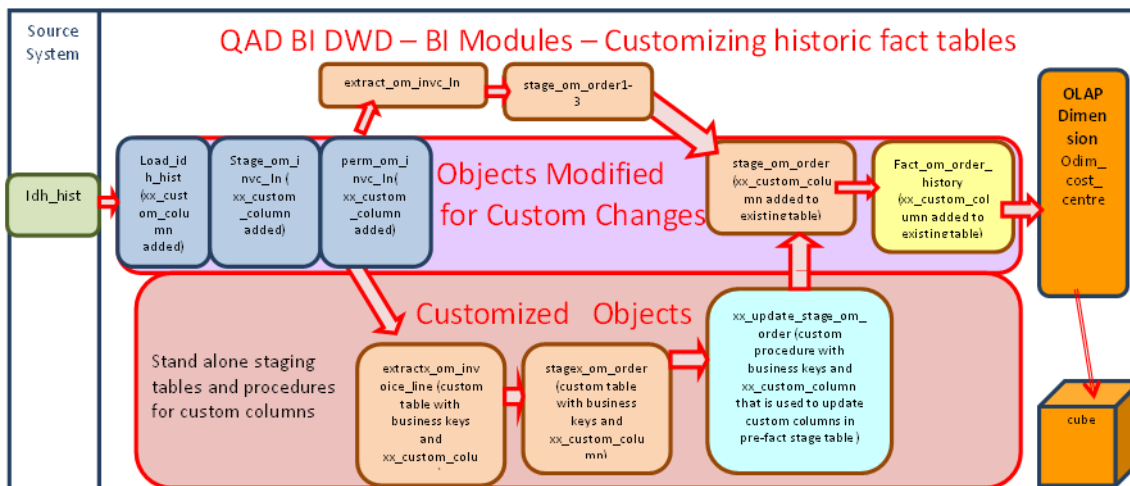


Fig. B.2

DWD Processing Architecture with Customized Objects and Update Procedure for a Fact (not History or Rollup)

**Fig. B.3**

DWD Processing Architecture with Customized Objects and Update Procedure for a Fact (History)



Pros and Cons

Pros

- Any existing reports would continue to work as the dimension only has additional fields added.
- Separation of baseline and customizations tables and stored procedures is maintained throughout the processing, making upgrades much easier.
- There is a fairly low risk of data loss with this method of integration because the core dimension and fact tables only have extra columns added to them by the customizations that are populated by stored procedure. In the event of a patch or upgrade, the columns need to be restored in the Designer, and the custom column update procedure re-added to any jobs it may have been lost from.

Cons

- This method of customization requires custom columns to be added directly to the dim/fact table. After an application upgrade or patch, custom columns need to be reapplied in the DWD so they are still visible to end users.
- Performance time-wise could be impacted by having this parallel path and by running a separate procedure to update dim or fact tables that were just populated by their normal insert/update processes.
- There is more up-front work in creating the parallel table path, instead of just adding columns to existing infrastructure.

Integrated Methodology (Last Resort)

There are times when the sheer complexity of the staging and transformation processes makes the use of a parallel table path too difficult to set up. This is particularly true for some fact tables, in particular those that do a lot of ranking, like the `fact_om_order_history` table. If an attempt has been made to create a parallel path and it is just too difficult to achieve a satisfactory result, adding custom columns to the existing infrastructure can be allowed. Future patches and upgrade installations will have to take these into account and they will have to be reimplemented when they get disrupted. The customer needs to be informed of this possibility as it may require an additional services engagement to reimplement the customizations.

Pros and Cons

Pros

- Easier to do an initial implementation.
- Ensures the customization follows the same path that like data is moving through, eliminating risk of somehow getting the data connections wrong.
- No need for keeping separate sets of views or staging tables.

Cons

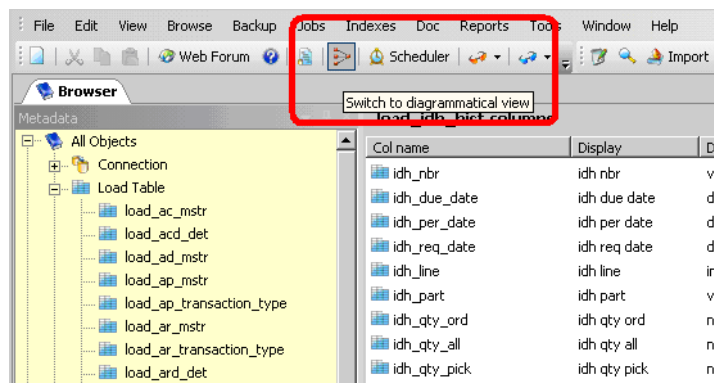
- Far more difficult to do upgrades or install patches as customization work needs to be reimplemented after the patch is applied. This is the number-one reason why this methodology is a last resort.
- Creates difficulties for support in trying to determine how to resolve issues with the custom code.
- There is some risk of disrupting existing logic if custom work is not done right. The implementer needs to be very careful and aware of Cartesian joins.

Customization Needs for Additional Columns

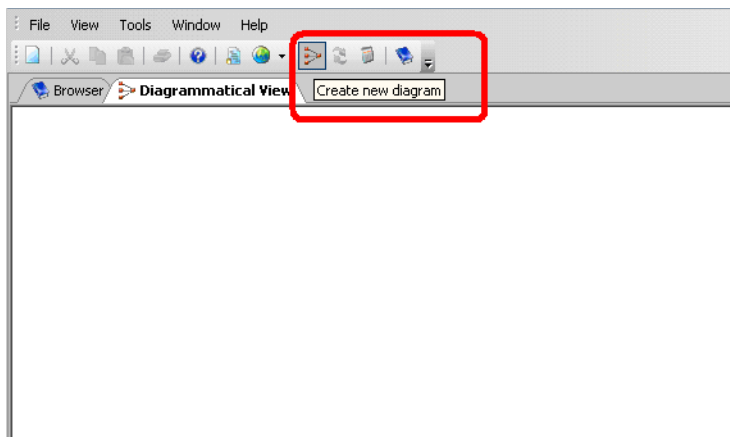
When data from a QAD Progress table is identified as important to a customer and does not exist in one of the BI fact tables, it is important to check first to see if the data is already being pulled over into one of the pre-staging tables as well as if the source table is already being referenced.

Use Track Back Diagram to Check Tables

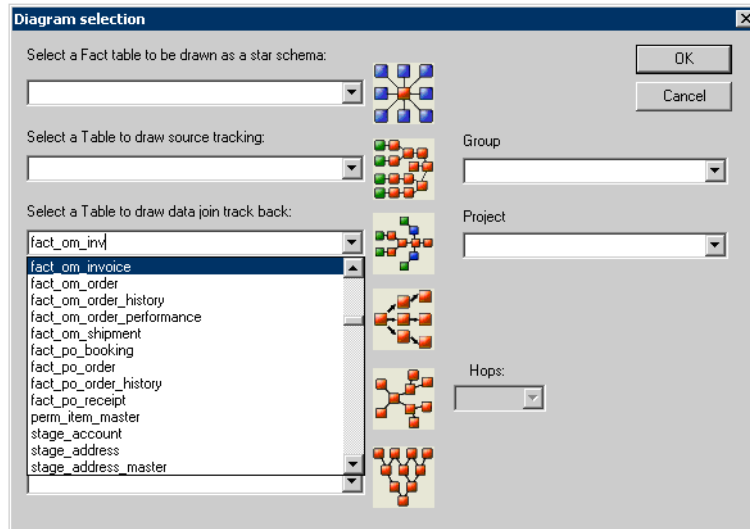
- 1 First do a track back diagram of the fact_ or dim_ table that the custom column is to be added to.
- 2 Click the Diagram icon to the left of the Scheduler icon in the QAD BI Data Warehouse Developer (DWD).



