Installation Guide

QAD Business Intelligence

78-0956-3.11-Rev1
QAD Business Intelligence 3.11
September 2015

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 © 2015 by QAD Inc.

BIPortal_IG_v0311.pdf/biw/biw

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

Contents

QAD Business Intelligence Install Guide Change Summary ix

Chapter 1 Installing QAD BI 1

Pre-Install Information 2
Installation Packages 2
SQL Server Notes 2
Requirements 2
Configuration Parameters 3
Installing BI 3
Installing QAD BI Mobile App on the iPad 9
Installing BI Modules 10
Installing the Common Module 10
Installing Additional BI Modules 11
Configuring BI Modules 12
Initial Parameter Setup 12
Initial Job Setup 12
OM Module Setup 14
TM Module Setup 16
Progress ODBC Connection Setup 17
Install Metadata Language Translation Files 18
Creating and Processing Cubes 20
Configuring the BI Portal 21
Install Standard Dashboards and Reports 21
Fonts Required to Display Reports Properly 21
Technical Documentation for the Data Warehouse 22
Installing Excel Workbooks 22
Customize BI Reports Logo 23
Note on Password Encryption in the BI Portal 23
Customizing the BI Portal Logo 24
QAD Standard CFO Dashboard 25
Modifying Existing Installations 28
Add New Source to Existing Installation 28
Adding New Module to Existing Installation 29
Auto-restart Set Up 29

Chapter 2 Reference 31

Jobs 32
Reference Tables 34
Common Module 35
Financials Module 51
Order Management Module 61
Operations Module 65
Enterprise Asset Management Module 72
Transportation Management Module 76
Service and Support Management Module 78
System Information 79
Parameter Values that Need to Be Set Before History Load 80

Chapter 3 QAD BI Upgrade Instructions. 85

Upgrade Information 86
Upgrade to QAD BI 3.11 from 3.10 86
Upgrade Steps for QAD BI Data Warehouse Designer 86
BI Portal Upgrade Procedure 91
New and Modified Tables 92
Upgrade to QAD BI 3.10 from QAD BI 3.9 102
Upgrade Steps for QAD BI Metadata 102
BI Portal Upgrade Procedure 105
Data Warehouse Tables Changed 106
Upgrade to QAD BI 3.9 from QAD BI 3.8.1/3.8.2 108
Upgrade Steps for QAD BI Metadata 108
BI Portal Upgrade Procedure 110
Data Warehouse Tables Changed 112
Upgrade to QAD BI 3.8.1 from QAD BI 3.7 115
Upgrade Steps for QAD BI Metadata 115
Data Warehouse Tables Changed 118
Upgrade to QAD BI 3.7 from QAD BI 3.6.x 121
Preparation 121
Upgrade Steps for QAD BI Data Warehouse Designer 121
BI Portal Upgrade Procedure 122
Upgrade Steps for QAD BI Metadata 122
Data Warehouse Tables Changed 126
Upgrade to QAD BI 3.6 from QAD BI 3.5 129
BI Metadata Upgrade Procedure 129
BI Portal Upgrade Procedure 133
Data Warehouse Tables Changed 134
Upgrade to QAD BI 3.5 from QAD BI 3.4.1 138

Chapter 4 Onsite Data Validation 145

Introduction 146
EAM Validation Reports 146
EAM Equipment Downtime Transaction Table 146
EAM Inventory Transaction Table 146
EAM Labor Transaction Table 147
Financials EE Accounts Payable Validation Reports 147
EE AP Supplier Invoice Fact Table 147
EE AP Supplier Invoice History Fact Table 147
EE AP Supplier Invoice Snapshot Fact Table 148
Financials EE Accounts Receivable Validation Reports 148
EE AR Customer Invoice Fact Table 148
EE AR Customer Invoice History Fact Table 148
EE AR Customer Invoice Snapshot Fact Table 149
Financials EE General Ledger Validation Reports 149
EE GL Balance Fact Table 149
EE GL Transaction Fact Table 149
CFO Dashboard Fact Table 149
Financials General Ledger Validation Reports 150
GL Transaction Validation 150
GL Trial Balance Validation 150
Financials SE Accounts Receivable Validation Reports 150
SE AR Invoice Fact Table 150
SE AR Invoice Fact Table 150
Financials Accounts Payable Validation Reports 151
AP Voucher Fact Table 151
AP Voucher Fact Table 151
Inventory Validation Reports 151
Inventory Balances Fact Monthly Snapshot Table 151
Inventory Balances Fact Monthly Snapshot Table 152
Overall Equipment Efficiency Reports 152
Work Order Validation 152
Work Order Routing Validation 152
Operation Transaction Validation 153
Order Management Validation Reports 153
Compares OM Booking Fact Table 153
OM Booking Fact for Scheduled Orders Table 153
OM Shipping Fact Table 154
OM Invoice Fact Table 154
OM Order History Fact Table 154
OM History Fact for Scheduled Orders Table 154
Purchasing Validation Reports 155
Purchase Order History Fact Table 155

Purchase Order Receipt Fact Table 155
Automated Order Management Validation 155
Running Validation of BI Against ERP Reports 156
Standard Edition – Invoice History 156

Appendix A Sizing Guidelines 163
Introduction 164
Important Restrictions and Caveats on BI Sizing 164
Business Intelligence Deployment Options and Terminology 164
BI Terminology 165
BI Deployment Options 166
Sizing Terminology Used Throughout this Sizing Section 167
Sizing Methodology for QAD BI 167
CPU Sizing 168
CPU Sizing for DWD/Scheduler Component 168
CPU Sizing for Concurrent User Workload (Interactive Portal Users) 168
RAM Sizing for DWD/Scheduler Component 168
CPU Sizing for BI Portal WebApp (Server Side) 169
RAM Sizing for Concurrent Users (Interactive Portal Users) 169
Disk and IO Subsystem 169
Note on Importance of Appropriate Disk and IO Subsystem Sizing . . 169
IO Subsystem Sizing 169
Key IO Subsystem Sizing Notes 170
Notes on Database Disk Sizing 171
Network Considerations 171
Network Recommendations 172
Example Sizing 172
Reference: Estimation of Initial Database Size 173
Dependence of Initial Database Size on BI Modules 173
Estimation of Initial BI Data Warehouse Size 174
Allowance for Future Database Growth 176
Reference: Network Sizing Details 176
Terminology 176
Network Sizing Considerations - Database to BI Database Server . . . 177
Network Sizing Considerations - Remote Client to BI Portal 178
WAN Considerations 178

Appendix B Customization Guidelines 179
Overview 180
Customization Challenges 180
QAD BI 3.x Customization Methodologies 183
Custom Path Methodology (Preferred) 183
Integrated Methodology (Last Resort) 185

Customization Needs for Additional Columns 185
Use Track Back Diagram to Check Tables 186
Double Checking the Data Path with a Column Track Back Report . . 188
Notes on Tables 190
Identifying Every Appearance of a Column in Existing Code 192
Naming Conventions for Custom Changes to Packaged QAD BI Modules . . . 192
Custom Tables 193
Custom Columns Joined to Standard Module 193
Custom Columns Not Joined to the Baseline Module Tables 193
Guidelines for Creating Custom Path Custom Tables 193
Load Tables 193
Perm Tables and Pre-perm Stage Tables 194
All Other Stage, Extract 194
Adding Required xx_ Prefixed Columns 194
Adding Additional Columns to a Load Table 195
Get Column from Data Source 195
Naming/Column Placement/Data Type 195
Transform 196
Create (Recreate) the Table 196
Adding Additional Custom Columns to Pre-perm Stage Tables 197
Connect 197
Transformation/Business Name 198
Create (Recreate) the Staging Table 198
Adding Additional Custom Columns to perm/dim/fact Tables 199
Data Source 199
Placement 199
Column Table Source and Transformation 199
Modify Table 200
Rebuild the Procedure 201
Adding Custom Path Stage/Extract Tables After the Perm Tables 201
Drag and Drop New Table 201
Identify and Set the Columns 201
Identify the Code to Use in the New Procedure 202
Change the Source Tables 203
Create (Recreate) the Table 204
Build the Procedure for the Custom Table 204
Load the Table 205
Confirm Results 205
Adding Additional Custom Loading and Staging Tables to Integrate in Additional Columns 206
Additional Customization Topics 206

Appendix C Historically Reloading Modules 209

Historic Reload Instructions 210
Checklist 210
Historic Reload from Perm Tables 210

Appendix D Manual Installation 213
Introduction 214
Manual Installation Requirements 214
Manually Installing the Data Warehouse 214
Installation of DWD on Separate Server than SQL Server 222
Accessing Defined Work Directory for the DWD Connection 222
Setting Up a Split Server System 222
Troubleshooting the Split Server System 225
Manually Installing the QAD BI Analytical Modules 226
Requirements 226
Installation Steps 226
Install BI Analytical Modules 227
Manually Installing the QAD BI Web Application 232
Installation Steps 232
Install BI Portal Web Application 233
Getting Started in the BI Portal Web Application 235

Product Information Resources 237

QAD Business Intelligence Install Guide Change Summary

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

Date/Version	Description	Reference
September 2015/QAD BI 3.11-Rev 1	Numerous editorial changes	---
March 2015/QAD BI 3.11	Numerous editorial changes	---
	Revised pre-installation Requirements	page 2
	Added Auto-restart Set Up section	page 29
	Added and revised Common Module parameters	page 35
	Added GLTR_HIST_LOOKBACK_DAYS to Financials Module parameters	page 51
	Revised INV_BALANCE_HIST_START_DATE Operations Module parameter	page 65
	Added load_eam_employee_erp_table_name to Enterprise Asset Management Module parameters	page 72
	Revised Parameter Values that Need to Be Set Before History Load section	page 80
	Added BI 3.10 to 3.11 Upgrade Instructions	page 86
	Added section Manual Installation Requirements	page 214
Sept 2014/QAD BI 3.10 Rev 1	Added OM Module setup steps to BI Installation chapter	page 14
	Corrected report names in Onsite Data Validation chapter: GL Trial Balance Validation and Operation Transaction Validation	page 150 page 153
	Revised Split Server Installation section	page 222
March 2014/QAD BI 3.10	Made various changes to the Ch 2 Reference (Parameter tables)	page 31
	Added Parameter table for SSM Module	page 78
	Added the BI 3.10 upgrade procedure	page 102
	Added Ch 4 - Onsite Data Validation	page 145
Oct 2013/QAD BI 3.9 Rev 1	Added the Order Management - Scheduled Orders to the Ch. 3	page 112
Sept 2013/QAD BI 3.9	Added GUI Installation instructions to Chapter 1	page 3
	Added 3.9 Upgrade procedure	page 108
	Moved manual installation instructions to Appendix D	page 213
	Combined Chapters 1, 2, and 3 into Chapter 1: Install QAD BI	page 1
	Made changes to the CPU Sizing section in Appendix A: Sizing	page 168
April 2013/QAD BI 3.8.1 Rev 1	Fixed abbreviation of Operations Module (OP) throughout manual	various

Date/Version	Description	Reference
	In the Initial Job Setup section, removed step 32 that stated the following: Some modules require some special settings for the History load as opposed to the Daily load. For instance, set GL_LOAD_DAY, ...	page 12
	In the parameter table, deleted the explanation/notes from GL_LOAD_DAYS and GL_LOAD_LOOKBACK_DAYS.	page 51
March 2013/QAD BI 3.8.1	Editorial review; rebranded for QAD BI 3.8.1	--
November 2012/QAD BI 3.8	No updates made	--
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	page 209
March 2012/QAD BI 3.6	Provided instructions on upgrading from previous version	--
	Added sizing guide	page 163
	Added customization guide	page 179
September 2011/QAD BI 3.5	Rebranded for QAD BI 3.5	--

Chapter 1

Installing QAD BI

This chapter describes how to install QAD BI. It discusses the following topics:

Pre-Install Information 2

Installing BI 3

Installing BI Modules 10

Configuring BI Modules 12

Configuring the BI Portal 21

Modifying Existing Installations 28

Pre-Install Information

To save time during the installation, record the following customer information in advance:

- See Chapter 2, “Reference,” for information.
- Use Table 2.9 on page 79 for the customer’s System information.
- Use Table 2.10 on page 80 for the customer’s Parameter information.

Installation Packages

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

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 If the installation requires that the DWD be on a different server than the database server, contact QAD customer support for information.

SQL Server Notes

- Configure the SQL Server for both SQL Server and Windows authentication.
- Set the Recovery Model for the Data Warehouse databases to Simple.
- Evaluate the default autogrowth settings for the data warehouse database. If the database is handling large quantities of data, increase the settings from the defaults.
- Ensure that there is plenty of space for tempdb. When the Data Warehouse processes large amounts of data, it sometimes uses a large amount of tempdb space for calculations. If necessary, move the tempdb data and log files to a larger drive.
- As with any database product, always create an appropriate database backup and restore plan.

Requirements

- Java 6 JDK or higher.
- SQL Server 2008 or 2012 installed and configured.
- SQL Server has TCP enabled, to connect via JDBC.
- **Arial Unicode MS fonts.** Unicode values in reports now export properly to PDF and Excel. This requires the existence of Arial Unicode MS fonts on the BI Portal server. If this font is not available on the Tomcat server, a different Unicode font must be provided. In this case, modify the font-extensions file (in the Portal web application's WEB-INF\classes directory) to reference the preferred Unicode font.

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

Configuration Parameters

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

- **SQL Server JDBC URL.** The URL required to connect to the SQL Server database (host name, port number, and database name).
- **QAD BI Data Warehouse Designer user / pwd.** The User name and password to connect to the SQL Server Database where QAD BI DWD is running.

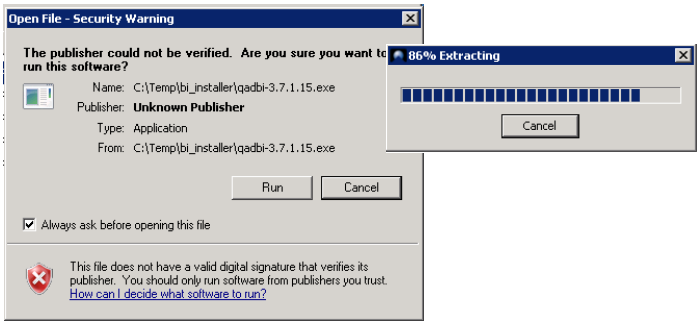
  Note This user must have “dbo” (owner) privileges to create the necessary data model security on the Data Warehouse database.
- **BI Portal database User / pwd.** The user name and password to connect to the SQL Server database where BI Portal repository is installed.

Installing BI

1 Navigate to the directory where the QAD BI Installer file was downloaded and double-click and launch the QAD BI Installer.

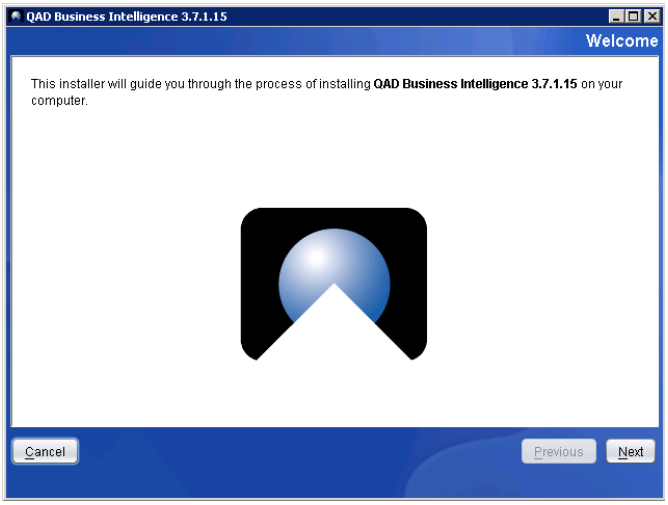
2 In the Open File window, click Run.

Fig. 1.1
Installing BI



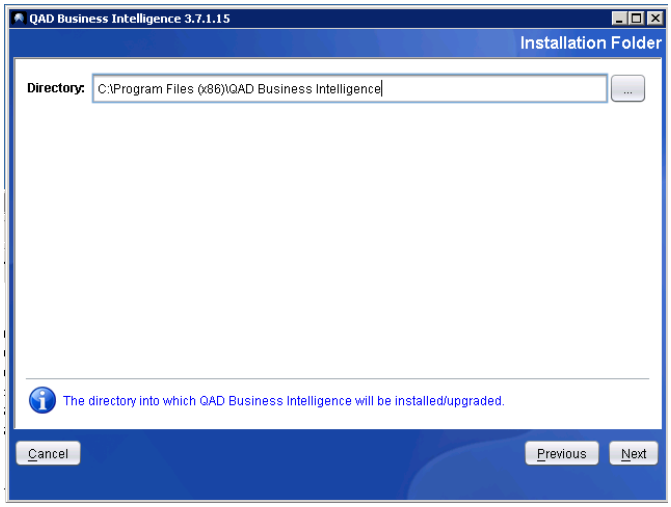
3 Click Next.

Fig. 1.2
Installing BI



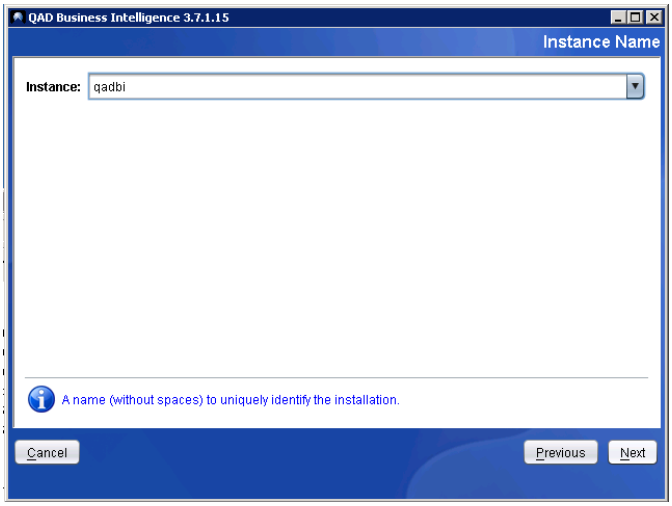
4 Choose the correct directory to install BI or accept the default location. Click Next.