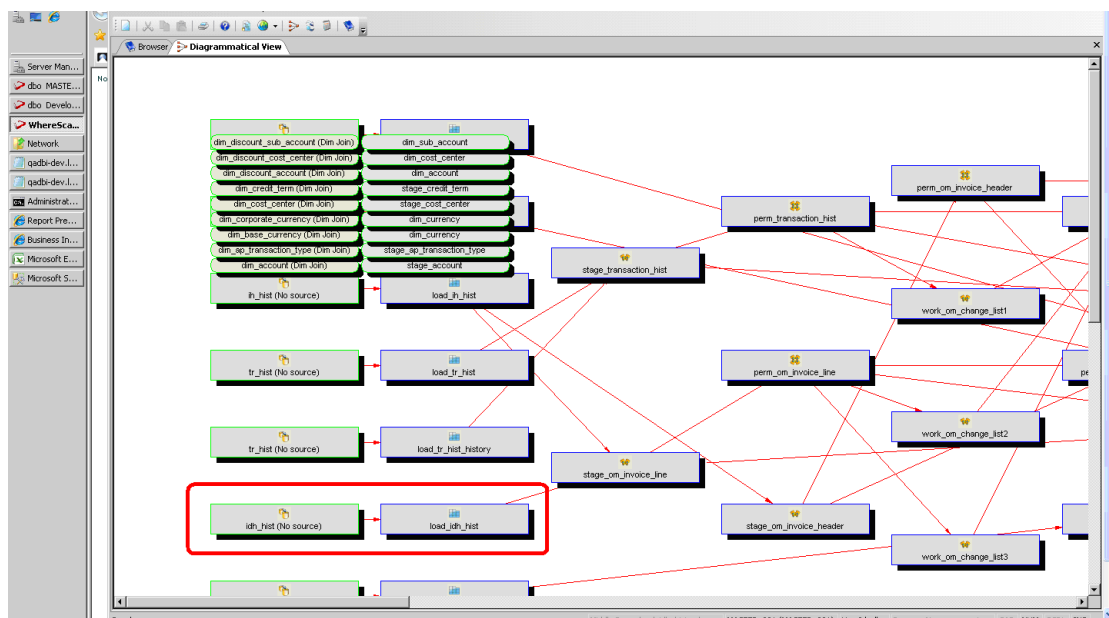
- 3 If the Diagrammatical View tab was not already open, you have to click the icon a second time.



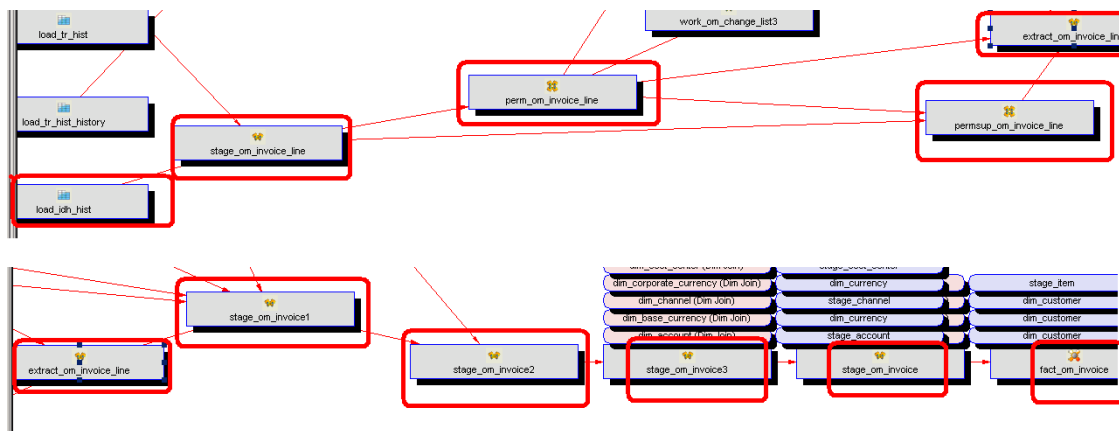
- 4 Pick the fact table you want to track back on and click OK. You can start to type in the name of the table to find it in the drop-down selection list faster, but if you type the whole name in, you still have to select the item from the drop-down list otherwise nothing appears when you hit OK.
- 5 For the purpose of this example, we use the fact_om_invoice table and we look to add a custom column pulling data idh_prodline from the idh_hist table.



- 6 A track back diagram should appear displaying all the tables that feed the fact_ table.
- 7 Identify here if the source table of the new data is already being referenced by the fact_ table.



- 8 If the source table is included in the diagram, note its data path, taking down the names of any load_, stage_, perm_, extract_, and other tables that might contain that information.



9 The list should contain:

- load_idh_hist
- stage_om_invoice_line
- perm_om_invoice_line
- permsup_om_invoice_line. The permsup_om_invoice_line table is used to store parallel date data about the perm_om_invoice_line table based on the primary business key. No custom work should be necessary here unless the business key is changing for some reason.
- work_om_change_list1-3 and hist. The work tables that you can see in the first of the two dataflow diagrams above are for determining new data based on dates, so they were not included in the list to be modified.
- extract_om_invoice_line
- stage_om_invoice1
- stage_om_invoice2
- stage_om_invoice3
- stage_om_invoice
- fact_om_invoice

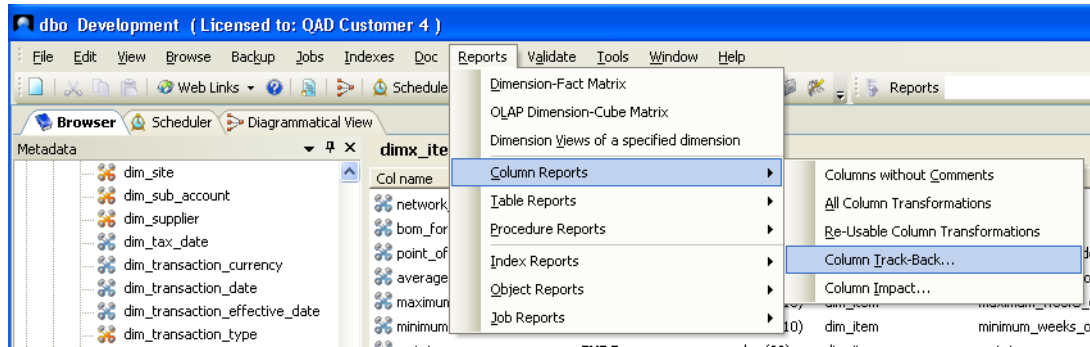
As a matter of practice, it is helpful to type this list into notepad or some other text editor for reference purposes as you work through the customization.

Double Checking the Data Path with a Column Track Back Report

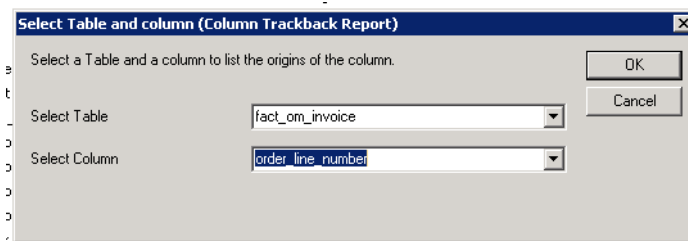
To be extra sure your table list is complete, you can do a track back on a column name from the fact table if you know that column to be from the same source table as the one that the custom column would be added from.

- 1 Go back to the Browser tab, select Reports|Columns|Column Track-back.

Note Prior to the 6.5.1 version of the DWD, this option was second from the bottom in the reports list as Track back on specific column usage.

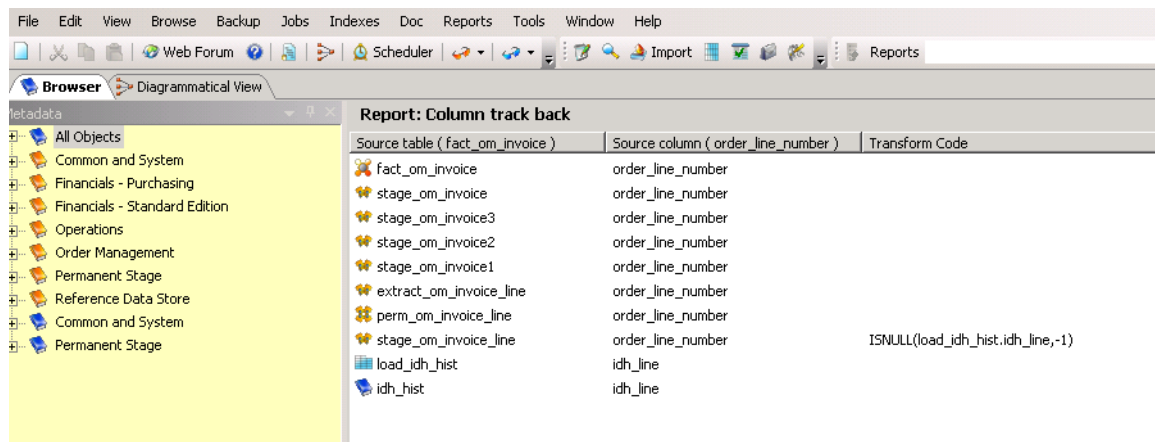


- 2 You are prompted with a box. Select the name of the fact_ table and a column you know comes from the source table in question. In this instance, we know that order_line_number comes from the idh_mstr table.



- 3 Once picked, click OK. A report is generated tracking back the column from the Select Table back to its source.

Note For some dimension values in dim_ or fact_ tables, this report is more effective starting prior to the staging table that the dim_ value keys was defined in.



- 4 Aside from showing you all the tables that lead to the fact table, you also see the column names in those tables and any transform code that occurs for a column. This table list closely matches the list we wrote down from the Diagramatic View.
- 5 We have identified that the QAD table the desired extra data is sourced from as already existing in the dataflow as a load_ table. We now need to see if the column in question also exists in that load_ table.

In the DWD go to Browser|All Objects|Load Table, select the load_ table, and see if the needed column is in the list.

Note You can find columns more quickly and effectively by clicking on the Col name. In this case, `idh_prodlne` is not included in the `load_idh_hist` table.

Col name	Display	Data type
idh_list_pr	idh list pr	numeric(38,
idh_loc	idh loc	nvarchar(80
idh_lot	idh lot	nvarchar(80
idh_nbr	idh nbr	nvarchar(80
idh_part	idh part	nvarchar(30
idh_per_date	idh per date	datetime
idh_pickdate	idh pickdate	datetime
idh_price	idh price	numeric(38,
idh_pricing_dt	idh pricing dt	datetime
idh_project	idh project	nvarchar(80
idh_promise_date	idh promise date	datetime
idh_qty_all	idh qty all	numeric(38,
idh_qty_chg	idh qty chg	numeric(38,
idh_qty_cons	idh qty cons	numeric(38,
idh_qty_exch	idh qty exch	numeric(38,
idh_qty_inv	idh qty inv	numeric(38,
idh_qty_ord	idh qty ord	numeric(38,
idh_qty_pend	idh qty pend	numeric(38,

- 6 If it did exist in the column list, the next step is to check to see if it exists in the `stage_table`. If it does not exist, review the steps in this process before proceeding.

- 7 Go to the Stage Table section of the Browser and find the staging table that was second in the list of tables above; in this instance, `stage_om_invoice_line`.

- 8 When you click on the table name, look for the source column that is to be added in the Src column list.

Note You can click on the header to sort alphabetically if the list is long and it is not immediately obvious if the column is there or not. In most instances, the column name in this first staging table is changed to something with a longer name than the source column name.

- 9 Continue going through this process of tracking the soon-to-be-added custom column through the `stage_`, `perm_`, `extract_` and more `stage_` until you find a table in the path that does not contain it.

Notes on Tables

This section describes some things to note about tables.

load_tables

`Load_tables` contain a straight extract from their source system. The majority of the BI extracts come directly from the QAD ERP system's Progress databases. The only change or transformation that should occur for these columns is to clean up any blank spaces that may have been inadvertently left in the Progress table or to remove the '|' symbol from the data as it will disrupt the data load.

Note There are some exceptions to this rule.

The transformation looks like this:

```
REPLACE(REPLACE(REPLACE(idh_hist.idh_type,'|',' '),CHR(10),' '),CHR(13),' ')
```

This logic should be used for any source table character (char,varchar) type column that gets added to the load table.

stage_ tables

Stage tables are used to do a variety of things that transform the data in some way. The stage tables that come right after load tables are typically used to just rename the column names into something more readable as they pass the data up to the perm or dim tables or other staging tables. Any later in the flow, stage tables are doing something with the data to make it more usable.

perm_ tables

In many instances, there are perm tables that hold data that is permanently maintained in the source system because it would put a heavy load on the system to be extracted each run. The perm has one stage table prior to it where the column names change from the load table and no transformations should occur prior to a perm table. The data in the perm table should essentially represent a like image of what is contained in its source system. Perm tables are one of the three types of tables with which great care must be taken to not inadvertently drop or delete data.

work_ tables

These tables are a type of stage_ table used to check the perm tables to see if any data has changed as per the date requirements set for that work table. This might mean that the work table is looking for any records that have been updated in the perm tables in the last 10 days. For a historic load, all the records are pulled from the perm table. The determination of what date values to use is usually derived from parameters, such as PO_HISTORY_DATE, which is the day to start looking back from, and PO_PROCESS_DAYS, which is the number of days that should be looked back from PO_HISTORY_DATE.

Custom paths may reference the existing work_ table infrastructure.

extract_ tables

Perm_ tables should have at least one and in some cases multiple extract tables. These tables use the records as determined by the work_ tables to extract those records from the correlating perm_ table. Then they prepare them to be picked up by the staging tables and fed on to fact_ tables of that particular module's Job.

A custom path referencing a perm_ table needs its own extractx_ table which contains the business key and the required additional columns.

dim_ tables

Dim_ tables (dimension tables) are where non-calculated potentially redundant information is stored in the data warehouse. Dim_ tables are the points of a star schema. It is very critical that dim_ tables are not accidentally dropped or that data is deleted from them. Because they hold the keys that the fact uses to link to them, there is no guarantee that a dim_ table will create keys in the

same sequence if it needs to be repopulated. Dropping and re-creating an existing dimension from a populated warehouse likely forces you to repopulate all the fact tables that reference that dim_table.

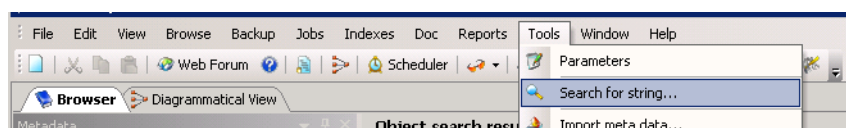
fact_tables

The fact_tables are where the calculated values of the data warehouse are stored. Fact_tables are the center of a star schema. It is very important that fact_table data is not inadvertently deleted or the table re-created, but recovering from accidentally doing so is not as painful as it is for perm_ or dim_tables.

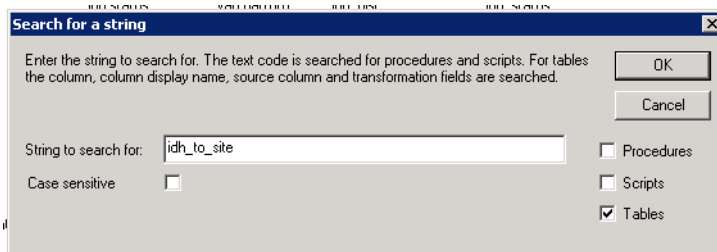
Identifying Every Appearance of a Column in Existing Code

The following steps are one last shortcut in the column search process:

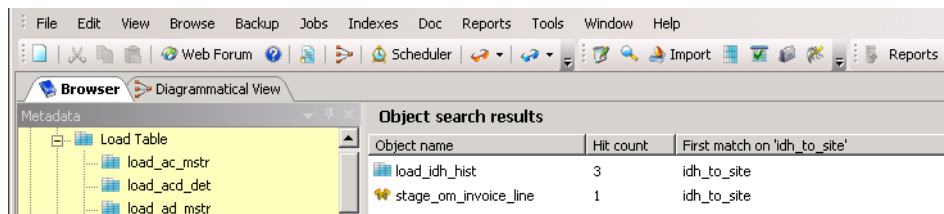
- 1 If you believe that the column is somewhere in the warehouse and it didn't show up through the prior methods, you can also do a search for the character string in the code.



- 2 Depending on what you are looking for, pick the relevant boxes; in this case, only Table would be the best choice.



- 3 Results are displayed as follows.



Naming Conventions for Custom Changes to Packaged QAD BI Modules

It is important that any custom work is always named per our defined standards so that anybody looking at code, such as support, services, or customers, can feel confident that the custom work has been applied in a uniform way and that identifying the custom work is straightforward. We have three different categories of naming conventions, which are outlined below.

Custom Tables

Any new tables that need to be created should have an x affixed to the first part of their name before the first underscore:

loadx_, stagex_, permx_, extractx_, factx_, dimx_, olapx_

By naming them this way, the custom tables can easily be distinguished from the regular tables in the modules and readily found. They show up at the bottom of the list for each type in the Browser pane in the DWD tool.

Custom Columns Joined to Standard Module

Any new columns added to any of the existing tables or new parallel path custom tables that eventually join with existing tables should be prefixed with an xx_ to identify them as such.

Example If you were to add a column called recipient_date to an existing table in the standard release of the module, it should be named xx_recipient_date. The reason for using xx_ is that the underline acts like wild card in the DWD's Search string feature, so a search for any columns that contain x_ will return any value that contains an x, such as tax_code. By using xx_, searches can be narrowed dramatically to only columns that contain xx—which should be nearly all customizations.

Custom Columns Not Joined to the Baseline Module Tables

Any columns in new custom tables that are not going to end up being joined to existing dim_ or fact_ tables as part of a view or similar methodology can forgo any prefixing of the column names. The table that they are in, however, should be properly prefixed with an xx as explained in “Custom Columns Joined to Standard Module”.