Fig. 1.3
Installing BI



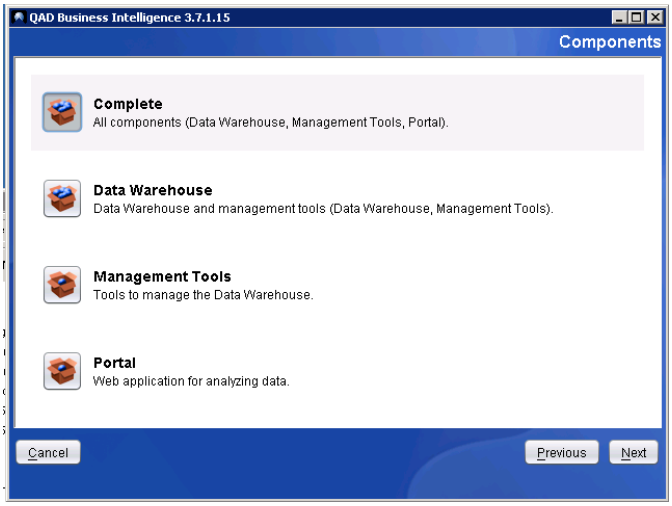
5 Pick the name of the database or use the default. Click Next.

Fig. 1.4
Installing BI



6 To install everything, select Complete. Click Next.

Fig. 1.5
Installing BI



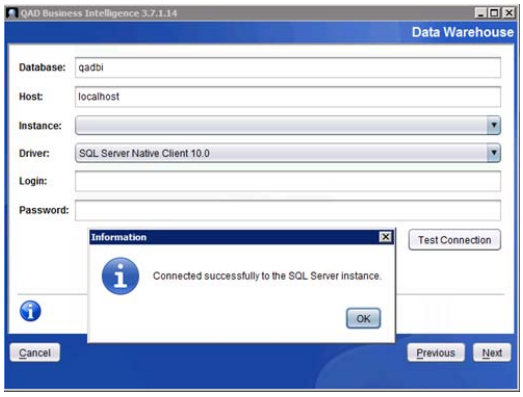
7 Pick the name of the database or use the default. Click Next.

8 Enter the database host and instance name.

Note QAD recommends that you leave the Login and Password fields blank to use Windows authentication when connecting to the database. Otherwise, enter a SQL Server login and password.

9 Click Test Connection to validate the database connection. Once confirmed, click Next.

Fig. 1.6
Installing BI



10 You need the following information about the customer's Progress databases.

Example

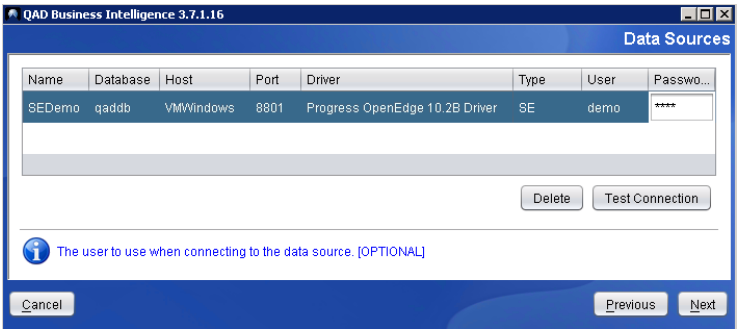
Required Information for Progress Database	**EE**	**SE**
Connection Name	EE_QADDB	SE_QADDB
Database	qaddb	mfgprod
Host	localhost	localhost
Port	20214	20214
Driver		
Type (EE, SE, TM, or EAM)	EE	EE
User ID	qaddb	qaddb
Password	demo	demo

11 Define the ODBC connections to the Progress databases by entering the following:

- **Name.** Enter the distinct name of the ODBC connection.

  Note: If you are setting up multiple data warehouses that source the same Progress databases, include a defining name extension for each ODBC Name, such as SEDemoDev, SEDemoTest, and SEDemoProd. When multiple data warehouses are set up on the same server, the installer does not allow you to use the same ODBC connection name, even if the source is the same.
- **Database**. Enter the source database name.
- **Host**. Enter the host name.
- **Port**. Enter the port number.
- **Driver**. Select the correct driver. For Progress database sources, select the Progress OpenEdge driver, which appears in the drop-down menu.
- **Type**. Select the type. For example, select whether this is for an SE or EE data source or for TM or EAM.
- **User**. Enter the username.
- **Password**. Enter the password.

Fig. 1.7
Installing BI



12 Test the database connection that you set up by clicking Test Connection.

Note If the connection test does not work, confirm the following:

- The database name, host, and user/passwords are spelled correctly.
- The port number is correct.
- The Progress database is currently running and it has a working SQL broker running on the specified port.

13 Repeat the process for every Progress data source being added.

Note You can add more non-Progress data sources later using the DWD Admin tool.

14 Enter the DWD License Name and Number. A specific license name and number for the installation is provided as part of the BI purchase.

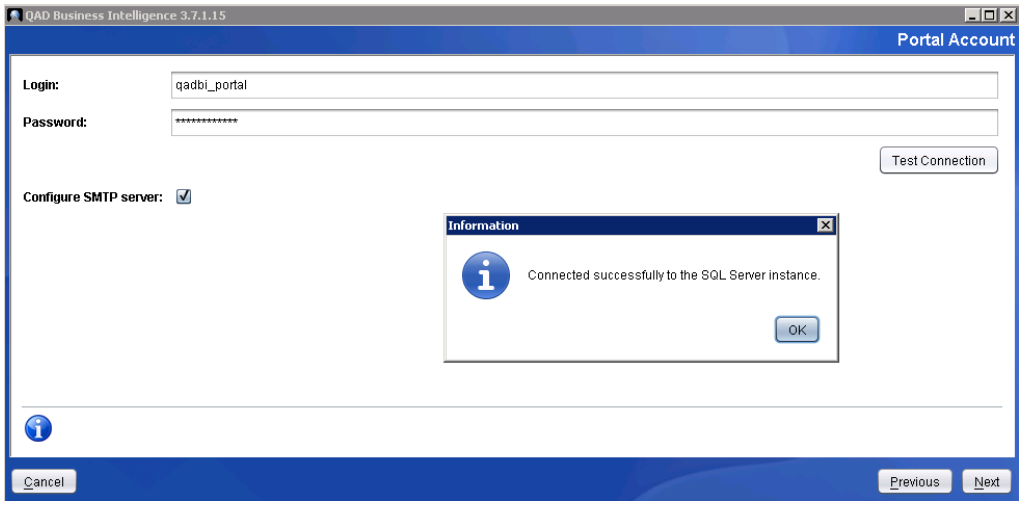
15 Enter setup Portal Login and Password. This information is for the SQL Server login that was set up earlier.

Note The Portal requires a SQL Server login and password. It cannot use Windows authentication to connect to the Portal database.

16 Select the Configure SMTP Server check box. Selecting this check box ensures that e-mails can be sent from the Portal.

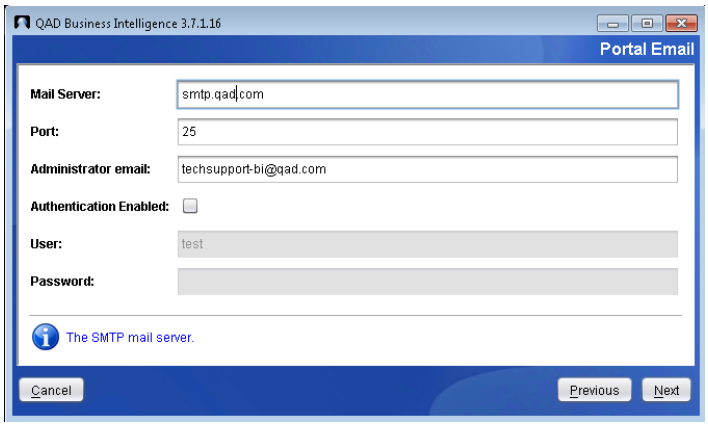
17 Click Test Connection to verify that the connection works.

Fig. 1.8
Installing BI



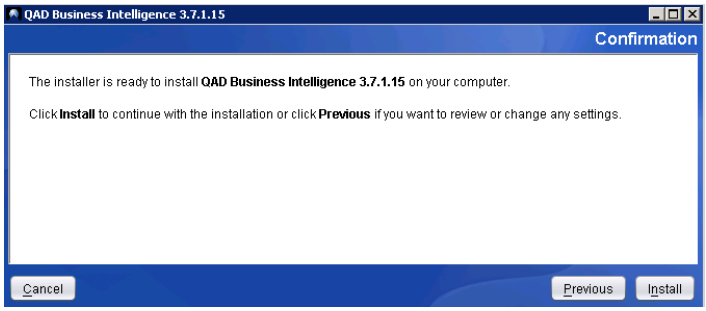
18 Enter setup SMTP Server information.

Fig. 1.9
Installing BI



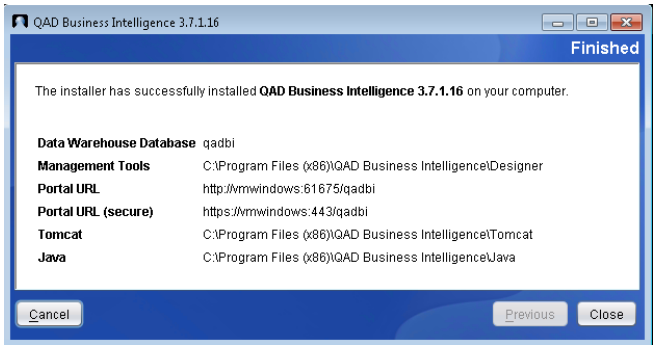
19 Click Install to create database.

Fig. 1.10
Installing BI



20 Once the installation is complete, a message appears indicating that the installation was successful. Click Close.

Fig. 1.11
Installing BI



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 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 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 the Save Password On/Off button and turn it on.
6. Click the check mark. You can now log in to BI from your iPad.

Installing BI Modules

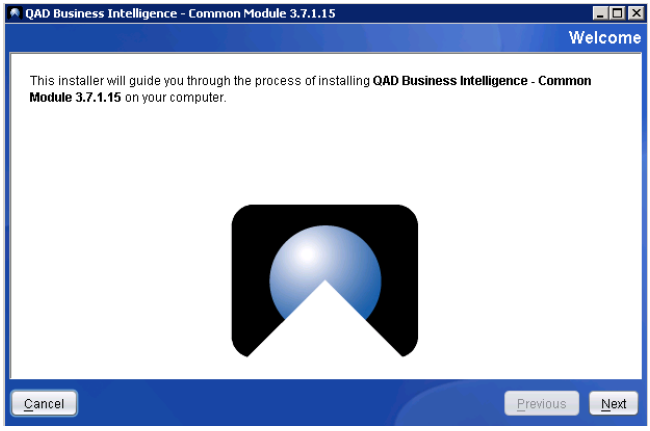
Important Install the Common module before any other module except the Transportation Management module. The TM module is the only module that does not require the Common module.

Installing the Common Module

Follow these steps to install the Common Module:

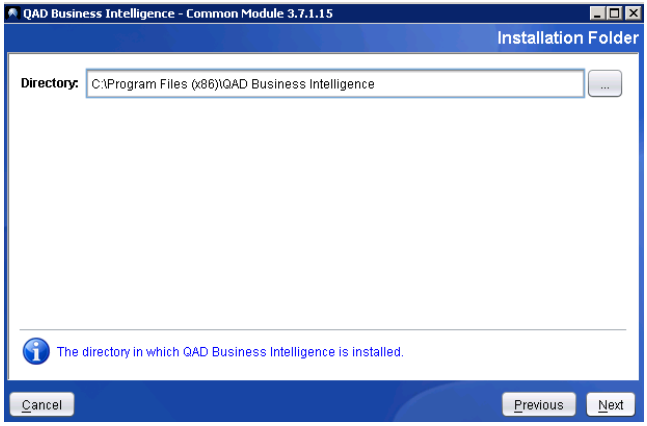
1. Double-click the `common.exe` file.
2. Click Next.

Fig. 1.12
Installing BI Modules



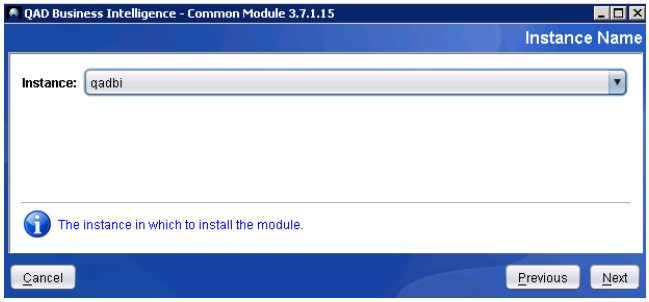
3. Choose the appropriate directory to install the Common Module in or accept the default location. Click Next.

Fig. 1.13
Installing BI Modules



4. Choose the database instance or accept the default. Click Next.

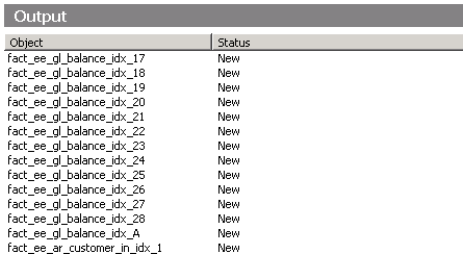
Fig. 1.14
Installing BI Modules



5 The installer starts.

Note The installation takes a while to load. If the screen says Not Responding, do not close it. Wait until it finishes loading the Common module components.

Fig. 1.15
Installing BI Modules

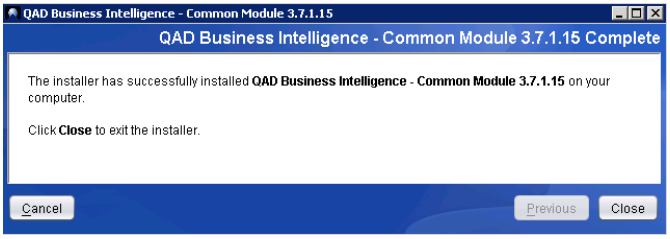


Output

Object	Status
fact_ee_gl_balance_idx_17	New
fact_ee_gl_balance_idx_18	New
fact_ee_gl_balance_idx_19	New
fact_ee_gl_balance_idx_20	New
fact_ee_gl_balance_idx_21	New
fact_ee_gl_balance_idx_22	New
fact_ee_gl_balance_idx_23	New
fact_ee_gl_balance_idx_24	New
fact_ee_gl_balance_idx_25	New
fact_ee_gl_balance_idx_26	New
fact_ee_gl_balance_idx_27	New
fact_ee_gl_balance_idx_28	New
fact_ee_gl_balance_idx_A	New
fact_ee_ar_customer_in_idx_1	New

6 Once the installation is complete, a message appears indicating that the installation was successful. Click Close to exit.

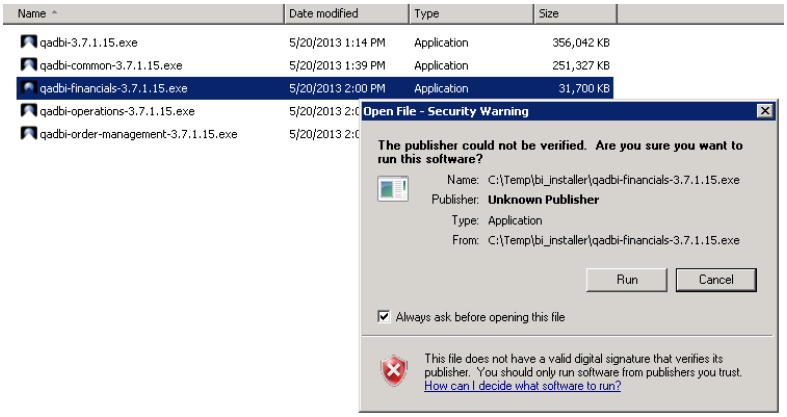
Fig. 1.16
Installing BI Modules



Installing Additional BI Modules

To install additional BI modules, double-click the correct `.exe` file and then repeat the steps in “Installing the Common Module” on page 10. Continue repeating the same steps until all required modules are installed.

Fig. 1.17
Installing Additional BI Modules



Configuring BI Modules

Initial Parameter Setup

Note In QAD BI DWD, choose Parameters from the Tools menu to display the current parameter definitions. Customize the parameter definitions for your specific installation.

Set runtime parameters to the proper settings based on the customer feedback to the parameters.

Note Parameters that MUST be set before running INITIAL_JOB_SETUP have an initial value of `<some value between these symbols>`. See Chapter 2, “Reference,” on page 31 for more details.

Initial Job Setup

Note The INITIAL_JOB_SETUP job creates the proper chained job flow for all of the source connections. This is not used for the Transportation Management (TM) module. For more information, see “TM Module Setup” on page 16.

1. Set the INITIAL_JOB_SETUP_CONNECTION_01 parameter to your first QAD ERP source.
2. 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.
3. 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 of 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 and separate them with commas. List the Common module 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 Asset Management Module – EAM sources only
- **SSM.** For the Service and Support Management module, ERP sources.

4 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 rerun INITIAL_JOB_SETUP. This method creates a Historical job flow that loads just the new module, so it is not necessary to repeat any historical loads for other modules.

5 Set INITIAL_JOB_SETUP_CONNECTION_01_VERSION to the version of your source system. This parameter is the major release year. For example, if your source is 2010 SE, set the version parameter to 2010; if your source is 2011.1 EE, set the version parameter to 2011. Do not include the .1.

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

7 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.

8 You can continue adding as many INITIAL_JOB_SETUP_CONNECTION_XX* and parameters as necessary. Make sure that you add all of 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
- INITIAL_JOB_SETUP_CONNECTION_XX_VERSION

9 Eliminate the values for the connections that are not being used.

10 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 on the job name and start the job.

e The status changes to Waiting.

f Refresh the screen until the job completes.

g If the job fails to start, check the scheduler status.

11 After INITIAL_JOB_SETUP runs, check Scheduled jobs to confirm that you now have SET_CONNECTION jobs with names that end with all connections you have created. If they were not created, check the Parameters. Verify that the INITIAL_JOB_SETUP_ENABLED parameter is set to Y and run INITIAL_JOB_SETUP again.