Note Putting in an xx_ prefix for these does ensure that we can easily find all custom columns in the system though.

Guidelines for Creating Custom Path Custom Tables

This section describes QAD's preferred and recommended custom methodology.

To avoid difficult and challenging upgrades and patch installs, custom work should ideally reside in its own set of parallel tables that closely mimic the tables and logic that we are already using to populate our dim_ and fact_ tables. If a data path has been identified as already existing from the same source as the requested custom column, and that path does not appear overly complex, it is a

simple matter to create a series of tables that follow that path in parallel to get our data to its ideal destination, a view created from the existing dim or fact and the dimx or factx. Even under this methodology, there are exceptions to the rules.

Load Tables

Despite the ideal design of having custom path tables to avoid problems with upgrades, for load tables, if the source of the data for the new custom column is coming from a source that is already being accessed by a load table in the data warehouse, it would be redundant to repeat the action of loading a second custom table from the same source for just one new column. So in an example as provided earlier where `idh_prodlne` needs to be added to the `fact_om_invoice` table, it should be added as a custom column `xx_idh_prodlne` to the existing `load_idh_hist` table. In cases where the required new columns from source tables that are not being utilized by the existing warehouse, a whole new `loadx_` table should be created and should contain all relevant columns for linking that data to the existing data warehouse if that is the intent.

Perm Tables and Pre-perm Stage Tables

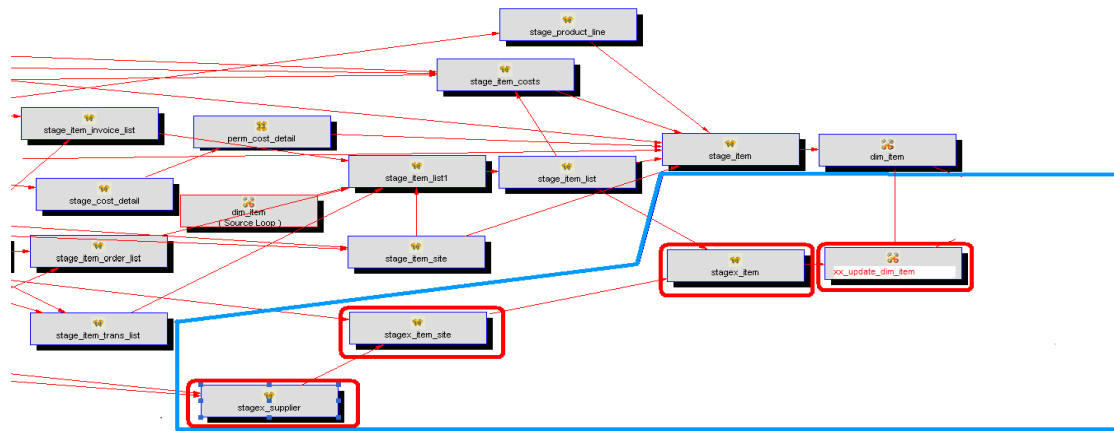
The data from the source table through the load, post-load/pre-perm stage and then perm table in the baseline code is usually left intact with few or no transformations. Changes to these tables in future releases are limited and require a minimal number of required changes to then re-add custom columns to this handful of tables if a patch or upgrade does disrupt their infrastructure. As a result, it is advisable to add new custom columns to the existing tables as is recommended for the load tables. If the customizations occur after a historic load for a perm table like `perm_transaction_history`, which only adds new records on an incremental basis, any custom columns added to that table need to get populated using a full historic reload or a special one-time script to update the perm table with the missing column data.

All Other Stage, Extract

To add parallel or custom path tables is a simple drag-and-drop procedure from the source pane where you copy the entire table and then delete the unnecessary columns. The logic of the WHERE clause that the new table is mimicking should be copied over and made part of the new custom path tables logic so that everything uses essentially the same logic.

Adding Required xx_ Prefixed Columns

After the custom path has been created all the way to the dim or fact (non-history) table, add the extra required `xx_` prefixed columns to that table. In the column definition, do *not* give the new `xx_` prefixed custom column a source table; it needs to be blank. Transform those column values to some dummy value and then build a procedure with an `xx_` prefix that updates the new custom `xx_` columns **after** the table is populated during its normal load process. Below is an illustration of what the resulting diagram is for a parallel path effort for the `dim_item` table.



Adding Additional Columns to a Load Table

To add an additional column to a load table that is coming from the same source, click on the load table that the column is to be added to so that you can see all the columns for that table in the target pane.

Then do the following:

Get Column from Data Source

- 1 Open a connection to the Progress database in the source pane and find your source table.
- 2 Double-click on that source table so that all columns appear.
- 3 Find the new column that currently is not being utilized and drag it into the Target Pane to the bottom of the column list so that it is the last column in the table.

Object name	Hit count	First match on 'x'
ConvertCurrency/value	18	DECLARE @v_ex_rate NUMERIC(38,10)
ConvertCurrency/valueTrans	39	, @p_trans_ex_rate1 NUMERIC(38,10)
custom_dim_credit_term	22	SET XACT_ABORT OFF -- Turn off auto abort on errors
custom_extract_ap_check...	28	-- Script Name : custom_extract_ap_check_det
custom_extract_ap_check...	30	-- Script Name : custom_extract_ap_check_rst

Naming/Column Placement/Data type

- 4 Now change the prefix of the column name to identify it as a custom addition.
- 5 Check the Column order and make sure that the new custom column appears after the last column number in the table—source_system_code above is Column order 490.
- 6 Finally, if the column is of a varchar datatype, change it to be an nvarchar datatype.

Column definition for load_idh_hist.xx_idh_prodlne

Column Properties | Transformation

Table: load_idh_hist

Column name: xx_idh_prodlne

Business display name: idh prodlne

Data type: nvarchar(30)

Format:

Column order: 500

Nulls: ☒

Numeric: ☐

Additive: ☐

Attribute: ☒

Source table: idh_hist

Source column: idh_prodlne

Business Definition:
Source strategy: (max 2048 chars.)

Transform

- 7 If the column is originating from any type of char, varchar, or nvarchar datatype column in the source, perform a transformation on the column to strip out blank spaces and | symbols, which will disrupt data loads. This is done using the following commands, which should be found in any other varchar data type columns in the table.

```
REPLACE(REPLACE(REPLACE(idh_hist.idh_prodlne,'"',''),CHR(10),' '),CHR(13),' ')
```

This is placed using the Transformation tab.

- 8** The result should be as seen below. Scrolling very far to the right shows the transformation.

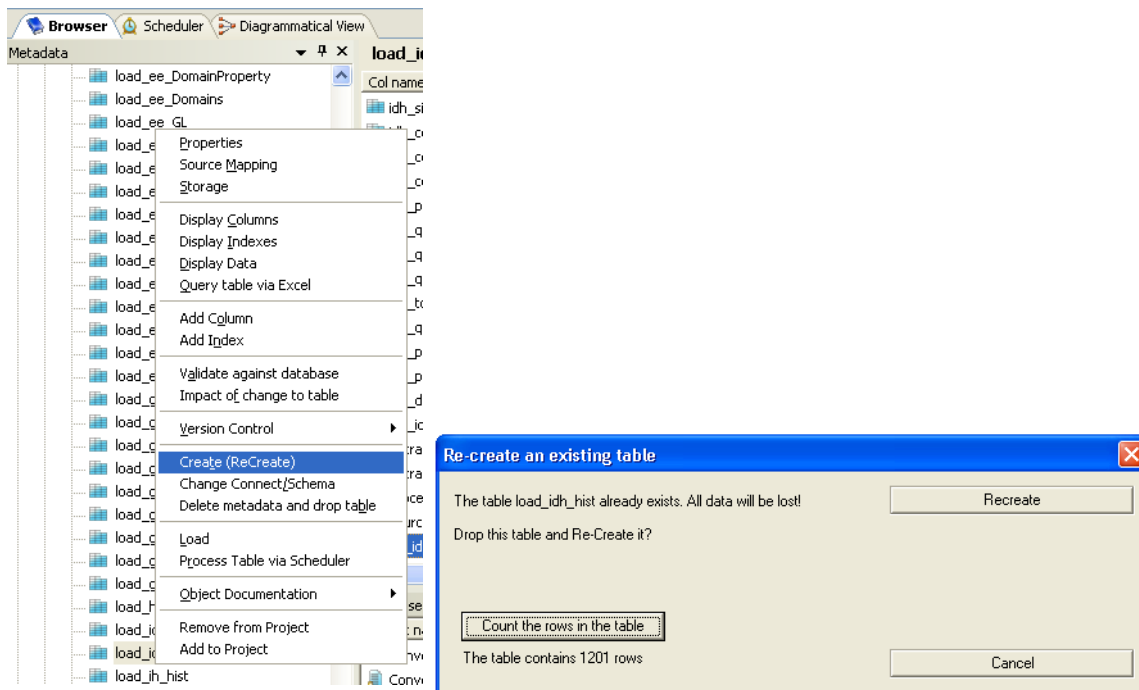
The screenshot shows the SQL Server Enterprise Manager interface. On the left, a tree view lists several tables: load_gen_hist_report_line_range, load_gen_transaction_type, load_gen_cic, load_gen_hist, load_gen_hist_history, load_gen_hist_list, load_gen_mstr, load_gen_ctrl, load_gen_hist, load_gen_hist, load_gen_hist, and load_gen_hist. The 'load_gen_hist' table is selected and highlighted in blue. The main pane displays the structure of the 'load_gen_hist' table. The table has four columns: extract_row_id (numeric(18)), process_batch_id (numeric(18)), source_system_code (nvarchar(64)), and isdh_hist (isdh_hist). The 'load_gen_hist' table is highlighted in blue, and the 'load_gen_hist_history' table is highlighted in yellow.