12 Verify that your parameters are all set properly for your installed modules. See Table 2.10 on page 80.

13 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 can result in errors in the history load, requiring the reloading of GL history.

14 Ensure that your ERP sources all have their GL Calendars configured properly, both historically and into the future, for all domains. In addition to Financials, several of the BI modules, including Inventory and Order Management, use the GL Calendar to determine financial periods.

15 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 process can take quite some time. It does not hurt to check in throughout this load process to verify that the jobs are all running successfully.

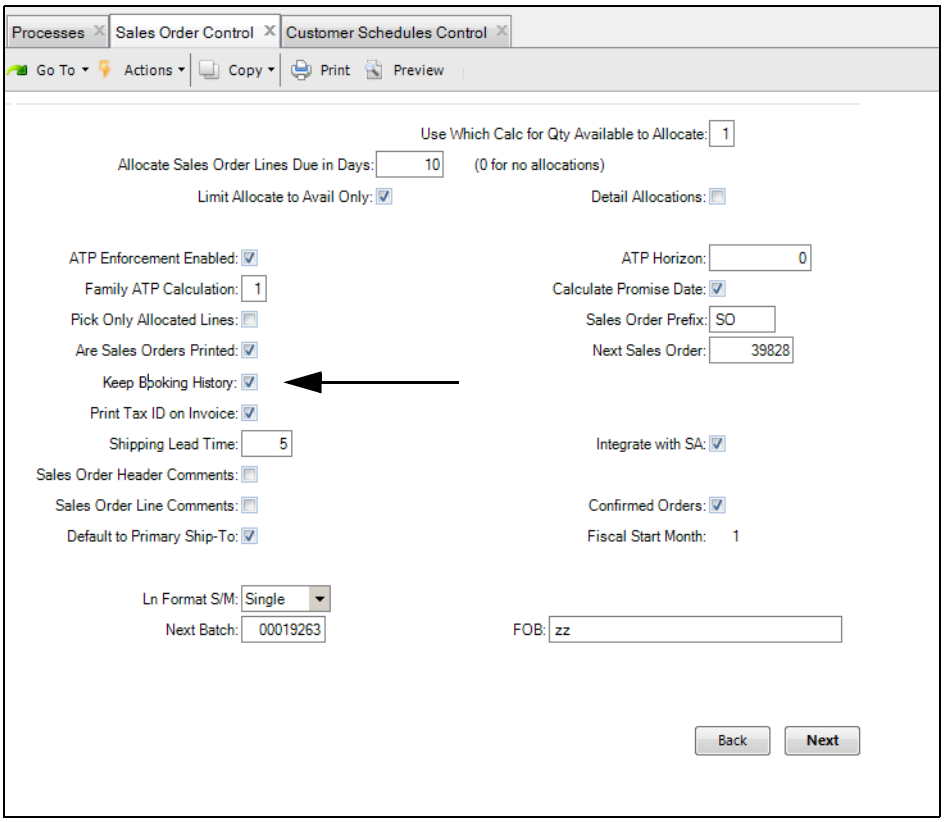
OM Module Setup

If the OM Module is installed, enable two settings in QAD Enterprise Applications (QAD EA) if you want ERP to create booking and scheduled order data to import into BI:

- **Keep Booking History in Sales Order Control.** When the Keep Booking History check box is selected (the default value), the system keeps an audit trail of all incoming discrete sales orders. These orders include new orders added or existing orders that were changed or deleted. This audit trail is stored in transaction history with record type ORD-SO. BI requires the ORD-SO records and uses them to generate booking, sales order history, and sales order performance transactions. If this check box is not selected, the system does not create ORD-SO records and there are no booking and related records for discrete sales orders in BI.

  To enable this setting, open QAD EA and open Sales Order Control (7.1.24). Select the Keep Booking History check box and then save.

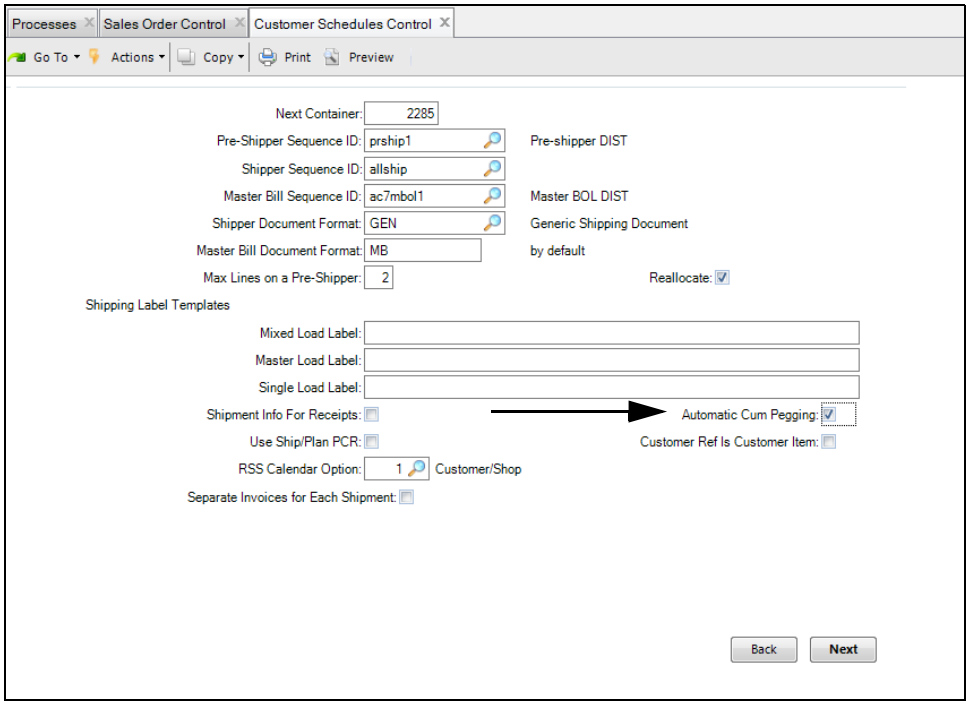
Fig. 1.18
OM Module Setup: Keep Booking History



- **Automatic Cum Pegging in Customer Schedules Control.** When Automatic Cum Pegging is selected (the default value), the system automatically links shipper lines to lines on the required ship schedule. BI uses this pegging information when generating shipment, order history, and order performance transactions for customer scheduled orders. If this check box is not selected, the system does not create pegging records. There are also incomplete shipment, order history, and order performance transactions for customer scheduled orders in BI.

  To enable this setting, open QAD EA and open Customer Schedules Control (7.3.24). Select the Automatic Cum Pegging check box and then save.

Fig. 1.19
OM Module Setup: Automatic Cum Pegging



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, preset 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 TM 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 the TMS source. Enclose each type in single quotes, with commas separating the types; 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 a specific source system.
- **TM_LOAD_LOOKBACK_DAYS.** Set this parameter to a large number for the initial data extract. The number must 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 parameter to a smaller value. Using a smaller value limits each daily load to just the period in which there can 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 TM system is based in Progress or Oracle, run the TM_ConvertSource job. Set these parameters before running the job:

- **TM_SourceConvert_SchemaORA.** If your source is in Oracle, enter the schema name. For Progress databases, the schema is assumed to be *PUB*.

4 **TM_SourceConvert_TargetSource.** Enter *oracle* or *progress*.

Progress ODBC Connection Setup

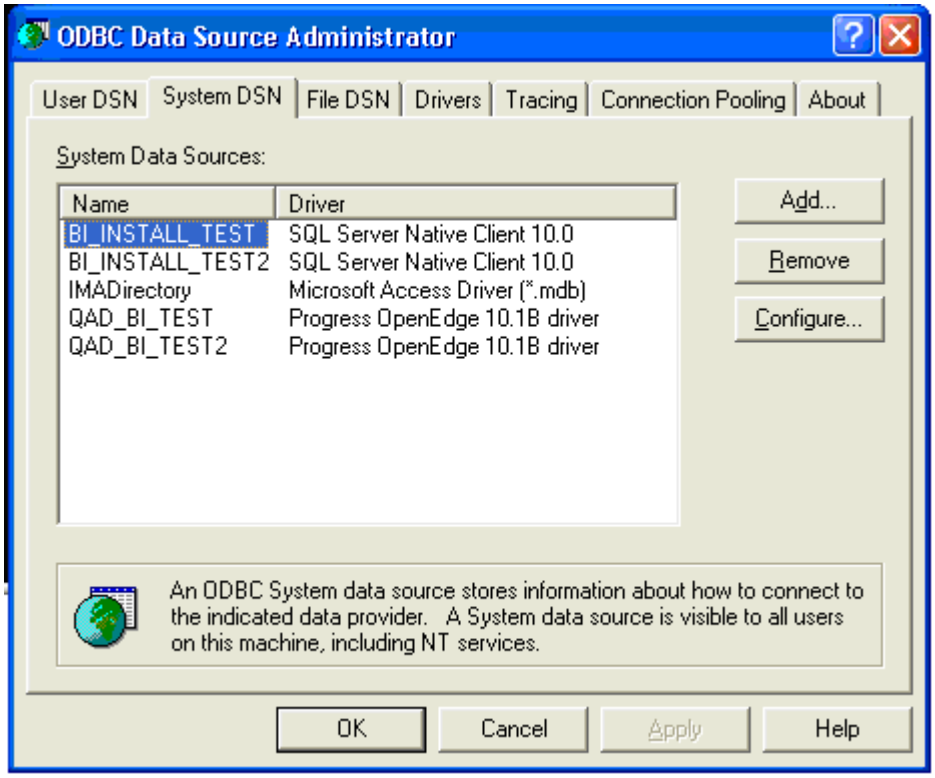
Follow these steps if you need to create ODBC DSNs outside of the GUI Installer:

1 If the Progress ODBC connects 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.

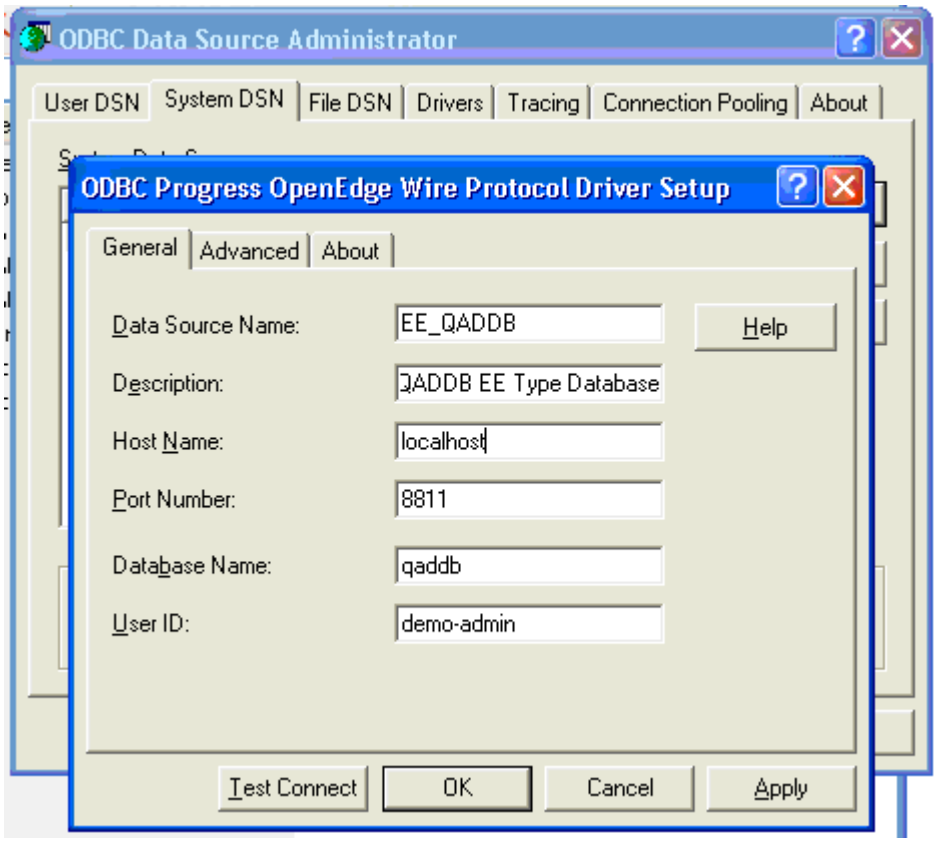
Fig. 1.20
Create ODBC Progress



b Pick the driver type Progress OpenEdge 10.1B driver and click Finish.

c Input the values provided to establish the connection to the Progress database.

Fig. 1.21
Create ODBC Progress



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.

2 If the client requires it, set up any additional ODBC connections to outside data sources. You can do this step later if the information is not available.

Install Metadata Language Translation Files

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

1 In the BI DWD tool, choose Tools|Options.

2 Click the Languages tab and add the following codes for the language files you want to install:

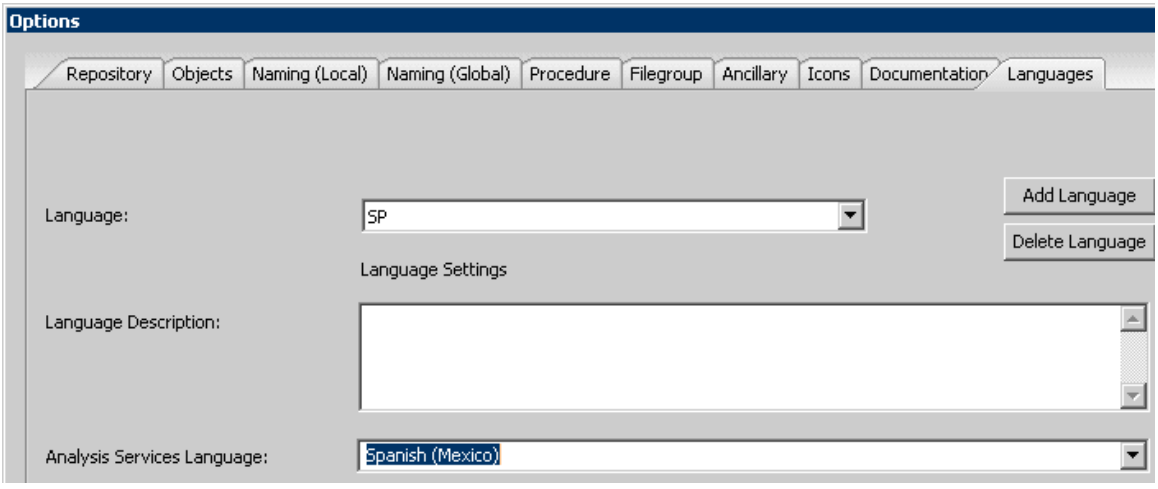
Language	Code
Czech	cs
German	de
Spanish	es
Spanish (Mexico)	es_MX
French	fr
Italian	it

Language	Code
Japanese	ja
Korean	ko
Dutch	nl
Polish	pl
Portuguese (Brazilian)	pt_BR
Chinese (Simplified)	zh
Chinese (Traditional)	zh_TW

3 The complete set of files is 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 must 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 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 scenario happens, click the Analysis Services Language drop-down list, scroll down, and select the Spanish option.

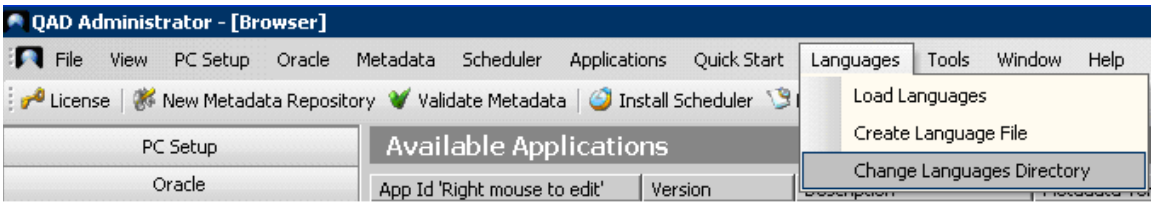
Fig. 1.22
Language Install



4 After adding the language codes, launch the QAD BI Data Warehouse Setup Administrator.

5 Select Languages|Change Languages Directory. Navigate to your package's metadata/translations directory.

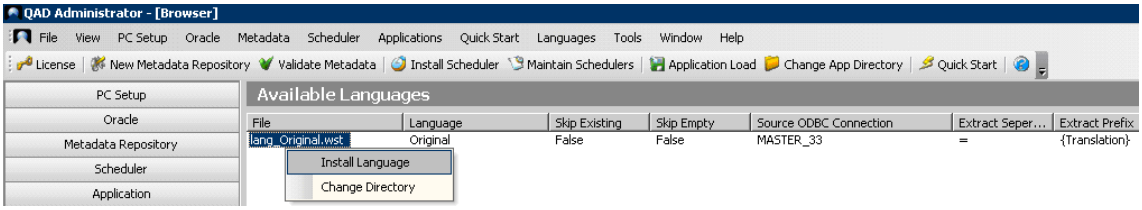
Fig. 1.23
Language Install 2



6 Choose Languages|Load Languages. Right-click on the language file you want to install and choose Install Language.

7 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 2.

Fig. 1.24
Language Install 3



8 Repeat these steps for each language you want to install.

Creating and Processing Cubes

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

1 Some of your analytical modules come with cube metadata already created. To use these cubes, you must first create them in SSAS.

2 Verify that the DataWarehouse connection is configured properly for use as a cube source. In the DWD, right-click on the DataWarehouse connection and choose Properties. Verify that the OLAP Data Source section is completed correctly:

- The Provider Type is set to SQLOLEDB.
- The Server and Database fields contain 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.

3 Locate the cube in the Metadata Browser. Right-click on the cube and choose Properties.

4 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.

5 Use the default Cube Database name or optionally create a different name for the SSAS database. Click OK to save.

6 Right-click on the cube and choose Create (Alter) Cube.

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

7 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 daily jobs update the fact tables.

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 you can process using the *Create (Alter) Cube* action in the Data Warehouse without changing SSAS first.
- Which modifications require the user to delete the SSAS cube database before re-creating the cube from the Data Warehouse.

Configuring the BI Portal

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.