Create (Recreate) the Table

- 9** With all the changes in place, it is time to re-create the table.

Warning Do not use Create (Recreate) for Dim, Fact or Datastore/Perm tables unless you are sure they are empty. For dim, fact and perm tables, you may need to do a Validate/Modify process instead. See “Modify Table” on page 160 for an explanation of the Modify table process.

- 10 In the Browser Pane, right-click on the table name, and click the Create (Recreate) option.
- 11 When the window comes up, do a quick check of the number of records in the table you are about to drop by clicking the Count rows in table button. If ready to move ahead, click Recreate.



- 12 A confirmation that the table re-created correctly should appear in the results pane.

Results		
Object	Message	
load_idh_hist	table comment added.	
load_idh_hist	44 column comments added.	

Adding Additional Custom Columns to Pre-perm Stage Tables

To add an additional column to a staging table, go to the Browser pane and click on that staging table so that the table's columns are in the Target pane.

Then do the following:

Connect

- 1 Connect to the Data Warehouse in the Source pane and find the appropriate source table.
- 2 Double-click on the table and find the column to be added.
- 3 Click on the column and drag it to the Target pane at the end of the column list.

Note Note that the DWD did not place the column at the very end of the column list. It may be easier to edit the last column (dss_update_time) and change the column order date for it to 535 so that xx_idh_procline is last.

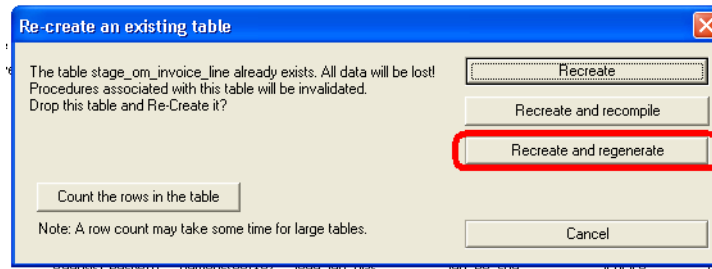
Transformation/Business Name

- 4 Apply any transformation that might be applicable to this column. If this is the pre-perm staging table, this is the best spot to also rename the business name of the column.

Create(Recreate) the Staging Table

- 5 With the new column in place, Create(Recreate) the staging table. In the Browser pane, right-click on the table name and select the Create(Recreate) option.
- 6 When the window pops up, click Recreate and Regenerate.

Note This only works as long as the existing procedure is not *Modified*. Otherwise manual changes to the procedure will need to be made.



Adding Additional Custom Columns to perm/dim/fact Tables

To add an additional column to a perm, dim or fact table, go to the Browser pane and click on that staging table so that the table's columns are in the Target pane.

Then do the following.:

Data Source

- 1 Open the Source Pane using the Data Warehouse ODBC connection.
- 2 Find the source staging table to be referenced.
- 3 Double-click the table to get a column list. Drag and drop the column into the perm (fact/dim) table's Target pane to the very bottom of the column list.

Placement

- 4 Ensure that the column is indeed at the very bottom of the list.
- 5 Drag and drop any columns below it if it will not go any lower, or renumber the Column Order if necessary to get it to be the last.

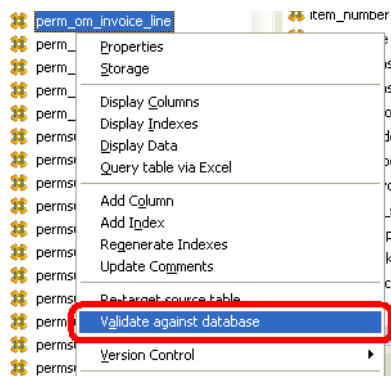
perm_po_order_header	due_date	Due Date	datetime	stage_om_invoice_line	due_date	Y	N	N	Y	Y	430
perm_po_order_line	performance_date	Performance Date	datetime	stage_om_invoice_line	performance_date	Y	N	N	Y	Y	440
perm_po_receipt_hist	required_date	Required Date	datetime	stage_om_invoice_line	required_date	Y	N	N	Y	Y	450
perm_transaction_hist	revised_date	Revised Date	datetime	stage_om_invoice_line	revised_date	Y	N	N	Y	Y	460
permsup_ee_credit_invoice	expiry_date	Expiry Date	datetime	stage_om_invoice_line	expiry_date	Y	N	N	Y	Y	470
permsup_ee_credit_invoice_move	pick_date	Pick Date	datetime	stage_om_invoice_line	pick_date	Y	N	N	Y	Y	480
permsup_ee_debit_invoice	pricing_date	Pricing Date	datetime	stage_om_invoice_line	pricing_date	Y	N	N	Y	Y	490
permsup_ee_debit_invoice_move	promise_date	Promise Date	datetime	stage_om_invoice_line	promise_date	Y	N	N	Y	Y	500
permsup_ee_posting	shipment_type_code	shipment_type...	nvarchar(30)	stage_om_invoice_line	shipment_type_c...	Y	N	N	Y	Y	510
permsup_ee_posting_line	dss_create_time	dss_create_time	datetime	stage_om_invoice_line	dss_create_time	Y	N	N	Y	Y	1000000008
permsup_mrp_detail	dss_update_time	dss_update_time	datetime	stage_om_invoice_line	dss_update_time	Y	N	N	Y	Y	1000000018
permsup_om_invoice_header	xx_idh_prodlne	Product Line	nvarchar(30)	stage_om_invoice_line	xx_idh_prodlne	Y	N	N	Y	Y	1000000028
permsup_om_invoice_line											

Column Table Source and Transformation

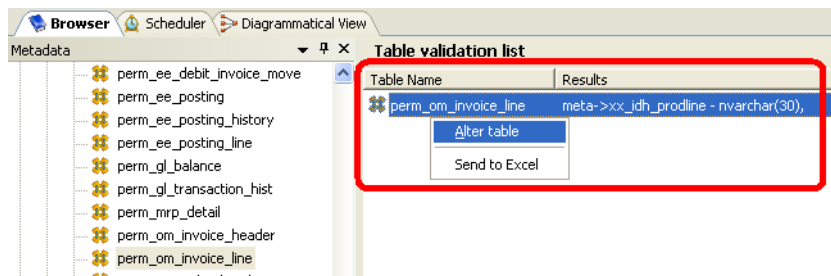
- 6 If this column has been added to a dim_ or a fact_ table and is being updated by a procedure, remove the Source Table from the column definition and put in some dummy value in the column Transformation so that a recompilation of the procedure affiliated with this table does not break.

Modify Table

- 7 Do **not** Create(Recreate) the table. If there is any data in one of these tables, the table needs to be altered to accept the new columns. This is done by right-clicking on the table in question, and selecting the Validate against database option.



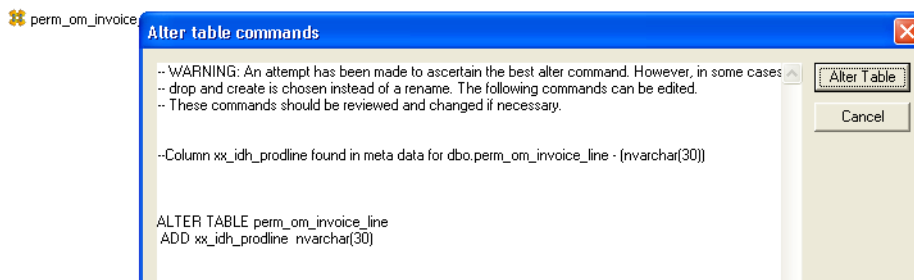
- 8 A list of new columns appears in the Results part of the Table validation list in the Target pane. Right-click on the list and choose the Alter table option.