Fonts Required to Display Reports Properly

If the Arial Unicode font is not installed on the BI Portal server, reports do not display properly. Because Arial Unicode is a Microsoft font, if you install any Microsoft product, such as Excel or Word, the application typically installs this font.

If you do not want to install a Microsoft product, you must to point to a different font file that exists on the BI Portal server. This server is the machine where Tomcat and the BI Portal web application are installed.

To point to a different font file, you must change the `fonts1284146005640.xml` file in WEB-INF\classes and modify the path to the `ARIALUNI.TTF` file in C:\Windows\Fonts.

If this font is not Unicode, and your BI portal users have language settings for languages that include Unicode characters, those characters do not display properly in reports.

Technical Documentation for the Data Warehouse

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.

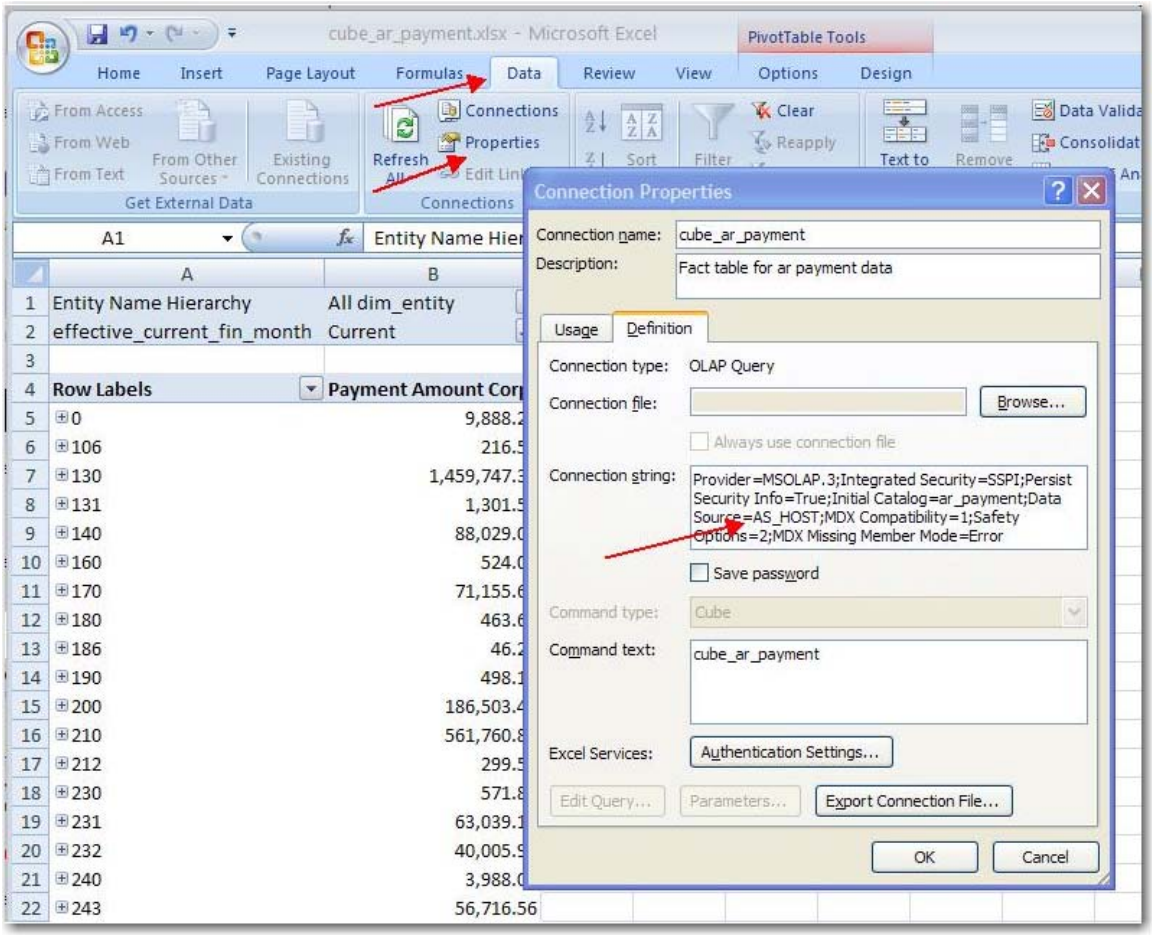
Installing Excel Workbooks

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 Web application directory, `<tomcat home>/webapps/<application name>/published`. Extract the file `<module_name>_adhoc_excel_files.zip` into that directory. The standard dashboards reference these files.

Each workbook uses a data source connection to the Analysis Services system. Because the connections used are embedded in the worksheets, modify them to connect to the local Analysis Services host. To do this modification, 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.

Full documentation on this process is available from Microsoft.

Fig. 1.25
Excel Workbooks



Customize BI Reports Logo

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.

Restart Apache Tomcat through the Services window.

Note on Password Encryption in the BI Portal

You can encrypt the SQL Server password in two places. The first is the `server-config.xml` file, which contains the connection information for the BI Portal database. The second is 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 tool, 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 verify that pwdEncrypted=true. If you change back to using an unencrypted password value, make sure that you set pwdEncrypted=false.

Similarly, if you want to encrypt the password the Portal uses 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 the encrypted value. Make sure that you add pwdEncrypted=true.

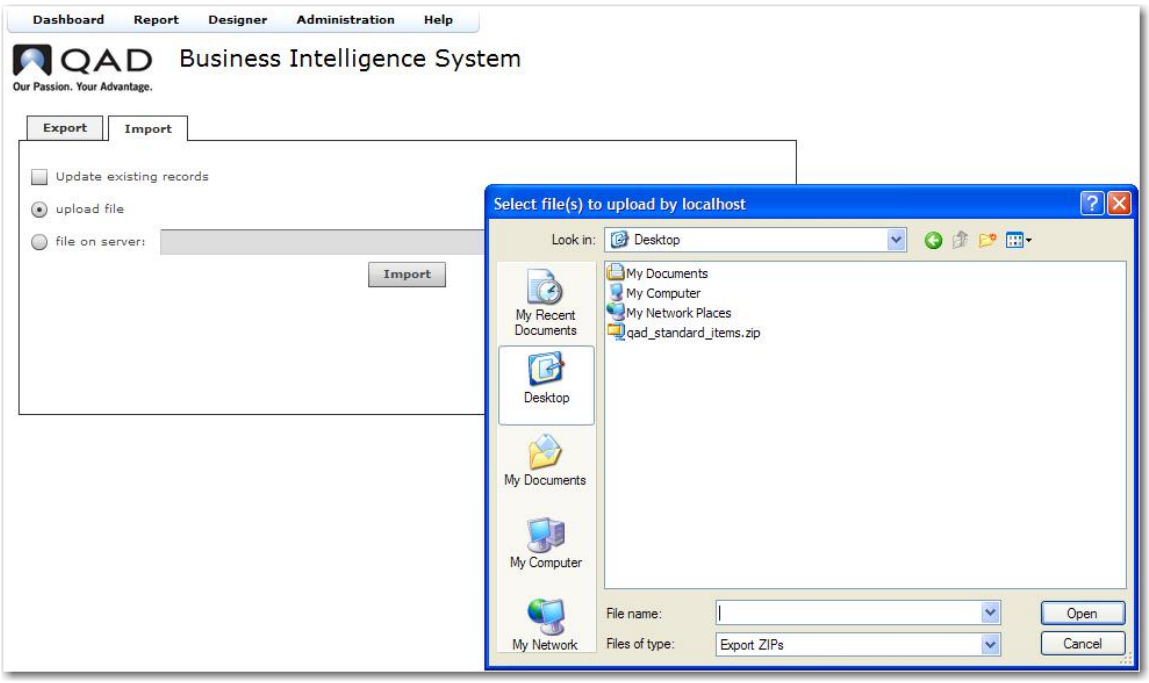
If you change either password, stop and restart Tomcat for the change to take effect.

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 logo 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`.

Fig. 1.26
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, log out and log in again to see them.

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, you can roll up report lines into groups and perform computations on the groups to generate the 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. The workbook 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 highest level of information to display. This page lists the name of the report sets that the CFO Dashboard describes. One set is given by default: CFO_METRICS. You can include multiple reports, or report versions, in the same spreadsheet.
- **Report Lines.** This tab defines the lines of each report. Report lines can be labels (which 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 are not summaries of accounts, but more complicated combinations of the report lines.
- **Report Calculations**. Allows the user to combine the lines of the report arbitrarily to create derived metrics.

These tabs are described in more detail in the next sections. Before configuring the CFO Dashboard, however, 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 are the average number of days that a payment takes. You can compute the debtor days metric on any account by dividing the total amount in accounts receivable by net revenue received. Then multiply the result by the number of days in the fiscal year. Furthermore, compute this metric by region so that you can see the measures for the North American and Pacific Rim regions.

To compute 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 does not change. You can collect this report in the set for the standard CFO_METRICS.

If the system calls for numerous reports that need dividing at the highest level, add another element to the reports tab for that highest-level grouping. That new element is then carried over to the Report Lines tab.

Report Lines Tab

This tab holds the high-level groupings on which the computations in the reports are performed. In this specific example, you must define four lines: 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 name 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 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. Set Indent_level to 0 for a label, and 1 otherwise. To make items visible to the end user, enter Y in the visible column.

All computations on a report line are done on the value of the report. The report computations are additive and you must subtract values of one line from another. Therefore, you can specify that a line is a negative value in the report lines tab by placing a Y in the change_sign column.

In this specific example, subtract the line of net revenues 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 putting a Y in the change_sign column. Change the sign for the North American and Pacific Rim Net Received lines.

Report Line Ranges 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. Enter the report line name (CFO_METRICS) 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 that are 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 QAD_PACRIM. The example tab continues to define the account ranges and source systems for all five data lines of the CFO_METRIC report.

Report Line Calculations Tab

This tab defines the computations that are performed on the report lines as defined on the Report Lines tab. You can determine the desired higher-order metrics by combining the report lines.

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
```

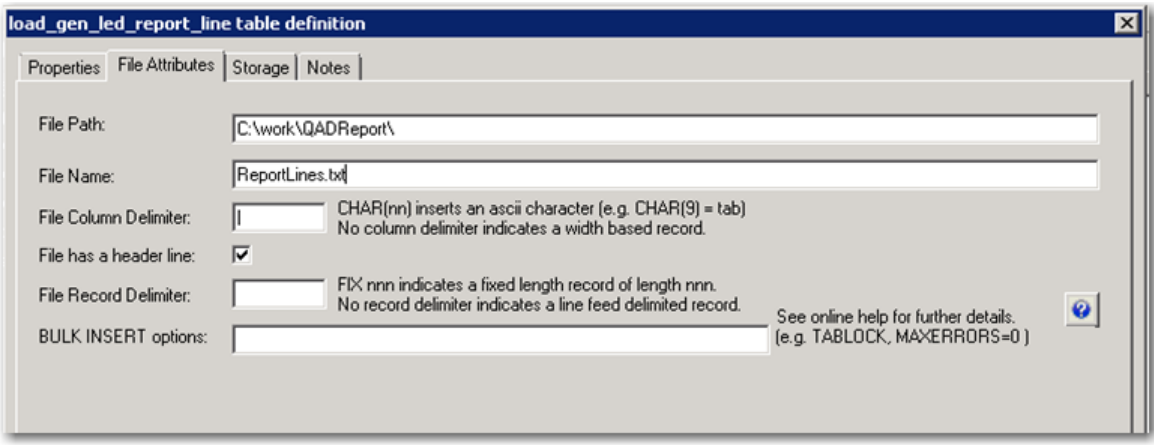
Providing the Workbook Data to the Data Warehouse

Once the data is entered into the workbook as described, translate the data 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. Place this section of cells into a text file, and move the text file to a location specified in the data warehouse.

Find the location to place the text files by double-clicking the load tables in QAD BI DWD and noting the directory location in the File Path field. By default, the location is set to `c:\work\QADReport`, but you can change this path.

Fig. 1.27
File Path



The files to use, 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 in the data warehouse and the CFO Dashboard contents are displayed. The daily jobs are usually run overnight.

Modifying Existing Installations

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 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_SETUP_CONNECTION 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
- INITIAL_JOB_SETUP_CONNECTION_XX_VERSION

4. Verify that all other INITIAL_JOB_SETUP_CONNECTION_XX_RUN parameters for the source are set to Y so that their history loads are not 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 is 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 must add a module and load the data for all your sources:

1. Install the new module. See "Installing Additional BI Modules" on page 11.
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 must be "COM,OM,FIN". If you want to add the Operations module, set the HIST_MODULES value to "COM,OP".

   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,OP".
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.

Auto-restart Set Up

The auto-restart feature allows you to check for failures during Daily processing and to automatically restart failed jobs.

How It Works

Tasks were added to the start and finish jobs that cause the DAILY_START job to continue running as the subsequent job chain progresses. It looks every five minutes for any failed jobs. If a failed job exists, it restarts the failed job. When the DAILY_RUN_COMPLETE parameter is set to Y by the end procedure, the auto-restart procedure finishes.

The auto-restart task attempts to restarts the failed job a specified number of times before giving up. The number of restart attempts is set in the configurable parameter JOB_FAILURE_NUMBER_OF_RESTARTS.

Set Up

The JOB_FAILURE_NUMBER_OF_RESTARTS default value is 0, which disables auto-restart. To use the auto-restart feature as part of the daily job chain, set the JOB_FAILURE_NUMBER_OF_RESTARTS parameter to a non-zero value.

Chapter 2

Reference

This chapter provides reference information:

Jobs ***32***

Reference Tables ***34***

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 before beginning a new load. This job type ensures that load jobs start 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, they are loaded in a job named PREPROCESS_[MODULE]_HISTONLY_LOADS or PREPROCESS_[MODULE]_DAILYONLY_LOADS, respectively. Also, there can 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 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 must 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 can be an _EE or _SE after the [MODULE] if the job applies to one type of ERP source only.

Extract

This set of jobs builds extract and work tables from the perm tables. Not all of the data in the perm tables is typically extracted at one time, because much of the data in the perm tables was already processed, and has not changed. So usually a subset of data, covering a certain period and/or 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 can be an _EE or _SE after the [MODULE] if the job applies to one type of ERP source only. 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 takes 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 can be an _EE or _SE after the [MODULE] if the job applies to one type of ERP source only.

Additionally, some modules can 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 that are already populated. 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 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 can 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 automatically installed in the job chain, you can add these jobs to the Daily job chain.

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 TR_HIST jobs use the following parameters:

Important Do not update them manually. The HIST_LOAD jobs use these to track the data that 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:

- Table 2.1, “Parameters for the Common Module,” on page 35
- Table 2.2, “Parameters for Various Module,” on page 51
- Table 2.3, “Parameters for the Financials Module,” on page 51
- Table 2.4, “Parameters for the Order Management Module,” on page 61
- Table 2.5, “Parameters for the Operations Module,” on page 65
- Table 2.6, “Parameters for the Enterprise Asset Management Module,” on page 72
- Table 2.7, “Parameters for the Transportation Management Module,” on page 76
- Table 2.8, “Parameters for the Service and Support Management Module,” on page 78
- Table 2.9, “System Information,” on page 79
- Table 2.10, “Parameter Values that Need to Be Set Before History Load,” on page 80

Common Module

Table 2.1
Parameters for the Common Module (Page 1 of 16)

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 must be 60 days to search for any backdated payments.		COM	M
COM_ADD_CLEANUP_JOBS	N	Y	Y - CLEANUP jobs added to HIST LOAD job list to minimize size of data warehouse; N - Let stage tables keep debug trail	Most customers should leave it the default Y. The N setting is for saving staging data during a historical load in case something runs improperly and needs debugging.		
COM_MULTI_SOURCE_STAGING	N	Y	Y/N Initial Job Setup should process one source at a time (N) or all sources at once (Y)	When set to Y, SE and EE source types are staged together in one set of process jobs in the DAILY scheduled job run. The HIST jobs, the pre-process jobs, and other source types like EAM are unaffected.		
AR_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name the AR process job uses.		COM	P
COM_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name the common process job uses.		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 2.1
Parameters for the Common Module (Page 2 of 16)

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 do not update this parameter 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
COM_RUN_LOAD_OP_HIST	N	Y	Determines if load table is populated based on which modules are installed. Set by Pre-Load SQL in table load_op_hist.	Helps determine if a given module exists for common objects that only are needed by a few modules and have large quantities of data.	COM	P
COM_RUN_LOAD_TR_HIST	N	Y	Determines if load table is populated based on which modules are installed. Set by Pre-Load SQL in table load_tr_hist, load_tr_hist_list and load_tr_hist_history.	Helps determine if a given module exists for common objects that only are needed by a few modules and have large quantities of data.	COM	P
COM_RUN_LOAD_TR_HIST_REMARKS	N	Y	Determines if load table is populated based on which modules are installed. Set by Pre-Load SQL in table load_tr_hist_remarks.	Helps determine if a given module exists for common objects that only are needed by a few modules and have large quantities of data.	COM	P

Table 2.1
Parameters for the Common Module (Page 3 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
COM_RUN_LOAD_WO_MSTR	N	Y	Determines if load table is populated based on which modules are installed. Set by Pre-Load SQL in table load_wo_mstr and load_wo_mstr_hist	Helps determine if a given module exists for common objects that only are needed by a few modules and have large quantities of data.	COM	P
COM_RUN_LOAD_WOD_DET	N	Y	Determines if load table is populated based on which modules are installed. Set by Pre-Load SQL in table load_wod_det and load_wod_det_hist	Helps determine if a given module exists for common objects that only are needed by a few modules and have large quantities of data.	COM	P
COM_RUN_LOAD_WR_ROUTE	N	Y	Determines if load table is populated based on which modules are installed. Set by Pre-Load SQL in table load_wr_route and load_wr_route_his	Helps determine if a given module exists for common objects that only are needed by a few modules and have large quantities of data.	COM	P
CORP_CALENDAR_DOMAIN	<Enter domain here for corporate calendar setup>	10USA	The Domain Code used for the corporate financial calendar	This source and domain's calendar are used to set the various "corp" fields in dim_date_qad.	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		COM	M
CORP_CURRENCY_CODE	<Enter three-letter currency code>	USD	Corporate currency code used for high-level reporting	Amounts that are in transaction and/or base currency for the domain are also converted to this Corporate currency, so that reporting can occur in a common currency. The value entered here must be a three-letter currency code that exists in dim_currency (which is sourced from PUB.cu_mstr).	COM	M

Table 2.1
Parameters for the Common Module (Page 4 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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
DAILY_HISTORY_DATE_SET_CURRENT	Y	Y	Default Y indicates that jobs always set the SALES and PO HISTORY DATE to the current date. If there is a need to run loads using a date before 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 are set to the 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_CONNECTION_QMI	Inserted via WsParameterWrite		COM	P
DAILY_LOAD_JOB0002 - DAILY_LOAD_JOBxxxx	Does not exist until INITIAL_JOB_SETUP is run; jobs and their order determined by INITIAL_JOB_SETUP 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

Table 2.1
Parameters for the Common Module (Page 5 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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
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_CONNECTION_NAME	Added in 3.7; for future use	COM	M
DAILY_RUN_COMPLETE	N	Y	Indicates if the daily run has run to completion.	Used by the autostart task, which looks for failures and restarts a job if it fails. Tasks need to be added to DAILY_START (admin_check_for_and_restart_failed_jobs) and DAILY_MODULES_LOAD_FINISH (end_daily_load_chained task)	COM	P
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

Table 2.1
Parameters for the Common Module (Page 6 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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
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_FIRST_DATE_FACT	1900-01-01	2013-01-01	Filter date for fact tables, reducing the data loaded to a fact to no sooner than this date. Should always be newer than HIST_FIRST_DATE_PERM and ideally at least one year newer.	Change at the direction of QAD only. Used for BI Lite. Currently only tested with Common and Order Management modules; do not use with any other modules.	COM	M
HIST_FIRST_DATE_PERM	1900-01-01	2011-01-01	Filter date for load tables populating perm tables to reduce overall size of initial data load to only records after this date.	Change at the direction of QAD only. Used for BI Lite. Currently only tested with Common and Order Management modules; do not use with any other modules.	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

Table 2.1
Parameters for the Common Module (Page 7 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
HIST_LOAD_JOB0001		SET_CONNECTION_QMI	Inserted via WsParameterWrite		COM	P
HIST_LOAD_JOB0002 - HIST_LOAD_JOBxxxx	Does not exist until INITIAL_JOB_SETUP is run; INITIAL_JOB_SETUP job determines jobs and their order; 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
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

Table 2.1
Parameters for the Common Module (Page 8 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INITIAL_JOB_SETUP_CONNECTION_01_RUN	N	Y	This parameter determines if the Initial Job Setup has been run against this data source. 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 rerun 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, SSM	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. The 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, SSM	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. The default is fine for most installations as any uninstalled modules are skipped.	COM	M
INITIAL_JOB_SETUP_CONNECTION_01_VERSION		2011	Version - specify version of the QAD Application, to skip tables in BI that do not exist in older versions. Example: 2012	Only specify the whole number value of the release version (for example, 2010, 2011, 2012, 2013); do not include any point values (for example, do NOT enter 2011.1 – only enter 2011)	COM	M
INITIAL_JOB_SETUP_CONNECTION_02		2010SE	Name of the connection 02 used as a source system for cloning jobs.		COM	M

Table 2.1
Parameters for the Common Module (Page 9 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
INITIAL_JOB_SETUP_CONNECTION_02_RUN		Y	This parameter determines if the Initial Job Setup has been run against this data source. 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 rerun 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, SSM	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. The 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, SSM	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. The default is fine for most installations as any uninstalled modules are skipped.	COM	M
INITIAL_JOB_SETUP_CONNECTION_02_VERSION		2011	Version - specify version of the QAD Application, to skip tables in BI that do not exist in older versions Example: 2012	Only specify the whole number value of the release version (for example, 2010, 2011, 2012, 2013); do not include any point values (for example do NOT enter 2011.1 – only enter 2011)	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 2.1
Parameters for the Common Module (Page 10 of 16)

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 most of the previously manually populated date fields for job setup.		COM	M
INITIAL_JOB_SETUP_ENABLED	Y	N	Indicates if 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.	If this flag is set to "N", the INITIAL_JOB_SETUP job does not do anything. If set to "Y," running the INITIAL_JOB_SETUP job creates 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_ XXXXXXXX	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 2.1
Parameters for the Common Module (Page 11 of 16)

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
JOB_FAILURE_NUMBER_OF_RESTARTS	0	3	Number of times the admin_check_for_and_restart_failed jobs fails on a job before giving up. Set to 0 to disable auto-restart. If there may be a system down for a long time, set the number higher, like 20 to give more time before giving up.	Used for auto-restart task	COM	M
JOB_PROCESS_RUNNING_TYPE		D	Values: H, D, {blank} - Shows current job chain type, (H)IST or (D)AILY	Shows which scheduled job chain is currently or has most recently ran. Used to control the job chain procedure. Set by DAILY_START and HIST_START so no need to user edit.	COM	Do Not Edit
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

Table 2.1
Parameters for the Common Module (Page 12 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
OP_HIST_LOAD_MAXSIZE	5000000	5000000	The maximum number of rows to load in an iteration loading op_hist		COM	P
OP_HIST_LOAD_RUNNING	C	C	Indicates if a load of op_hist is running (Y/N/C)		COM	P
OP_HIST_TRAN_MAX		19975	DO NOT UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		COM	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)	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	DO NOT 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
TR_HIST_DATE_MAX_SXX		2012-08-08	DO NOT 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

Table 2.1
Parameters for the Common Module (Page 13 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TR_HIST_DOMAIN		'AUTO','QP','QPAU','QPCAN','QPMEX','train'	DO NOT 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	DO NOT 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 parameter is the maximum number allowed for the tr_hist.tr_trnbr field, or in other words, the number after which the tr_trnbr rolls 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 2.1
Parameters for the Common Module (Page 14 of 16)

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 parameter 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 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	5000	Number of records that would typically be in three days worth of data to ensure readiness for tr_hist.tr_trnbr rollover.	TR_HIST processing in the common module takes longer when looking for the TR_HIST transaction number to roll back to all zeros, but this number should not be smaller than the number of transactions performed every 2-3 days by a customer. We estimate 5000.	COM	M

Table 2.1
Parameters for the Common Module (Page 15 of 16)

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 parameter 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 to find the new, lower-numbered transactions. 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 fails and indicates that this number must be increased. The default value is 3.	COM	M
TR_HIST_REMARKS_LOOKBACK_DAYS	60	60	The number of days to look back from ih_hist to ensure any recent changes to tr_hist records get picked up.	This parameter was used from SALES_PROCESS_LOOKBACK_DAYS but the load_tr_hist_remarks table has now become Common for use by OM and SSM.	COM	M
TR_HIST_ROW_COUNT	0	24186	DO NOT 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

Table 2.1
Parameters for the Common Module (Page 16 of 16)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
TR_HIST_TRAN_MAX		2134	DO NOT UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		COM	P
TR_HIST_TRAN_MAX_SXX	Does not exist until after a load of source's data	27971	DO NOT UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		COM	P
TR_HIST_TRAN_MIN	-1	-1	DO NOT 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 2.2
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 2.3
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 process which covers up to 60 days backdated.		FIN	M
AP_EE_HISTORY_DATE		20081231	The date from which the AP aging table is refreshed. 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 refreshes from. Three years back is common. For example, 20081231	Added 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

Table 2.3
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_FREQUENCY	M	M	The frequency of snapshots in the AP aging snapshot tables. Possible values are: W for Weekly, M for Monthly, B for both weekly and monthly.	Setting this parameter to weekly results in approximately four times the amount of data as Monthly. Both Weekly and Monthly means that you get a record for each end-of-week date as well as a record for each end-of-month date. Determine frequency based upon the desired view level (for example, do you want to be able to compare things on a monthly, weekly, or daily basis), number of days of data to view (for example, how far back the data goes in the fact table), and amount of physical disk space available	FIN	M
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 if snapshots are replaced. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to add new records and not update previously calculated records. The 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_NUM-DAYS	FIN	P
AP_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name the AP process job uses	Connection name the currently running AP process uses, or if no AP job is running, the last-used source connection	FIN	P

Table 2.3
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_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, 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
AP_PROCESS_RUNNING_JOB_NAME		DAILY_EE_AP_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_HISTORY_PROCESS_DATE	Current date	2012-01-31	Generally internal use only. Internally set to current date, unless DAILY_HISTORY_DATE_SET_CURRENT = N.	Used for testing purposes to process AP records for a different date. Typically should be programatically set to the current date.	FIN	M/P
AP_SE_SNAPSHOT_FREQUENCY	M	M	The frequency of snapshots in the AP aging snapshot tables. Possible values are: 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

Table 2.3
Parameters for the Financials Module (Page 4 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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 if snapshots are replaced. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to add new records and not update previously calculated records. The 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 that loads this run.	Calculated as TODAY minus AP_SE_SNAPSHOT_NUM_DAYS	FIN	P
AR_EE_HISTORY_DATE		20081231	The date the AR aging table refreshes from. Three years back is common. For example, 20081231		FIN	P
AR_EE_HISTORY_DATE_MIN	<customer defined - usually three years back. YYYYMMDD>	20081231	The date the AP aging table refreshes from. Three years back is common. For example, 20081231	Added 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: W for Weekly, M for Monthly, B for both weekly and monthly.		FIN	M

Table 2.3
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_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 if snapshots are replaced. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to add new records and not update previously calculated records. The 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_NUM_DAYS	FIN	P