- 9 An editable list of table alteration commands appears in the window called Alter table commands.

Important It is critical to review the commands listed in this window as sometime it creates commands that drop all the columns in the table. This is particularly true of dim_ tables.

Any excess incorrect commands can be removed. The only commands that should be used here are to ADD columns or possibly modify data types.



- 10 When satisfied with the commands, click Alter Table to make the required additions to the table.

Rebuild the Procedure

- 11 Double-click on the table that has just been modified, and if there is **not** an indicator that the procedure is *MODIFIED* to the right of the Regenerate button, click Rebuild to go through the steps to rebuild the procedure.
- 12 Select the default settings on all three screens to quickly pass through the procedure rebuild process.

Note If any custom code is affiliated with this table, it needs to be modified manually by editing that custom procedure.

Adding Custom Path stage/extract Tables After the perm Tables

If possible and to ensure smooth upgrades, any additional columns beyond the perm table should be added via custom tables. In the Browser Pane, double-click on the Stage Table header to pull up the complete list of stage tables in the Target pane.

Then do the following:

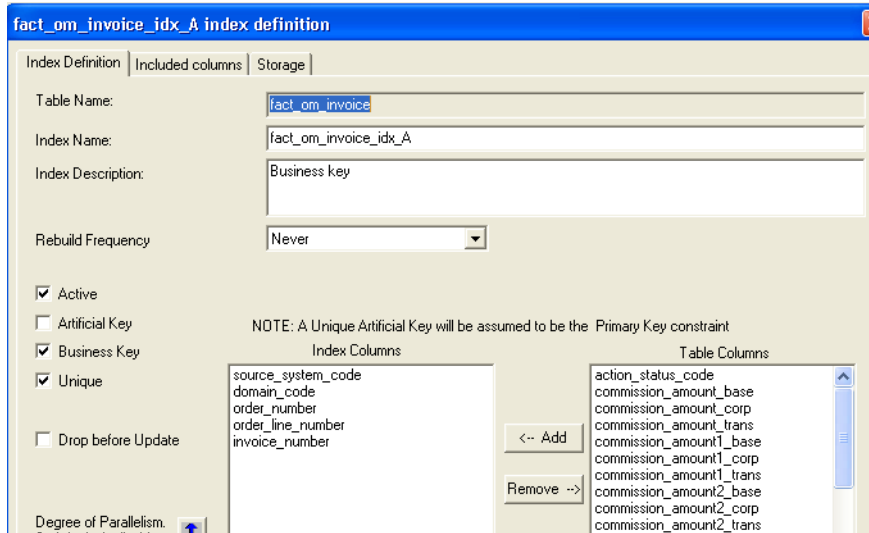
Drag and Drop New Table

- 1 In the source pane, be sure the Data Warehouse connection is open and find the source table for the new parallel table.
- 2 Drag and drop that source table into the Target Pane. A prompt to rename the table appears.
- 3 Rename it to match the table it is supposed to be parallel to, that is, if there is a parallel table.
- 4 Make sure that the Object type is correct. It should be for Object Type Stage Table whether it is a stage or an extract.

Identify and Set the Columns

- 5 Identify the components of the primary key that this table uses to later connect with a dim or fact.
- 6 Delete all columns except those columns, the new custom column, and any critical date columns at the end of the list.

- 7 To identify the primary key components, go to the destination table to be mimicked and Display Indexes.
- 8 Pick the Business key index from the list.



- 9 The end result should be a table with only a handful of columns.

Col name	Display	Data type	Src table	Src column	Format	Nulls	Nu...	Ad...	Att...	Ke...	Order
source_system_code	Source System ...	nvarchar(64)	perm_om_invoice_line	source_system_code		N	N	N	Y		10
domain_code	Domain Code	nvarchar(8)	perm_om_invoice_line	domain_code		N	N	N	Y		20
invoice_number	Invoice Number	nvarchar(80)	perm_om_invoice_line	invoice_number		N	N	N	Y		30
order_number	Order Number	nvarchar(80)	perm_om_invoice_line	order_number		N	N	N	Y		40
order_line_number	Order Line Num...	integer	perm_om_invoice_line	order_line_number	###0	N	Y	Y	N		60
dss_update_time	dss update time	datetime		dss_update_time		Y	N	N	Y		530
xx_idh_proline	Product Line	nvarchar(30)	perm_om_invoice_line	xx_idh_proline		Y	N	N	Y		540

- 10 Create (Recreate) the table. In the Browser pane, right-click on the table name and choose Create(Recreate).

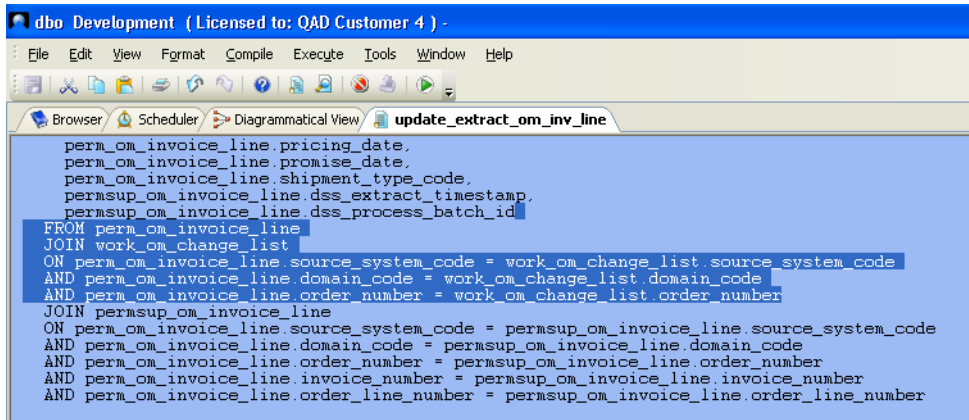
Identify the Code to be Used in the New Procedure

- 11 Go to the parallel table, if there is one, and view its column sources and its procedure code. Note in the column sources below the inclusion of work_om_change_list.

Col name	Display	Data type	Src table	Src column	Format	Null
source_system_code	Source System ...	nvarchar(64)	perm_om_invoice_line	source_system_code		N
domain_code	Domain Code	nvarchar(8)	work_om_change_list	domain_code		N
invoice_number	Invoice Number	nvarchar(80)	perm_om_invoice_line	invoice_number		N
order_number	Order Number	nvarchar(80)	perm_om_invoice_line	order_number		N
order_line_number	Order Line Num...	integer	perm_om_invoice_line	order_line_number	###0	N
item_number	Item Number	nvarchar(30)	perm_om_invoice_line	item_number		Y
invoice_date	Invoice Date	datetime	perm_om_invoice_line	invoice_date		Y

- 12 To view the procedure code Right click on the table name in the Browser pane and choose Code -> View <code name>.

- 13 In the code screen, scroll down to the FROM/WHERE clause of the query, highlight it, and copy it into a notepad. In this instance we do not need the permsup_ table data as that data is passed through in the standard modules.



```

dbo Development ( Licensed to: QAD Customer 4 ) -
File Edit View Format Compile Execute Tools Window Help
Browser Scheduler Diagrammatical View update_extract_om_inv_line
perm_om_invoice_line.pricing_date,
perm_om_invoice_line.promise_date,
perm_om_invoice_line.shipment_type_code,
permsup_om_invoice_line.dss_extract_timestamp,
permsup_om_invoice_line.dss_process_batch_id
FROM perm_om_invoice_line
JOIN work_om_change_list
ON perm_om_invoice_line.source_system_code = work_om_change_list.source_system_code
AND perm_om_invoice_line.domain_code = work_om_change_list.domain_code
AND perm_om_invoice_line.order_number = work_om_change_list.order_number
JOIN permsup_om_invoice_line
ON perm_om_invoice_line.source_system_code = permsup_om_invoice_line.source_system_code
AND perm_om_invoice_line.domain_code = permsup_om_invoice_line.domain_code
AND perm_om_invoice_line.order_number = permsup_om_invoice_line.order_number
AND perm_om_invoice_line.invoice_number = permsup_om_invoice_line.invoice_number
AND perm_om_invoice_line.order_line_number = permsup_om_invoice_line.order_line_number

```

Change the Source Tables

Change any source tables for the columns with which to build the query.

- 14 As was seen in the baseline table, there is an important reference to a work table. We need to have at least one of the columns in this new table defined with a source from work_om_change_list. Each custom table being set up needs to be considered differently based on what the original baseline table does. It is also possible that there may be several baseline tables and the custom path merges all their logic into one table for expediency.