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 an 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, 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
AR_PROCESS_RUNNING_JOB_NAME		DAILY_EE_AR_CHAINED	Indicates the current or last run job name for AR processing.		FIN	P

Table 2.3
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_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: 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
AR_SE_SNAPSHOT_REPLACE_FLAG	N	N	Indicates if snapshots are replaced. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to add new records and not update previously calculated records. The 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 loads this run.	Calculated as TODAY minus AR_SE_SNAPSHOT_NUM_DAYS	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	DO NOT UPDATE MANUALLY. Maximum Posting Id to extract this iteration.		FIN	P

Table 2.3
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_EE_POSTLINE_MAX	0	0	0		FIN	P
GL_EE_POSTLINE_MAX_Sxx	Does not exist until first EE GL load	441569	DO NOT 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
GLTR_HIST_LOOKBACK_DAYS	500000	1500	Limit HIST job load of load_gltr_hist_history by number of days prior to date of HIST run (impacts perm_gl_transaction_hist)	During installation, if a customer chooses to only have a certain number of months/years of general ledger transaction history, this parameter limits the load and speeds up the historical job process.	FIN	M
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
GL_LOAD_DAYS	152	52	The number of days to load from the look back days forward for GL-related transactional tables		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		FIN	M

Table 2.3
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_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.	Ensure that this parameter is formatted so it can be inserted directly into an SQL “In” clause – including single-quotes around the value(s)	FIN	M
GL_POST_LOOP_NO	0	0			FIN	P
GL_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name the GL process job uses.		FIN	P
GL_PROCESS_DAYS	10	10	The number of days of GL to extract and process from permanent staging tables.		FIN	M

Table 2.3
Parameters for the Financials Module (Page 9 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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, do not update this manually. However, if a job fails and is deleted, 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
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 carries 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	DO NOT UPDATE MANUALLY. The iteration number loading tr_hist.		FIN	P
GLTR_HIST_ROW_COUNT	0	4919	DO NOT UPDATE MANUALLY. Number of rows loaded into gltr_hist.		FIN	P
GLTR_HIST_TRAN_MAX	0	200704110004148000.00	DO NOT UPDATE MANUALLY. Maximum Transaction Id to extract this iteration.		FIN	P

Table 2.3
Parameters for the Financials Module (Page 10 of 10)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
GLTR_HIST_TRAN_MIN	-1	-1	DO NOT UPDATE MANUALLY. Minimum Transaction Id to extract this iteration.		FIN	P
HIST_LOAD_EE_AP_RUNNING	N	C	Indicates the EXTRA_DAY_EE_AP_HISTORY 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_HISTORY 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
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 2.4
Parameters for the Order Management Module (Page 1 of 4)

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
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	Number of days an item can be early and still considered On Time.		OM	M

Table 2.4
Parameters for the Order Management Module (Page 2 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
SALES_LATE_TOLERANCE_DELAY	0	2	Number 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
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

Table 2.4
Parameters for the Order Management Module (Page 3 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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 the SALES process job uses.		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, do not update this manually. However, if a job fails and is deleted, 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."	OM	P
SALES_PROCESS_RUNNING_JOB_NAME		DAILY_SALES_PROCESS_2010SE	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

Table 2.4
Parameters for the Order Management Module (Page 4 of 4)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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
SALES_SNAPSHOT_REPLACE_FLAG	N	N	Indicates if snapshots are replaced. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to add new records and not update previously calculated records. The 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 2.5
Parameters for the Operations Module (Page 1 of 8)

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.	OP	P
INV_BALANCE_HIST_START_DATE	<customer defined. Format YYYY-MM-DD>	2008-01-01	The START date used for Inventory Transactions history loading	Used to determine start date for fact_inv_mth_balance table	OP	M
INV_FUTURE_DEMAND_MONTHS	12	3	The number of months to include in future demand calculation (12 or less).	The days_on_hand_projected field calculates how many days there will be sufficient inventory for a particular item (by source/domain/site). The calculation requires the sum of future orders for the item that were created by MRP (Manufacturing Resource Planning, automated future orders based on need/use). This parameter allows you to limit how many months ahead to look for future orders in order to calculate projected days on hand values.	OP	M
INV_HISTORICAL_PERIOD_COUNT	-24	-24	Number of periods to use to determine Average daily issued quantity historical.		OP	M

Table 2.5
Parameters for the Operations Module (Page 2 of 8)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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		OP	P
INV_LAST_IN_MSTR_EXTRACT		40574.53221	Last Extract Timestamp of data currently in load_in_mstr		OP	P
INV_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name the Inventory process job uses.		OP	P
INV_PROCESS_DAYS	10	10	The number of days of Inventory to extract and process from permanent Data Store tables		OP	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, do not update this manually. However, if a job fails and is deleted, 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."	OP	P
INV_PROCESS_RUNNING_JOB_NAME		DAILY_INV_PROCESS_CHAINED	Indicates the current or last run job name for INVENTORY processing.		OP	P
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.		OP	M
INV_REVALUE_PERIODS	100	10	The number of periods to revalue in inventory balances		OP	M

Table 2.5
Parameters for the Operations Module (Page 3 of 8)

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-SCRP','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		OP	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		OP	Do Not Edit

Table 2.5
Parameters for the Operations Module (Page 4 of 8)

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		OP	Do Not Edit
INV_TRANS_TYPES_COUNT_TR	'CYC-CNT'	'CYC-CNT'	Transaction types (tr_type) that are related to count transactions		OP	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		OP	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).		OP	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		OP	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		OP	Do Not Edit
INV_TRANS_TYPES_RECOUNT	'CYC-RCNT'	'CYC-RCNT'	Transaction types (tr_type) that are related to recount transactions		OP	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		OP	Do Not Edit
INV_TRANS_TYPES_SUPPLIER	'ISS-PRV','RCT-PO'	'ISS-PRV','RCT-PO'	Transaction types (tr_type) that are related to suppliers		OP	Do Not Edit
JOB_INV_PROCESS_BATCH_ID	0	754	The batch id for the current DAILY_INV_PROCESS_JOB job.		OP	P

Table 2.5
Parameters for the Operations Module (Page 5 of 8)

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.		OP	P
JOB_PO_PROCESS_BATCH_ID	0	752	The batch id for the current DAILY_PO_PROCESS_JOB job.		OP	P
OP_MOVING_AVG_WEEKS	13	13	The Number of Weeks to base Moving Average on.		OP	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.		OP	M
OP_PROCESS_CONNECTION_NAME		QMI	Indicates the connection name the OP process job uses.		OP	P
OP_PROCESS_DAYS	5	5	The number of days of OP to extract and process from permanent staging tables.		OP	M
OP_PROCESS_RUNNING	N	N	Indicates if an OP process is currently running (Y, N)		OP	P
OP_PROCESS_RUNNING_JOB_NAME		DAILY_OP_PROCESS_CHAINED	Indicates the current or last run job name for module processing.		OP	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		OP	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		OP	M/P

Table 2.5
Parameters for the Operations Module (Page 6 of 8)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
PO_EARLY_TOLERANCE_DELAY	2	4	Number of days an order is early and still is considered as "on-time".		OP	M
PO_LATE_TOLERANCE_DELAY	2	1	Number of days a PO can be late and still considered as on time.		OP	M
PO_LOOKBACK_DAYS	30	60	Set to the number of days the user is able to backdate the receipt date. This parameter is used by the load table load_prh_hist.		OP	M
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		OP	P
PO_PERCENT_QTY_OVER	10	10	Percentage of Receipt Quantity over Order Quantity for a Purchase Order to still be considered In Full		OP	M
PO_PERCENT_QTY_UNDER	20	20	Percentage of Receipt Quantity under Order Quantity for a Purchase Order to still be considered In Full		OP	M
PO_PROCESS_CONNECTION_NAME		2010SE	Indicates the connection name the PO process job uses.		OP	P
PO_PROCESS_DAYS	10	10	The number of days of PO to extract and process from permanent staging tables.		OP	M

Table 2.5
Parameters for the Operations Module (Page 7 of 8)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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, do not update this manually. However, if a job fails and is deleted, 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."	OP	P
PO_PROCESS_RUNNING_JOB_NAME		DAILY_PO _PROCESS _2010SE	Indicates the current or last run job name for PO processing.		OP	P
PO_RET_QTY_TOLERANCE	15	15	Percentage of Returns Quantity used for calculating Defect Rate.		OP	M
PO_SNAPSHOT_FREQUENCY	M	W	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.	Default modified to M in BI 3.10 (default was B, but that results in a much larger snapshot table that takes longer to process).	OP	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		OP	P
PO_SNAPSHOT_NUM_DAYS	31	31	The number of days of snapshots to load back each run.		OP	M

Table 2.5
Parameters for the Operations Module (Page 8 of 8)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
PO_SNAPSHOT_REPLACE_FLAG	N	N	Indicates if snapshots are replaced. Y for replace, N for not replaced.	Y to replace the snapshot for recalculated periods when the snapshot job is run; N to add new records and not update previously calculated records. The default value is N.	OP	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.		OP	P

Enterprise Asset Management Module

Table 2.6
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	If no valid domain exists, default Domain to use when loading the EAM Data.		EAM	M

Table 2.6
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_DEFAULT_ENTITY	000000	000000	If no valid entity exists, default entity to use when loading the EAM Data.		EAM	M
EAM_DEFAULT_SOURCE		Unknown Source	If no valid source system exists, the default value used for unknown EAM source system.		EAM	M
EAM_PROCESS_CONNECTION_NAME		EAM QPS	EAM_QPS Internal 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 EAM_PROCESS_DAYS _DAILY or EAM_PROCESS_DAYS _HIST, depending on type of run.		EAM	P
EAM_PROCESS_RUNNING_JOB_NAME		DAILY_EAM _PROCESS _CHAINED	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	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

Table 2.6
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_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
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

Table 2.6
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_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_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
JOB_EAM_PROCESS_BATCH_ID	0	730	The batch id for the current DAILY_EAM_PROCESS job.		EAM	P
load_eam_employee_erp_table_name	<none>	employee_eam	Internal use only. Specifies the table name in EAM based on version; employee or employee_eam	Programatically assigned to the Progress ERP table name for EAM employees. Older versions use ‘employee' while later EAM versions compatible with EE use 'employee_eam' for uniqueness	EAM	P

Transportation Management Module

Table 2.7
Parameters for the Transportation Management Module (Page 1 of 2)

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, and so on).		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
TM_CORP_CURRENCY_CODE	USD	EUR	Corporate currency code		TMS	M
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 the TM process job uses.		TMS	P
TM_PROCESS_DAYS	100		The number of days of TM to extract and process from permanent staging tables.		TMS	M

Table 2.7
Parameters for the Transportation Management Module (Page 2 of 2)

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
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_UNKNOWN_VALUE	Unknown	Unknown	Default value to insert for a column containing NULLs. Used for Primary Key Columns.		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
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: 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

Service and Support Management Module

Table 2.8
Parameters for the Service and Support Management Module

Parameter Name	Parameter Default Value	Parameter Sample Value	Parameter Comments	Explanation/Notes	Module	Manual/ Programmatic
JOB_SSM_PROCESS_BATCH_ID	0	117	The batch id for the current DAILY_SSM_PROCESS_JOB job.		SSM	P
SSM_HISTORY_DATE	Set by INITIAL_JOB_SETUP_DATE parameter.	2010-01-01	The processing date used for SSM orders history calculations. Format is YYYY-MM-DD.		SSM	M
SSM_LOOKBACK_DAYS	30	30	Set to the number of days the user is able to backdate the effective_date. This parameter is used by the load table load_cah_hist.		SSM	M
SSM_PROCESS_CONNECTION_NAME		QMI	Indicates the connection name the SSM process job uses.		SSM	P
SSM_PROCESS_DAYS	10	5	The number of days of SSM to extract and process from permanent staging tables.		SSM	M
SSM_PROCESS_RUNNING	N	N	Indicates if an SSM process is currently running (Y, N)		SSM	P
SSM_PROCESS_RUNNING_JOB_NAME		DAILY_SSM_PROCESS_CHAINED	Indicates the current or last run job name for module processing.		SSM	P

System Information

Table 2.9
System Information

Related to:	Information Needed
Progress Database	Have they already set up ODBC connections to Progress? If not, 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

Parameter Values that Need to Be Set Before History Load

Table 2.10
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 refreshes 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 refreshes 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
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_CALENDAR_DOMAIN	<Enter domain here for corporate calendar setup>	The Domain Code used for the corporate financial calendar		COM
CORP_CALENDAR_SOURCE	<Enter data source name here for corporate calendar setup>	The Source System Code used for the corporate financial calendar		COM
CORP_CURRENCY_CODE	USD	Corporate currency code used for high-level reporting	The currency code for corporate amounts calculations. Same for entire warehouse.	COM

Table 2.10
Parameter Values that Need to Be Set Before History Load (Page 2 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
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
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	Must 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.	Must always be smaller than GL_LOAD_ DAYS.	FIN

Table 2.10
Parameter Values that Need to Be Set Before History Load (Page 3 of 5)

Parameter Name	Parameter Default Value	Parameter Comments	Notes	Module
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
GLTR_HIST_LOOKBACK_DAYS	50000	Limit HIST job load of load_gltr_hist_history by number of days prior to date of HIST run (impacts perm_gl_transaction_hist)	During installation, if a customer chooses to only have a certain number of months/years of general ledger transaction history, this parameter limits the load and speeds up the historical job process.	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	2010 SE		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
INITIAL_JOB_SETUP_CONNECTION_01_RUN	N	This determines if the Initial Job Setup was run against this data source. Any new data source must 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 rerun 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

Table 2.10
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_VERSION	2011	Version - specify version of the QAD Application, to skip tables in BI that do not exist in older versions; for example, 2012		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. The 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. The 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 was run against this data source. Any new data source must have an N.	Added in 3.4.1 – Used by INITIAL_JOB_SETUP job to ensure that existing data sources are NOT rerun as part of the historic load when new sources are added.	COM
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. The 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. The default is fine for most installations as any uninstalled modules are skipped.	COM

Table 2.10
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 _ VERSION	2011	Version - specify version of the QAD Application, to skip tables in BI that do not exist in older versions; for example, 2012		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
INV_BALANCE_HIST_START_DATE	<customer defined. Format YYYY-MM-DD>	The START date used for Inventory Transactions history loading. How far back in time should the fact table look back for Inventory?	Used to determine start date for fact_inv_mth_balance table	OP
SALES_SNAPSHOT_HIST_START_DATE	<Set to today's date before starting the initial history load>	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

Chapter 3

QAD BI Upgrade Instructions

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

Upgrade Information 86

Upgrade to QAD BI 3.11 from 3.10 86

Upgrade to QAD BI 3.10 from QAD BI 3.9 102

Upgrade to QAD BI 3.9 from QAD BI 3.8.1/3.8.2 108

Upgrade to QAD BI 3.8.1 from QAD BI 3.7 115

Upgrade to QAD BI 3.7 from QAD BI 3.6.x 121

Upgrade to QAD BI 3.6 from QAD BI 3.5 129

Upgrade to QAD BI 3.5 from QAD BI 3.4.1 138

Upgrade Information

To upgrade to the latest version of QAD BI, 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 3.11 from 3.10

- Upgrade Steps for QAD BI Data Warehouse Designer
- Upgrade Steps for QAD BI Metadata
  - Preparation
  - Upgrade Steps
- BI Portal Upgrade Procedure
  - Preparation
  - Upgrade Steps for QAD BI Portal
  - Verification
  - Update Portal Model
  - Update Portal Documentation and Standard Objects

Upgrade Steps for QAD BI Data Warehouse Designer

1. Stop all Schedulers.
2. Install the new QAD BI Data Warehouse Designer, version 6.7.4, by running the QAD_67400.msi installer application.

   Note You can install the new version in the same location as the old version, overwriting the old version; or, you can install the new version into a different location, preserving the old version. If you install the new version into a different location, you must remove and re-create the scheduler for each Data Warehouse Database that you update to use the new version.
3. Run the new version 6.7.4 Setup Administrator. Choose Metadata Repository|Validate Metadata Repository.
   a. Select the ODBC DSN of the DWD that you want to upgrade and click OK.
   b. Verify that the screen says “Version is 6.6. Current version is 6.7.” When the system prompts you to validate all Meta Tables, click OK.
   c. A dialog box asks “Do you wish to upgrade?” Click OK.
   d. A list of update commands is displayed. Click OK.
   e. At the “Define FileGroups” dialog, click OK.
4. Launch the new version 6.7.4 BI Data Warehouse Designer and verify that you can connect to your upgraded Data Warehouse.

5 If you installed the new version into a different directory than the old version, you must update your Schedulers to ensure they are running the new version. To do this, run the QAD System Administrator application. Go to Scheduler Maintenance. For each active Scheduler, right-click and choose Configure. Write down all of the Scheduler configuration details. Then, delete the Scheduler and re-create it.

Upgrade Steps for QAD BI Metadata

Preparation

1 Plan for a couple of hours of down time.

2 Ensure that there are no loads in progress. All jobs must be completed.

3 Put all scheduled jobs On Hold until the upgrade is complete. Shut down the scheduler.

Important Make a backup of the current warehouse database and the current portal database.

Upgrade Steps

1 Locate the metadata upgrade applications for the Common module and any other modules you have installed.

2 You can find these in the install-files\metadata subdirectory of the Module physical media or QAD Store product download. Because the metadata files are segregated by module, it is only necessary to install the modules that you need.

3 Perform the Common upgrade first. You can do the other upgrades in any order.

4 If you process any of the Order Management (Sales) fact_om_* tables, install the Order Management (OM) upgrade application.

Note This upgrade overwrites your Sales OLAP cube. If you have customized your cube in any way, make a copy of it before installing the upgrade.

5 If you process Inventory, Purchase Order, or Capacity Planning/Operations Efficiency tables, install the Operations module upgrade application.

6 If you process any Financial tables, install the Financials module upgrade application.

7 If you have an EAM source, install the EAM module upgrade application.

8 To install an upgrade module, log in to Setup Administrator.

9 Choose Tools|Start Logging. Select a name and location for your log file and save.

10 Load the Common Module Upgrade Application as follows:

a Set your application directory to the new `metadata/common` folder.

b Load the `upgradeCom` application 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.

e 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.

f 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 the box next to *Existing Dimension objects will be* and select Altered.
- On the left, select Dimension View. On the right, click the box next to *Existing Dimension View objects will be* and select Recreated.
- On the left, select Stage Table. On the right, click the box next to *Existing Stage table objects will be* and select Recreated.
- On the left, select Permanent Stage Table. On the right, click the box next to *Existing Permanent Stage table objects will be* and select Altered.
- On the left, select Fact Table. On the right, click the box next to *Existing Fact table objects will be* and select Altered.

g Click OK.

11 Repeat for any other modules that you must upgrade.

a **Order Management.** Change the Application Directory to the `metadata /order_management` subdirectory and install the `upgradeOM` application.

Note You may see an error that the index fact_om_shipment_idx_A could not be created. Make note of this, because you must create the index manually after the upgrade jobs are run.

b **Operations.** Change the Application Directory to the `metadata/operations` subdirectory and install the `upgradeOps` application.

c **Financials.** Change the Application Directory to the `metadata/financials` subdirectory and install the `upgradeFin` application.

d **EAM.** Change the Application Directory to the `metadata/eam` subdirectory and install the `upgradeEAM` application.

12 Start the Scheduler. Make sure that all jobs are still suspended.

13 If the upgrade affects any customized tables, reapply your customizations. For a list of new and modified tables, see “New and Modified Tables” on page 92.

14 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:

Job Changes/New Jobs for Module common

- DAILY_START
- DAILY_MODULES_LOAD_FINISH
- ITEM_PROCESS
- DAILY_LOAD_TR_HIST
- HIST_LOAD_TR_HIST

- HIST_COMMON_PERM_CHAINED
- HIST_COMMON_PROCESS_CHAINED
- CONFIRM_CONNECTION_XXXXXXXX
- CLEANUP_COM_TABLES
- PREPROCESS_COMMON_TRUNCATE
- PREPROCESS_COMMON_LOADS
- DAILY_COMMON_PERM_CHAINED
- DAILY_COMMON_PROCESS_CHAINED
- PREPROCESS_COMMON_DAILYONLY_LOADS
- UPGRADE_310_TO_311_COMMON_STEP1
- UPGRADE_310_TO_311_COMMON
- UPGRADE_310_TO_311_COMMON_EE_SE_LOADS
- UPGRADE_310_TO_311_COMMON_EE_SE_LOADS_EE2010UP
- UPGRADE_310_TO_311_COMMON_EE_SE_LOADS_EE
- UPGRADE_310_TO_311_COMMON_EE_SE_LOADS_SE

Job Changes/New Jobs for Module financials

- HIST_FIN_EE_PERM_CHAINED
- HIST_FIN_EE_PERM_EXTRACT
- DAILY_FIN_EE_PERM_CHAINED
- CLEANUP_FIN_TABLES
- UPGRADE_310_TO_311_FIN_SE_STEP3
- UPGRADE_310_TO_311_FIN_SE_STEP2_XXXXXXXX
- UPGRADE_310_TO_311_FIN_SE_STEP2
- UPGRADE_310_TO_311_FIN_SE_STEP1
- UPGRADE_310_TO_311_FIN_SE
- UPGRADE_310_TO_311_FIN_EE

Job Changes/New Jobs for Module eam

- CLEANUP_EAM_TABLES
- PREPROCESS_EAM_TRUNCATE
- PREPROCESS_EAM_LOADS
- HIST_EAM_PERM_CHAINED
- DAILY_EAM_PERM_CHAINED

Job Changes/New Jobs for Module operations

- CLEANUP_OP_TABLES
- DAILY_OP_PERM_CHAINED
- HIST_OP_PERM_CHAINED
- DAILY_OP_PO_PROCESS_CHAINED
- DAILY_OP_INV_PROCESS_CHAINED

- HIST_OP_INV_PROCESS_CHAINED
- UPGRADE_310_TO_311_OP

Job Changes/New Jobs for Module order_management

- POSTPROCESS_OM_CUBES
- HIST_OM_PROCESS_CHAINED
- CLEANUP_OM_TABLES
- VALIDATION_OM
- DAILY_OM_PROCESS_CHAINED
- DAILY_OM_PERM_EXTRACT
- DAILY_OM_PERM_CHAINED
- HIST_OM_PERM_EXTRACT
- HIST_OM_PERM_CHAINED
- UPGRADE_310_TO_311_OM_QBI_4364
- UPGRADE_310_TO_311_OM_STEP3
- UPGRADE_310_TO_311_OM_STEP2_XXXXXXXX
- UPGRADE_310_TO_311_OM_STEP2
- UPGRADE_310_TO_311_OM_STEP1
- UPGRADE_310_TO_311_OM

Job Changes/New Jobs for Module tm

- CLEANUP_TM_TABLES

Job Changes/New Jobs for Module ssm

- CLEANUP_SSM_TABLES

15 Make sure the INITIAL_JOB_SETUP_ENABLED parameter is set to Y and run INITIAL_JOB_SETUP to rebuild the daily and historical job chains.

16 Run the upgrade jobs for BI 3.11 (UPGRADE_310_TO_311%), making sure you run the Common Module upgrade job first, then the others in any order:

a **Common Module.** Run the Upgrade_310_to_311_COM job.

b **Order Management Module.** Run the UPGRADE_310_to_311_OM job. Note that if you received an error during the application load that the index fact_om_shipment_idx_A could not be created, create it after the upgrade job completes. To do this, go to fact_om_shipment in the Metadata Browser, right-click and select Display Indexes, then right-click on fact_om_shipment_idx_A and select Create Index.

c **Operations Module.** Run the UPGRADE_310_to_311_OP job.

d **Financials SE Module.** Run the UPGRADE_310_to_311_FIN_SE job.

e **Financials EE Module.** Run the UPGRADE_310_to_311_ FIN_EE job.

f **EAM Module.** There are no EAM upgrade jobs.

g **SSM Module.** There are no SSM upgrade jobs.

h **TM Module.** There are no Transportation Management upgrade jobs.

17 Run the Daily load.

18 If you used the Sales OLAP cube, this was re-created as separate cubes for each fact table. If you want to use the new cubes, you must configure them in the DWD and then create them in SQL Server Analysis Services.

a In Analysis Services, delete each cube and all dimensions.

b In the DWD, if you want to change the Analysis Services database name from the default value, set the name in each Cube's Properties dialog.

c For the cubes that you want to build, right-click on the cube and select Create Cube.

d Run the jobs to process each cube.

Note that the old (non-OLAP) Sales cubes are no longer supported.

BI Portal Upgrade Procedure

Preparation

1 Shut down Tomcat.

2 Back up the BI Portal database.

Upgrade Steps for QAD BI Portal

1 Back up your BI 3.10 Web application by copying the `TOMCAT_HOME\webapps\your_bi_appname` directory to `TOMCAT_HOME\your_bi_appname.save10`.

2 Unzip the new BI 3.11 Portal `qadbi-portal.war` file into `TOMCAT_HOME\webapps\your_bi_appname`, overwriting any existing files with the newer version. Your configuration files are not overwritten.

3 In SQL Server Management Studio, run the upgrade script against your BI Portal database: `TOMCAT_HOME\webapps\your_bi_appname\install\sqlserver\v310tov311.sql`

4 Start Tomcat.

Verification

Connect to your BI Portal. Click the About link on the login screen and verify that the Version is 3.11.

Note If necessary, clear the user's browser cache before 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.

Update Portal Model

1. Log in to the Portal as an Administrator.
2. Go to the Administration menu and choose Model Administration. Click Rebuild Master.
3. After the model is 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

1. Update the metadata documentation:
   a. Delete the existing `TOMCAT_HOME\webapps\your_bi_appname\documentation\suite` directory.
   b. Unzip the latest `qadbi-documentation.zip`, on the BI 3.11 Common Module physical media or QAD Store product download (`install-files` directory), into your `TOMCAT_HOME\webapps\your_bi_appname\documentation` directory.
2. To install the latest BI Portal standard objects, copy the files from the BI 3.11 Module physical media or QAD Store product download (`install-files\portal` subdirectory) 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.
3. In the BI Portal, log in as an Administrator.
4. Go to the Administration|Data Migration screen.
5. Click the Import tab.
6. Select the Update existing records check box.
7. Select the File on Server radio button.
8. Enter `qad_common_dashboards.zip` file name in the field and click Import.
9. Repeat for any other module reports and dashboards `.zip` files that are required.

New and Modified Tables

Metadata changes for Version 3.11

Changes for Module common

- perm_item_master
- dim_address
- dim_business_relation
- dim_channel
- dim_credit_term

- dim_customer
- dim_item
- dim_supplier
- stage_address
- stage_address_master
- stage_business_relation
- stage_channel
- stage_code_master
- stage_customer
- stage_ee_account
- stage_ee_address
- stage_ee_contact
- stage_ee_customer
- stage_ee_daybook
- stage_ee_supplier
- stage_item
- stage_item_list
- stage_item_master
- stage_item_site
- stage_se_credit_term
- stage_se_customer
- stage_se_entity
- stage_se_supplier
- stage_site
- stage_site_missing
- stage_site_working_calendar
- stage_source_system
- stage_supplier
- stage_transaction_hist
- stage_wo_mstr
- stage_wod_det
- stage_wr_route
- load_ad_mstr
- load_ar_mstr
- load_code_mstr
- load_cs_mstr
- load_ct_mstr
- load_ee_Address

- load_ee_BusinessRelation
- load_ee_Contact
- load_ee_Country
- load_in_mstr
- load_ld_det
- load_pt_mstr
- load_rnd_mstr
- load_source_list
- load_tr_hist
- load_tr_hist_history
- load_tr_hist_list
- load_tr_hist_remarks
- load_wo_mstr
- load_wo_mstr_hist
- dim_customer_bill_to
- dim_customer_ship_to
- dim_customer_sold_to
- dim_ship_to_address
- perm_code_master
- perm_site_working_calendar
- perm_transaction_hist
- perm_work_order_detail
- perm_work_order_master
- perm_work_order_routing
- permsup_transaction_hist

Changes for Module financials

- fact_ap_check
- fact_ap_voucher
- fact_ee_ar_customer_invoice_history
- extract_ap_master
- extract_gl_transaction_hist
- stage_ap_check_base
- stage_ap_check_base1
- stage_ap_check_base2
- stage_ap_check_base3
- stage_ap_check_base4
- stage_ap_master

- stage_ap_voucher_base
- stage_ap_voucher_base1
- stage_ap_voucher_base2
- stage_ap_voucher_base3
- stage_ap_voucher_base4
- stage_ap_voucher_history
- stage_ap_voucher_history1
- stage_ar_invc_hist_age_recalc
- stage_ar_invoice_history
- stage_ee_accounting_period
- stage_ee_ap_supplier_invoice
- stage_ee_ap_supplier_invoice_hist1
- stage_ee_ap_supplier_invoice_history
- stage_ee_ap_supplier_invoice_history3
- stage_ee_ap_supplier_invoice_history4
- stage_ee_ap_supplier_invoice_snapshot1
- stage_ee_ap_supplier_invoice2
- stage_ee_ar_customer_invoice
- stage_ee_ar_customer_invoice_history
- stage_ee_ar_customer_invoice_history2
- stage_ee_ar_customer_invoice_history3
- stage_ee_ar_customer_invoice_snapshot1
- stage_ee_ar_customer_invoice2
- stage_ee_ar_customer_invoice3
- stage_ee_customer_invoice_movement_perm
- stage_ee_customer_invoice_perm
- stage_ee_customer_invoice_stage_perm
- stage_ee_customer_ship_to_perm
- stage_ee_posting_hist_perm
- stage_ee_posting_line_perm
- stage_ee_posting_perm
- stage_ee_supplier_invoice_movement_perm
- stage_ee_supplier_invoice_perm
- stage_ee_supplier_invoice_stage_perm
- stage_gl_balance
- stage_gl_balance_perm1
- stage_gl_balance_perm2
- stage_gl_balance_ptd

- stage_gl_transaction_budget
- stage_se_ap_snap2
- stage_se_ar_invoice_history
- work_ap_change_list1
- work_ap_change_list2
- work_ap_change_list3
- work_ap_change_list4
- work_ap_change_list5
- work_ar_change_list1
- work_ar_change_list2
- work_ar_change_list3
- work_ee_customer_invoice_hist_list
- work_ee_customer_invoice_stage_change_list
- work_ee_customer_invoice1
- work_ee_customer_invoice2
- work_ee_customer_invoice3
- work_ee_gl_change_hist_list
- work_ee_gl_change_list1
- work_ee_gl_change_list2
- work_ee_gl_change_list3
- work_ee_gl_change_list4
- work_ee_posting_history_list
- work_ee_supplier_invoice_hist_list
- work_ee_supplier_invoice_stage_change_list
- work_ee_supplier_invoice1
- work_ee_supplier_invoice2
- work_ee_supplier_invoice3
- work_gl_change_list1
- work_gl_change_list2
- work_gl_trans_change_list
- load_ap_mstr
- load_ee_PostingHist
- load_gltr_hist_history
- perm_ap_master
- perm_ee_posting

Changes for Module eam

- fact_eam_inv_transaction

- stage_eam_inv_trans_perm
- stage_eam_inv_transaction
- load_eam_employee
- load_eam_parts_status
- perm_eam_inv_transaction

Changes for Module operations

- fact_inv_transaction
- fact_po_order_history
- fact_po_receipt
- fact_wo_order
- fact_wo_routing_transaction
- dim_production_line
- extract_inv_trans_hist_cons
- extract_inv_transaction_hist
- extract_op_transaction_hist
- extract_po_order_header
- extract_po_order_line
- extract_po_receipt_hist
- extract_po_transaction_hist
- stage_in_mstr_max_dates
- stage_inv_bal_current_max_mth
- stage_inv_bal_item_list
- stage_inv_bal_mth_cons
- stage_inv_bal_mth_cons1
- stage_inv_bal_mth_cons3
- stage_inv_bal_mth_cons4
- stage_inv_bal_mth_cons5
- stage_inv_bal_mth_current
- stage_inv_bal_mth_current1
- stage_inv_bal_mth_current2
- stage_inv_bal_mth_current3
- stage_inv_bal_mth_current4
- stage_inv_bal_mth_date1
- stage_inv_bal_mth_list
- stage_inv_bal_mth_trans
- stage_inv_bal_mth_trans1
- stage_inv_bal_revalue2

- stage_inv_mth_balance
- stage_inv_mth_balance_hist_calc
- stage_inv_mth_balance_hist_fiscal_periods
- stage_inv_mth_balance_hist_periods
- stage_inv_mth_balance1
- stage_inv_mth_end_cst
- stage_inv_mth_end_cst1
- stage_inv_transaction
- stage_inv_transaction_con_cus3
- stage_inv_transaction2
- stage_inv_transaction3
- stage_inv_transaction4
- stage_inv_trx_chg_mths
- stage_inv_trx_chg_mths1
- stage_mrp_calculations
- stage_mrp_calculations1
- stage_op_tr_hist_rjct_wo
- stage_po_buyer_code_master
- stage_po_buyer_po_header
- stage_po_order_hist
- stage_po_order_hist_daily
- stage_po_order_hist_daily1
- stage_po_order_hist_daily3
- stage_po_order_hist_daily6
- stage_po_order_hist_daily7
- stage_po_order_hist_initial
- stage_po_order_hist_initial1
- stage_po_order_hist_initial3
- 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_line
- stage_wo_order0
- stage_wo_routing_transaction0

- work_inv_change_hist_list
- work_inv_change_list1
- work_po_change_hist_list
- work_po_change_list1
- work_po_change_list2
- work_po_change_list3
- work_po_change_list4
- load_po_mstr
- load_pod_det
- dim_release_date
- perm_po_order_header
- perm_po_order_line
- perm_po_receipt_hist
- perm_work_center_master
- permsup_po_order_header
- permsup_po_order_line
- permsup_po_receipt_hist

Changes for Module order_management

- fact_om_order_history
- fact_om_order_performance
- fact_om_shipment
- extract_om_invoice_header
- extract_om_invoice_line
- extract_om_schedule_line
- extract_om_shipment_requirement_line
- extract_om_transaction_hist
- stage_om_booking_max_tran
- stage_om_booking_orig_order
- stage_om_booking_transaction1
- stage_om_booking1
- stage_om_booking2
- stage_om_booking3
- stage_om_invoice
- stage_om_invoice_order_line
- stage_om_invoice_sched_discrete_qty
- stage_om_invoice_scheduled_transaction
- stage_om_invoice_transaction1

- stage_om_invoice1
- stage_om_invoice2
- stage_om_invoice3
- stage_om_ord_perf
- stage_om_ord_perf_all
- stage_om_order
- 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_order_snap2
- stage_om_order_transaction1
- stage_om_order_transaction5
- stage_om_order1
- stage_om_order2
- stage_om_order3
- stage_om_schedule_detail1
- stage_om_schedule_line
- stage_om_scheduled_order_perf
- stage_om_scheduled_order_perf1
- stage_om_shipment
- stage_om_shipment_requirement
- stage_om_shipment_requirement_line
- stage_om_shipment_requirement1
- stage_om_shipment_requirement2
- stage_om_shipment1
- stage_om_shipment2
- stage_om_shipment3
- val_OM_SE_Invoice_History
- work_om_change_hist_list
- work_om_change_list
- work_om_change_list1
- work_om_change_list2
- work_om_change_list3
- work_om_change_list4
- work_om_change_list5
- load_abs_mstr

- load_absr_det
- load_cncu_mstr
- load_idh_hist
- load_ih_hist
- load_sch_mstr
- load_schd_det
- load_sod_det
- load_val_OM_SE_BI_Invoice_History
- load_val_OM_SE_ERP_Invoice_History
- perm_customer_consignment
- perm_om_invoice_header
- perm_om_invoice_line
- perm_om_schedule_line
- perm_om_shipment_requirement_line
- permsup_om_invoice_header
- permsup_om_invoice_line
- SalesBookings
- SalesInvoices
- SalesOrders
- SalesPerformance
- SalesShipments

Changes for Module tm

None

Changes for Module ssm

- fact_ssm_call_history
- extract_ssm_call_history
- stage_ssm_call_event_labor_snap1
- stage_ssm_call_event_non_labor_snap1
- stage_ssm_call_master
- work_ssm_change_list1
- work_ssm_change_list2
- work_ssm_change_list3
- work_ssm_change_list4
- work_ssm_change_list5

Upgrade to QAD BI 3.10 from QAD BI 3.9

Upgrade Steps for QAD BI Metadata

Preparation

1. Plan for a couple of hours of down time.
2. Ensure that there are no loads in progress. All jobs must be completed.
3. Put all scheduled jobs On Hold until the upgrade is complete. Shut down the scheduler.

Important Make a backup of the current warehouse database and the current portal database.

Upgrade Steps

1. Locate the metadata upgrade applications for the Common module and any other modules you have installed. You can find these in the `install-files\metadata` subdirectory of the Module physical media or QAD Store product download. Because the metadata files are segregated by module, it is only necessary to install the modules that you need.
2. Perform the Common upgrade first. You can do the other upgrades in any order.
   a. If you process any of the Order Management (Sales) fact_om_* tables, install the Order Management (OM) upgrade application.

      Note: This upgrade overwrites your Sales OLAP cube. If you have customized your cube in any way, make a copy of it before installing the upgrade.

   b. If you process Inventory, Purchase Order, or Capacity Planning/Operations Efficiency tables, install the Operations module upgrade application.

      Note: This upgrade overwrites your Purchasing and Inventory OLAP cubes. If you have customized your cube in any way, make a copy of it before installing the upgrade.

   c. If you process any Financial tables, install the Financials module upgrade application.
   d. If you have an EAM source, install the EAM module upgrade application.
3. To install an upgrade module, log in to Setup Administrator.
4. Choose Tools|Start Logging. Select a name and location for your log file and save.
5. Load the Common Module Upgrade Application as follows:
   a. Set your application directory to the new `metadata/common` folder.
   b. Load the `upgradeCom` application 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.

e 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.

f 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 the box next to *Existing Dimension objects will be* and select Altered.
- On the left, select Dimension View. On the right, click the box next to *Existing Dimension View objects will be* and select Recreated.
- On the left, select Stage Table. On the right, click the box next to *Existing Stage table objects will be* and select Recreated.
- On the left, select Permanent Stage Table. On the right, click the box next to *Existing Permanent Stage table objects will be* and select Altered.
- On the left, select Fact Table. On the right, click the box next to *Existing Fact table objects will be* and select Altered.

g Click OK.

6 Repeat for any other modules that you must upgrade.

a **Order Management.** Change the Application Directory to the `metadata /order_management` subdirectory and install the `upgradeOM` application.

b **Operations.** Change the Application Directory to the `metadata/operations` subdirectory and install the `upgradeOps` application.

c **Financials.** Change the Application Directory to the `metadata/financials` subdirectory and install the `upgradeFin` application.

Note: If you currently have EE Financials and have your snapshot frequency set to 'B' for the AP or AR snapshots (or both), you may get an error message that an index cannot be created on the fact_ee_ar_customer_invoice_snapshot and/or fact_ee_ap_supplier_invoice_snapshot tables. If this occurs, note the index name, because you must create it after the upgrade job runs.

d **EAM.** Change the Application Directory to the `metadata/eam` subdirectory and install the `upgradeEAM` application.

7 Start the Scheduler. Make sure that all jobs are still suspended.

8 If the upgrade affects any customized tables, reapply your customizations. For a list of new and modified tables, see “Data Warehouse Tables Changed” on page 112.

9 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:

- OP_HIST_LOAD
- DAILY_COMMON_PERM_CHAINED
- HIST_COMMON_PERM_CHAINED
- DAILY_COMMON_PROCESS_CHAINED
- HIST_EAM_PROCESS_CHAINED
- DAILY_OP_PERM_CHAINED
- HIST_OP_PERM_CHAINED
- DAILY_OP_INV_PROCESS_CHAINED

- HIST_COMMON_PROCESS_CHAINED
- CLEANUP_COMMON_TABLES
- PREPROCESS_COMMON_LOADS
- PREPROCESS_COMMON_TRUNCATE
- PREPROCESS_COMMON_DAILYONLY_LOADS
- PREPROCESS_COMMON_HISTONLY_LOADS
- DAILY_EAM_PROCESS_CHAINED
- HIST_OP_INV_PROCESS_CHAINED
- PREPROCESS_OP_LOADS
- PREPROCESS_OP_TRUNCATE
- PREPROCESS_OP_DAILYONLY_LOADS
- PREPROCESS_OP_HISTONLY_LOADS
- DAILY_OM_PROCESS_CHAINED
- HIST_OM_PROCESS_CHAINED

10 Make sure the INITIAL_JOB_SETUP_ENABLED parameter is set to Y and run INITIAL_JOB_SETUP to rebuild the daily and historical job chains.

11 Run the upgrade jobs for BI 3.10, making sure you run the Common Module upgrade job first, then the others in any order:

a **Common Module.** Run the Upgrade_39_to_310_COM job.

b **Operations module.** Run the UPGRADE_39_to_310_OP job.

c **Financials SE Module.** Run the UPGRADE_39_to_310_FIN_SE_SNAPSHOTS job.

d **Financials EE Module.** Run the UPGRADE_39_to_310_ FIN_EE_AR_SNAPSHOT and UPGRADE_39_to_310_FIN_EE_AP_SNAPSHOT jobs. If you had errors on index creation during the installation of the upgrade application, you should be able to create those indexes now:

In the DWD, go to the fact_ee_ap_supplier_invoice_snapshot table._Right-click on the table and choose Display Indexes. Locate index fact_ee_ap_si_snapshot_idx_A; right-click on this index and choose Create Index. Ensure that the index is created successfully.

Go to the fact_ee_ar_customer_invoice_snapshot table. Right-click on the table and choose Display Indexes. Locate index fact_ee_ar_ci_snapshot_idx_A; right-click on this index and choose Create Index. Ensure that the index is created successfully.

e **EAM Module.** Run the UPGRADE_39_TO_310_EAM job.

12 Run the Daily load.

13 If you used the OLAP cubes, re-create them in SQL Server Analysis Services.

a In Analysis Services, delete each cube and all dimensions.

b In the DWD, if you want to change the Analysis Services database name from the default value, set the name in each Cube’s Properties dialog.

c For the cubes that you want to build, right-click on the cube and select Create Cube.

d Run the jobs to process each cube.

14 In prior releases, the PO_SNAPSHOT_FREQUENCY parameter was set to "B" (Both Monthly and Weekly) by default. This results in a much larger snapshot table than Monthly-only snapshots. If you want to change from Both to Monthly-only snapshots, set your PO_SNAPSHOT_FREQUENCY to "M." If you want to remove any weekly snapshot rows from your Purchase Order Snapshot table and only retain the monthly snapshot rows, run the UPGRADE_39_TO_310_PO_SNAPSHOT_FOR_B_TO_M job.

BI Portal Upgrade Procedure

Preparation

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

Upgrade Steps for QAD BI Portal

1. Back up your BI 3.9Web application by copying the `TOMCAT_HOME\webapps\your_bi_appname` directory to `TOMCAT_HOME\your_bi_appname.save9`.
2. Delete all files in the `TOMCAT_HOME\webapps\your_bi_appname\WEB-INF\lib` directory.
3. Unzip the new BI 3.10 Portal `qadbi-portal.war` file into `TOMCAT_HOME\webapps\your_bi_appname`, overwriting any existing files with the newer version. Your configuration files are not overwritten.
4. Edit your `TOMCAT_HOME\webapps\your_bi_appname\WEB-INF\config\server-config.xml` and add the new *<saml>* and *<userMaint>* sections, which you can copy from the new `server-config.xml.default`.
5. Start Tomcat.

Verification

1. Connect to your BI Portal. Click the About link on the login screen and verify that the Version is 3.10.

   Note If necessary, clear the user's browser cache before 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.

Update Portal Model

1. Log in to the Portal as an Administrator.
2. Go to the Administration menu and choose Model Administration. Click Rebuild Master.
3. After the model is 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

1. Update the metadata documentation:
   a. Delete the existing `TOMCAT_HOME\webapps\your_bi_appname\documentation\suite` directory.
   b. Unzip the latest `qadbi-documentation.zip`, on the BI 3.10 Common Module physical media or QAD Store product download (`install-files` directory), into your `TOMCAT_HOME\webapps\your_bi_appname\documentation` directory.
2. To install the latest BI Portal standard objects, copy the files from the BI 3.10 Module physical media or QAD Store product download (`install-files\portal` subdirectory) 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.
3. In the BI Portal, log in as an Administrator.
4. Go to the Administration|Data Migration screen.
5. Click the Import tab.
6. Select the Update existing records check box.
7. Select the File on Server radio button.
8. Enter the `qad_common_dashboards.zip` file name in the field and click Import.
9. Repeat for any other module reports and dashboards `.zip` files that are required.

Data Warehouse Tables Changed

The following section lists the Data Warehouse tables that were changed, added, or deleted.

Note The tables on the Modified list may have been changed structurally. For example:

- Columns added or deleted
- Indexes added or deleted
- Changes made to business display names
- Changes to visibility of particular columns in the BI Portal
- Any combination of these modifications
- The procedure that populates the table has changed

List of New and Modified Table

Common Module - New and Modified Tables

Note Some of these tables have moved from the Operations module to the Common module since they are shared by the new SSM module; for example, the tables related to the processing of op_hist and wo_mstr.

- dim_item
- dim_site
- load_op_hist
- load_pt_mstr
- load_tr_hist
- load_tr_hist_history
- load_tr_hist_list
- load_tr_hist_remarks
- load_wo_mstr
- load_wo_mstr_hist
- load_wod_det
- load_wod_det_hist
- load_wr_route
- load_wr_route_hist
- perm_op_hist
- perm_site_working_calendar
- perm_transaction_hist
- perm_work_order_detail
- perm_work_order_master
- perm_work_order_routing
- stage_customer
- stage_eam_site
- stage_item
- stage_item_call_list
- stage_item_list1
- stage_item_master
- stage_op_hist
- stage_site
- stage_site_working_calendar
- stage_site1
- stage_transaction_hist
- stage_unit_of_measure
- stage_unit_of_measure4
- stage_wo_mstr
- stage_wod_det
- stage_wr_route

Financials Module - New and Modified Tables

- fact_ap_invoice_snapshot
- fact_ap_voucher_history
- fact_ar_invoice_history
- fact_ar_invoice_snapshot
- fact_ee_ap_supplier_invoice_snapshot
- fact_ee_ar_customer_invoice_snapshot
- stage_ee_ap_supplier_invoice_snap1
- stage_ee_ap_supplier_invoice_snap2
- stage_ee_ap_supplier_invoice_snap3
- stage_ee_ap_supplier_invoice_snap4
- stage_ee_ar_customer_invoice_snap1
- stage_ee_ar_customer_invoice_snap2
- stage_ee_ar_customer_invoice_snap3
- stage_ee_ar_customer_invoice_snap4
- stage_ee_ar_customer_invoice1
- stage_se_ap_invoice_snapshot
- stage_se_ap_snap1
- stage_se_ar_invoice_snapshot
- stage_se_ar_snap1

Operations Module - New and Modified Tables

- stage_inv_bal_item_list
- stage_inv_bal_mth_current4
- stage_inv_bal_mth_date1
- stage_inv_bal_mth_trans1
- stage_inv_mth_end_cst
- stage_inv_mth_end_cst1
- stage_inv_trx_chg_mths
- stage_po_order_snap1

Order Management Module - New and Modified Tables

- fact_om_booking
- fact_om_invoice
- fact_om_order
- fact_om_order_history
- fact_om_order_performance
- fact_om_shipment

EAM Module - New and Modified Tables

- load_eam_domain
- fact_eam_wo_order_history

TM Module - New and Modified Tables

There are no TM module changes for this release.

List of Tables Deleted

For the BI 3.10 release, there were no tables deleted.

Upgrade to QAD BI 3.9 from QAD BI 3.8.1/3.8.2

Note For the Financials module, you must upgrade from 3.8.2 to 3.9. Upgrading from 3.8.1 to 3.9 is not supported. You can install the BI 3.9 Financials upgrade module, but it requires that you truncate your Financials fact table and perform a historical reload. You can upgrade all other modules from 3.8.1 to 3.9.

Upgrade Steps for QAD BI Metadata

Preparation

1. Plan for a couple of hours of down time.
2. Ensure that there are no loads in progress. All jobs must be completed.
3. Put all scheduled jobs On Hold until the upgrade is complete. Shut down the scheduler.

Important Make a backup of the current warehouse database and the current portal database.

Upgrade Steps

1. Unzip the metadata for the Common module and any other modules you have installed. Because the metadata files are now segregated by module, it is only necessary to install the modules that you need.
2. Perform the Common upgrade first. You can do the other upgrades in any order.
3. Run the Common metadata upgrade application.
   a. If you process any of the Order Management (Sales) fact_om_* tables, install the Order Management (OM) upgrade application.
   b. If you process Inventory, Purchase Order, or Capacity Planning/Operations Efficiency tables, install the Operations module upgrade application.
   c. If you process any Financial tables, install the Financials module upgrade application.
   d. If you have an EAM source, install the EAM module upgrade application.
4. Log in to Setup Administrator.
5. Choose Tools|Start Logging. Select a name and location for your log file and save.

6 Load the Common Module Upgrade Application as follows:

a Set your application directory to the new `metadata/common` folder.

b Load the 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.

e 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.

f 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 the box next to *Existing Dimension objects will be* and select Altered.
- On the left, select Dimension View. On the right, click the box next to *Existing Dimension View objects will be* and select Recreated.
- On the left, select Stage Table. On the right, click the box next to *Existing Stage table objects will be* and select Recreated.
- On the left, select Permanent Stage Table. On the right, click the box next to *Existing Permanent Stage table objects will be* and select Altered.
- On the left, select Fact Table. On the right, click the box next to *Existing Fact table objects will be* and select Altered.

g Click OK.

7 Repeat the process for any other modules that you must upgrade.

a **Order Management.** Change the Application Directory to the `metadata /order_management` subdirectory and install the application `upgradeOM`. The upgrade application installations take some time because the load process alters and re-creates various objects.

b **Operations.** Change the Application Directory to the `metadata/operations` subdirectory and install the application `upgradeOps`.

c **Financials.** Change the Application Directory to the `metadata/financials` subdirectory and install the application `upgradeFin`. The upgrade application installations take some time because the load process alters and re-creates various objects.

d **EAM.** Change the Application Directory to the `metadata/eam` subdirectory and install the application `upgradeEAM`.

8 Start the Scheduler. Verify that all jobs are still suspended.

9 If the upgrade affects any customized tables, reapply your customizations. For a list of new and modified tables, see “Data Warehouse Tables Changed” on page 112.

10 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:

- CLEANUP_OM_TABLES
- DAILY_COMMON_PROCESS_CHAINED
- DAILY_OM_PERM_CHAINED
- DAILY_OM_PROCESS_CHAINED
- DAILY_OP_PO_PROCESS_CHAINED
- HIST_OM_PERM_CHAINED
- HIST_OM_PROCESS_CHAINED
- POSTPROCESS_OM_SUMMARY
- PREPROCESS_FIN_TRUNCATE
- PREPROCESS_OM_LOADS
- PREPROCESS_OM_TRUNCATE

11 Make sure the INITIAL_JOB_SETUP_ENABLED parameter is set to Y and run INITIAL_JOB_SETUP to rebuild the daily and historical job chains.

12 Run the upgrade jobs for BI 3.9:

a Edit your INITIAL_JOB_SETUP_CONNECTION_xx_DAILY_MODULES parameter values to ensure that there are no spaces in the comma-separated list. For example, if your DAILY_MODULES parameter currently lists "COM, OM, OP" change it to "COM,OM,OP".

b **Order Management Module.** Run the UPGRADE_381_to_39_ OM job.

c **Operations module.** Run the UPGRADE_381_to_39_OP job.

d **Financials SE Module.** Run the UPGRADE_381_to_39_FINSE job.

e **Financials EE Module.** Run the UPGRADE_381_to_39_ FINEE job. **Note:** This is for Financials EE modules that are at 3.8.2. If you are at 3.8.1 or earlier, you must run a full History load of your Financials module.

13 Run the Daily load.

Note If you have the Order Management module, set the SALES_PROCESS_DAYS to 5000 for the first daily run after the upgrade. This setting ensures that all sales records are processed properly and that the first daily load runs longer than a normal daily load. After the first daily load runs successfully, you can set the SALES_PROCESS_DAYS parameter back to its normal value for your standard processing.

BI Portal Upgrade Procedure

Preparation

1 Shut down Tomcat.

2 Back up the BI Portal database.

Upgrade Steps for QAD BI Portal

1. Back up your BI 3.8.1 Web application by copying the `TOMCAT_HOME\webapps\your_bi_appname` directory to `TOMCAT_HOME\your_bi_appname.save381`.
2. Unzip the new BI 3.9 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 are not overwritten.
3. In the SQL Server Management Studio, run the upgrade script against your BI Portal database: `TOMCAT_HOME\webapps\your_bi_appname\install\sqlserver\v381tov39.sql`
4. Start Tomcat.

Verification

1. Connect to your BI Portal. Click the About link on the login screen and verify that the Version is 3.9.

   Note If necessary, clear the user's browser cache before 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.

Update Portal Model

1. Log in to the Portal as an Administrator.
2. Go to the Administration menu and choose Model Administration. Click Rebuild Model.
3. After the model is 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

1. Update the metadata documentation:
   a. Delete the existing `TOMCAT_HOME\webapps\your_bi_appname\documentation\suite` directory.
   b. Unzip the latest `qadbi-documentation.zip`, on the BI 3.9 Modules physical media or QAD Store product download, into your `TOMCAT_HOME\webapps\your_bi_appname\documentation` directory.
2. To install the latest BI Portal standard objects, copy the files from the BI 3.9 Modules physical media or QAD Store product download 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
3. In the BI Portal, log in as an Administrator.

4 Go to the Administration|Data Migration screen.

5 Click the Import tab.

6 Select the Update existing records check box.

7 Select the File on Server radio button.

8 Enter `qad_common_dashboards.zip` file name in the field and click Import.

9 Repeat for any other module reports and dashboards `.zip` files that are required.

Order Management - Scheduled Orders

Scheduled orders are now included the Order Management fact tables. By default, scheduled orders are included in custom BI Portal queries. If you want to exclude scheduled orders, modify the queries to filter on the scheduled order flag or the order type.

If you have custom BI Portal objects that query the Order History file and you do not want to include scheduled orders in your BI Portal dashboards or reports, enable the scheduled_order_flag field in the BI Portal and filter out scheduled orders in queries.

Data Warehouse Tables Changed

The following section lists the Data Warehouse tables that were changed, added, or deleted.

Note The tables on the Modified list may have been changed structurally. For example:

- Columns added or deleted
- Indexes added or deleted
- Changes made to business display names
- Changes to visibility of particular columns in the BI Portal
- Any combination of these modifications
- The procedure that populates the table has changed

List of Tables Modified

Common Module Modified Tables

- dim_credit_term
- dim_ee_saf
- dim_item
- load_ad_mstr
- load_ar_mstr
- load_cm_mstr
- load_com_mstr
- load_comd_det
- load_ee_DInvoice
- load_ee_Saf
- load_ee_SafConcept
- load_exr_rate
- load_icc_ctrl
- load_pt_mstr
- load_tr_hist
- load_tr_hist_history
- load_vd_mstr
- perm_transaction_hist
- stage_ee_saf
- stage_gl_calendar_year
- stage_transaction_hist

Financials Module Modified Tables

- extract_ee_customer_invoice
- extract_ee_customer_invoice_stage
- stage_ee_ap_supplier_invoice_history2
- stage_ee_ap_supplier_invoice_history3

- extract_ee_posting_line
- extract_ee_posting_saf
- extract_ee_supplier_invoice
- extract_ee_supplier_invoice_stage
- extract_gl_trans_hist
- fact_gl_transaction
- load_ap_mstr
- load_ard_det
- load_ck_mstr
- load_ckd_det
- load_ee_ad_mstr
- load_ee_CInvoice
- load_ee_CInvoiceMovement
- load_ee_Posting
- load_ee_Posting_all
- load_ee_PostingLine_all
- load_ee_PostingSaf
- load_gl_transaction_type
- load_vo_mstr
- perm_ee_customer_invoice
- perm_ee_customer_invoice_stage
- perm_ee_posting_line
- perm_ee_supplier_invoice
- perm_ee_supplier_invoice_stage
- stage_ap_invoice_snaps
- stage_ap_voucher_chk_match
- stage_ar_invoice_snap
- stage_ee_ap_supplier_invoice
- stage_ee_ap_supplier_invoice_hist
- stage_ee_ap_supplier_invoice_hist1
- stage_ee_ap_supplier_invoice_history0
- stage_ee_ap_supplier_invoice_history1
- stage_ee_ap_supplier_invoice_history4
- stage_ee_ap_supplier_invoice_movement_hist
- stage_ee_ap_supplier_invoice1
- stage_ee_ap_supplier_invoice2
- stage_ee_ar_customer_invoice_hist
- stage_ee_ar_customer_invoice_hist1
- stage_ee_ar_customer_invoice_history
- stage_ee_ar_customer_invoice_history0
- stage_ee_ar_customer_invoice_history1
- stage_ee_ar_customer_invoice_history2
- stage_ee_ar_customer_invoice_history3
- stage_ee_ar_customer_invoice_movement_hist
- stage_ee_ar_customer_invoice_stage_hist
- stage_ee_ar_customer_invoice1
- stage_ee_customer_invoice_perm
- stage_ee_customer_invoice_stage_perm
- stage_ee_gl_transaction1
- stage_ee_posting_line_perm
- stage_ee_posting_saf
- stage_ee_supplier_invoice_perm
- stage_ee_supplier_invoice_stage_perm
- stage_gl_balance1
- stage_gl_transaction1
- stage_gl_transaction2
- stage_gl_transaction3
- stage_se_ap_inv_snap
- stage_se_ap_snap1
- stage_se_ap_snap2
- stage_se_ar_inv_snap
- stage_se_ar_snap1
- stage_se_ar_snap2
- stage_se_ar_snap3

Operations Module Modified Tables

- fact_po_order
- fact_inv_transaction
- fact_po_order_history
- fact_po_order_performance
- load_pod_det
- load_shft_det
- load_wod_det
- load_wod_det_hist
- stage_inv_mth_balance
- stage_inv_transaction1
- stage_inv_trx_chg_mth1
- stage_po_credit_term
- stage_po_credit_term1
- stage_po_ord_perf
- stage_po_order_hist
- stage_po_order_hist_initial
- stage_po_order_hist_initial1
- stage_po_order_hist_initial3
- 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

Order Management Module Modified Tables

- extract_om_invoice_line
- extract_om_transaction_hist
- fact_om_booking
- fact_om_invoice
- fact_om_order
- fact_om_order_history
- fact_om_order_performance
- fact_om_shipment
- load_cncu_mstr
- load_idh_hist
- load_idh_hist
- load_ih_hist
- load_sod_det
- perm_om_invoice_line
- stage_om_booking
- stage_om_booking_previous
- stage_om_booking_transaction
- stage_om_booking_transaction1
- stage_om_booking_transaction2
- stage_om_booking1
- stage_om_booking2
- stage_om_booking3
- stage_om_booking4
- stage_om_invoice
- stage_om_invoice_line
- stage_om_invoice_order_line
- stage_om_invoice_transaction
- stage_om_invoice1
- stage_om_invoice2
- stage_om_invoice3
- stage_om_ord_perf
- stage_om_ord_perf_full
- stage_om_ord_perf1
- stage_om_order
- stage_om_order_header
- stage_om_order_header_all
- stage_om_order_inv_header
- 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_order_snap1
- stage_om_order_snap2
- stage_om_order_snap3
- stage_om_order_transaction
- stage_om_order_transaction4
- stage_om_order1
- stage_om_order2
- stage_om_order3
- stage_om_shipment
- stage_om_shipment_transaction
- stage_om_shipment1
- stage_om_shipment2
- stage_om_shipment3

EAM Module Modified Tables

- fact_eam_inv_transaction
- load_eam_account
- load_eam_domain
- load_eam_part_type
- load_eam_squawks
- perm_eam_inv_transaction
- stage_eam_inv_trans_perm
- stage_eam_inv_transaction

Tables Added to Existing Modules

For the BI 3.9 release, no tables were added to the Common and Financial Modules.

Order Management Module Added Tables

- load_abs_mstr
- load_absr_det
- load_idh_hist_cust_ref
- load_idh_hist_pr_list
- load_idh_hist_rlse_id
- load_pc_mstr
- load_sch_mstr
- load_schd_det
- load_tr_hist_ref
- load_tr_hist_ship_id
- perm_customer_consigment
- stage_om_booking_transaction_all
- stage_om_invoice_scheduled_transaction
- stage_om_ord_perf_all
- stage_om_order_transaction7
- stage_om_price_list
- stage_om_schedule_detail
- stage_om_schedule_detail1
- stage_om_schedule_detail2
- stage_om_scheduled_order_perf
- stage_om_scheduled_order_perf1
- stage_om_scheduled_order_transaction
- stage_om_scheduled_order_transaction 1
- stage_om_shipment_requirement
- stage_om_shipment_requirement1
- stage_om_shipment_requirement2

Operations Module Added Tables

- dim_liability_date
- stage_po_order_hist_daily
- stage_po_order_hist_daily1
- stage_po_order_hist_daily2
- stage_po_order_hist_daily3
- stage_po_order_hist_daily4
- stage_po_order_hist_daily5
- stage_po_order_hist_daily6
- stage_po_order_hist_daily7

List of Tables Deleted

For the BI 3.9 release, no tables were deleted.

Upgrade to QAD BI 3.8.1 from QAD BI 3.7

Warning If you have a BI 3.7 or earlier EE Financials module installed, the 3.8.1 upgrade completely deletes all EE Financials tables and re-creates them, and it requires you to do a complete history reload.

Upgrade Steps for QAD BI Metadata

Preparation

1. Plan for a couple of hours of down time.
2. Ensure that there are no loads in progress. All jobs must be completed.
3. Put all scheduled jobs On Hold until the upgrade is complete. Shut down the scheduler.

Important Make a backup of the current warehouse database and the current portal database.

Upgrade Steps

1. Unzip the metadata for the Common module and any other modules you have installed. Because the metadata files are now segregated by module, it is only necessary to install the modules that you need.
2. Perform the Common upgrade first. You can do the other upgrades in any order.

   Warning If you use EE Financials, this upgrade deletes all EE Financials tables and installs an entirely new set.
3. Run the Common metadata upgrade application.
   a. If you process any of the Order Management (Sales) fact_om_* tables, install the Order Management (OM) upgrade application.
   b. If you process Inventory and/or Purchase