The screenshot displays the 'extractx_om_invoice_line' table definition in the QAD Business Intelligence interface. The 'domain_code' column is highlighted in red, and its properties are shown in the 'Column definition' pane. The 'Source table' is set to 'work_om_change_list'.

Col name	Display	Data type	Src table	Src column
source_system_code	Source System ...	nvarchar(64)	perm_om_invoice_line	source_system_code
domain_code	Domain Code	nvarchar(8)	perm_om_invoice_line	domain_code
invoice_number	Invoice Number	nvarchar(80)	perm_om_invoice_line	invoice_number
order_number	Order Number	nvarchar(80)	perm_om_invoice_line	order_number
order_line_number	Order Line Num...	integer	perm_om_invoice_line	order_line_number
dss_update_time	dss update time	datetime	perm_om_invoice_line	dss_update_time
xx_idh_prodlne	Product Line	nvarchar(30)	perm_om_invoice_line	xx_idh_prodlne

Column definition

Column Properties | Transformation

Table: extractx_om_invoice_line

Column name: domain_code

Business display name: Domain Code

Data type: nvarchar(8)

Format:

Column order: 20 Default Value:

Nulls: ☐

Numeric: ☐ Dimension Join:

Additive: ☐ Key Type (0,A,B,C):

Attribute: ☒ Primary Business Key:

Note: When the Dimension Join flag is set then the contents denotes the foreign key. Business definition denotes the join.

Source table: work_om_change_list

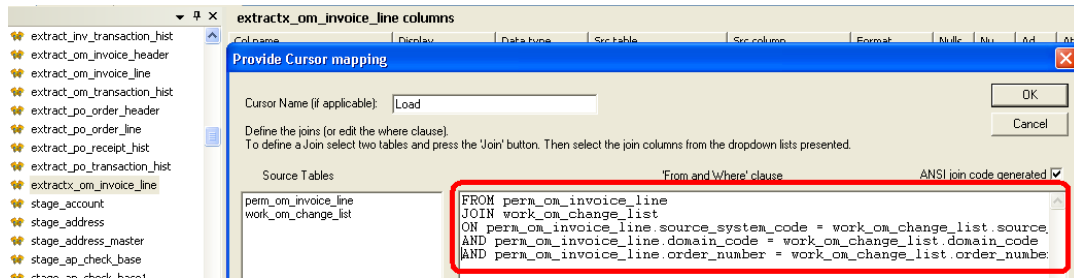
Source column: domain_code

Create (Recreate) the table

- 15 Right click on the table name and pick Create(Recreate). The table should be created.

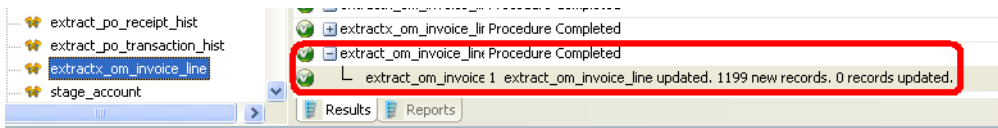
Build the procedure for the custom table

- 16 Double-click the new custom table name in the Browser pane to open up the Properties window.
- 17 Click on Rebuild.
- 18 Follow the standard settings through the first two screens, unless a parameter is required that you would see in the originating table.
- 19 At the third screen, paste in the code copied in "Identify the Code to be Used in the New Procedure" on page 162. Then click OK.



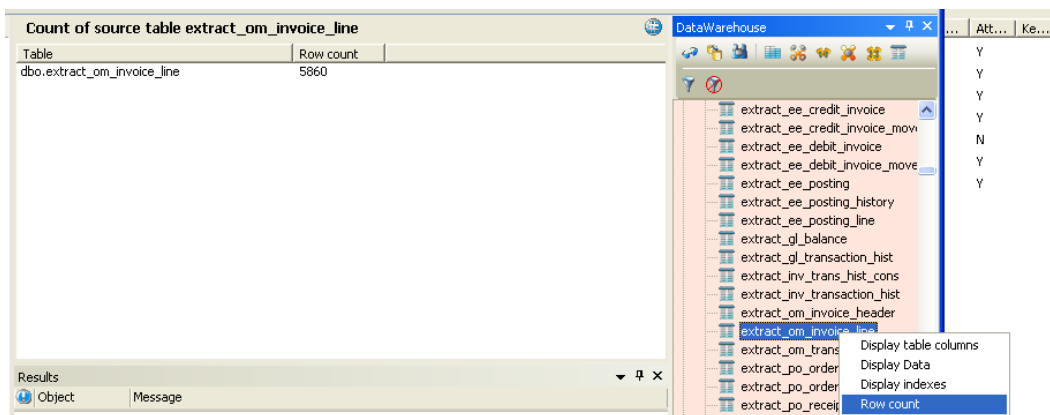
Load the table

- 20** With the new procedure compiled, load the table by right-clicking on it in the browser pane and clicking on the Execute Update Procedure option. Note the results quantity in the Results pane.



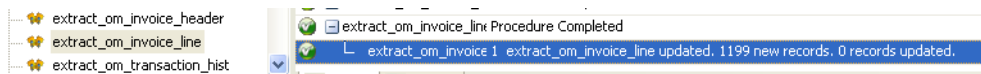
Confirm Results

- 21** Confirm that the number of records matches the correlating original table.
- 22** Refresh the Source pane and right-click on original and select Display table count. That number should match the number just loaded in “Build the procedure for the custom table” on page 164.



- 23** If this is a multi-source system and the results here do not match and, right-click on the original table in the browser and Execute Update Procedure for that as well. Compare the results again. If they match, move on to building the next table.

If they do not match, it you must extract the original procedure query and the parallel procedure query and start working through them in SQL to see what part of the FROM/WHERE clause may be causing the results to differ. This can be due to lack of an OUTER JOIN or the use of an OUTER JOIN or some other table that was not built into the parallel query.



Adding Additional Custom Loading and Staging Tables to Integrate in Additional Columns

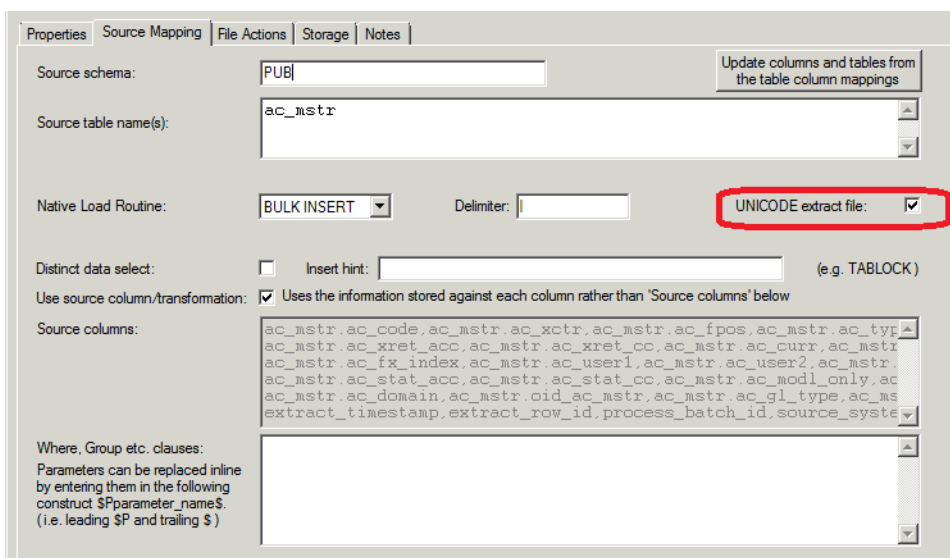
If certain source tables are required by a customer for data that is currently not in the warehouse and we do not currently have matching load tables for these source tables, they can be added and integrated into the warehouse as custom loadx and stagex tables.

If new loadx tables are created, it is critical that:

- All char/varchar columns from the ERP or wherever are *nvarchar* columns in the new loadx table.

And

- In the source mapping for that load table, the Unicode mapping option is checked.



Using those custom tables, one of the staging tables must reference existing staging tables to find a match to the business key of the destination dim or fact table.

The new column is added to the dim or fact table, but it doesn't have a source table or any dummy transformation value set for it.

A post fact or dim table update procedure populates the contents of the new columns, unless for a history table, as per the diagrams.

Additional Customization Topics

- **Using the wizard to re-create a procedure.** Presuming the procedure is not modified, the wizard should be used to regenerate the procedure that loads that table, in the few instances where existing baseline tables have custom columns added to them.

In considering dim and non-history fact tables that have the new custom columns populated by a post-load update procedure, it is not actually necessary to update the existing procedure for that dim/non-history fact.

- **Dealing with custom/modified code.** The column needs to be carefully added to all relevant places in the corresponding Custom/Modified procedures if:
 - In the few instances where existing baseline tables have custom columns added to them, and
 - the correlating load procedures for that table are either Custom or Modified, and
 - the new column needs to be populated by a prior table; for instance history fact tables.
- **Populating new custom columns in *empty* perm/fact tables.** If custom columns have been added to empty perm or fact tables, the tables have been re-created, and their procedures regenerated, these tables should populate normally as part of the historic load. In some instances, there needs to be a custom tables created to mimic the original historic load for the population of fact tables.
- **Populating new custom columns in already populated perm.** In the event that a new custom column needs to be added to an existing already populated perm table, some type of creative update script needs to be devised that can update that perm table for all of the history records it contains; the exception here is the perm_om_invoice_header and perm_om_invoice_line tables, which are currently getting repopulated every day. This is a particularly interesting exercise for the very large tr_hist and gltr_hist and their correlating perm tables, especially if the customer has already archived some of the data in those tables.

Once the data is loaded into the perm tables for those columns, it can be fed on up to the fact tables. Depending on the structure of the customization, this may require a full historic reload of that fact, or just a historic reload of that individual column via the custom path. If at all feasible, try to use the latter option. See Appendix C, “Historically Reloading Modules,” on page 169.
- **Populating new custom columns in populated non-history fact table** where custom columns were added to tables that come after the perm table; column is in perm table, but not tables after. In the case where the new tables added to a fact already existed in the perm table, the ideal first option is to use the new custom path to do a full historic update of the column in question. If that is difficult, option two would be to simply do a historic reload from perm to accomplish this. See Appendix C, “Historically Reloading Modules,” on page 169.
- **Populating new custom columns in populated history fact table** where custom columns were added to tables that come after the perm table; column is in perm table, but not tables after. In the case where the new tables added to a history fact already existed in the perm table, it is likely that a historic reload from perm that includes the new custom path staging tables needs to be done in order to accomplish this. See Appendix C, “Historically Reloading Modules,” on page 169.
- **Cloning load tables to bring in outside data sources (flat files or other databases).** If the data from an outside data source needs to be added to the data warehouse, the customer needs to provide a column map of which tables/columns in their outside data source map to which ERP tables and columns. A clone of the existing load tables can be made and mapped to their outside data source. A post-load procedure would need to be created to insert the data from the clone into the existing baseline load tables. The job sequence would then be to:
 - Truncate the existing load table.

- Run the cloned table load process, which should include the post-load procedure that populates the baseline load table once the cloned load is complete.

All the remaining steps should be essentially the same. There may be a requirement to reload all data in the fact for that given source if the source system for the outside source is not incremental.

- **Post load procedures on cloned tables to populate existing load tables.** The post-load procedure on a cloned load table simply truncates the load table, then it inserts the contents of the cloned table into the baseline load table, mapped as specified per the customer.
- *****Critical*** - Groups/Projects.** If any customization work is to occur in a customer environment, a Group called Custom Work should be set up with two projects:
 - Custom Work – Modified Objects
 - Custom Work – New Objects.

The projects need to be given the extended name, not just Modified Objects or New Objects, because the DWD tool has some issues around confusing same named Projects. Any custom work done in that environment should then be allocated to the appropriate Project that you just created. This ensures that a person can find all custom changes in one place when they get into the environment.

- **Customizations for a pre-domain environment (see specific document).** There is a document outlining the steps for reconfiguring the baseline load tables to work with an eB2 environment. This does not contain load columns and a few other changes. Contact Services for this document. Going this route is OK, but if the customer has an eB2 environment and a post-eB2 environment, making both work is a fairly big effort. It appears to be much more efficient to have a Progress DBA create a schema in the Progress environment where the environment has mapped views of the eB2 tables and all the custom work from the document is instead mapped into the schema views. If you do this, no customizations need be done to the existing load tables. An effort is underway to provide eB2 customers with a Progress package that their DBA just installs. For now, this has only been done at customer sites by the customer's DBA.

Historically Reloading Modules

This chapter describes how to historically reload modules:

Historic Reload Instructions **170**

Historic Reload Instructions

Checklist

The following is a checklist of things to consider and keep in mind when doing a historic reload:

- Is this a complete reload from the ERP or is it a historic reload from the perm tables? A complete load of the ERP is usually required only if there is a problem with the data in the relevant perm tables, such as perm_transaction_history. It is a very time-consuming process, in particular with the large tables like tr_hist or gltr_hist.
- Is the environment being reloaded a single source or a multi-source environment? For multiple data source environments, there are extra considerations that have to be accounted for to ensure that parameters are set properly so that the load tables pick up the data from the right source and label that source properly.
- Is there a need to only load a single fact or do all facts for the module need to be loaded? If all the fact tables need to be reloaded fewer steps are needed because the relative historic load job for that module can be run. If only one table needs to be loaded, it is usually smarter to clone an existing job and remove all the parts relative to loading of other fact tables, leaving in only the things needed for the fact table in question.

Historic Reload from Perm Tables

With the answers to the previous questions in hand and presuming that this is not a complete historic reload, follow these steps to perform a historic reload from the perm tables:

Note If a situation arises that requires a complete historic reload from the ERP of the perm and fact tables, contact QAD customer support. Check back for later releases of this reload guide that may provide more details.

- 1 Set the DAILY_START job to On Hold.
- 2 Set the JOB_CHAINING_ENABLED parameter to N.
- 3 Run the SET_CONNECTION_<datasource1> job from the Scheduler.
- 4 If this is the Inventory module, go to Step 13.
If this is a single source system, go to Step 6.
If this is a multi-source system, go to Step 5.
- 5 Run the following jobs, in sequence, kicking off each after the prior has completed. The reason for this is to populate certain load tables that are needed by the module loads, without trying to guess which ones are populated correctly. This is only applicable if this is a multi-source system because if one source has populated a load table and a different source is being run for the modules, the data will not get loaded correctly. Specific examples of these are load_si_mstr and load_en_mstr, which are referenced by OM, PO, and INV, as well as possibly others.
 - a PREPROCESS_COMMON_TRUNCATE
 - b PREPROCESS_<submodule_symbol>_TRUNCATE
 - c PREPROCESS_COMMON_LOADS

- d PREPROCESS_COMMON_DAILYONLY_LOADS
 - e PREPROCESS_<submodule_symbol>_LOADS
 - f DAILY_COMMON_PROCESS_CHAINED
 - g DAILY_COMMON_PERM_CHAINED
- 6 Run HIST_<submodule>_PERM_EXTRACT
 - 7 If only one fact table needs to be loaded, then go to step 8. If all the fact tables need to be reloaded, just run HIST_<module>_<submodule>_PROCESS_CHAINED. Go to Step 9.
 - 8 If only one fact table needs to be reloaded, such as fact_om_order_history, make a clone of the existing HIST_<module>_<submodule>_PROCESS_CHAINED job. To do this, follow these steps:
 - a Right-click on the job name and choosing Insert Copy of Job.
 - b Click Scheduled at the top of the screen to refresh the job list. Look for the new cloned job that has the same name as the job just copied from but with a number extension as part of the name.
 - c Edit the job name, replacing the number extension to something like _ONE_TIME_RELOAD. For instance the job might be named HIST_OM_PROCESS_ORD_HIST_ONETIME_RELOAD.
 - d Edit the Task list of the job. In the example of only needing to reload fact_om_order_history, remove all the tasks that are not relevant to the loading of that table, such as the stage and fact table for fact_om_booking, fact_om_shipping, fact_om_invoice. Make sure to leave jobs in the stage_om_order_hist_revalue that are tied to the fact_om_order_history table. If it is not clear which tasks should be removed, it might be easiest to just remove the non-relevant fact tables from the list so that the historic reload is not trying to repopulate them unnecessarily.
 - e Run the newly created one time job.
 - f Go to step 10.
 - 9 Assuming all facts need to be loaded, then just run the HIST_<module>_<submodule>_PROCESS_CHAINED job.
 - 10 Repeat steps 3-9 for all data sources.
 - 11 Run the POSTPROCESS_<module>_<submodule>_<xxx>_SUMMARY to populate the snapshot and performance tables.
 - 12 Set the DAILY_START parameter back to the regularly scheduled time.
 - 13 For the Inventory modules, set the Parameter INV_PROCESS_DAYS to 50000. Let the Daily run normally. This may cause the run to take much longer than normal. Once the daily run has completed, set the INV_PROCESS_DAYS parameter back to some much lower number like 5 or 10.

Warning This methodology will ensure that the Inventory module takes longer than normal for a few days until it passes the reset INV_PROCESS_DAYS parameter again.

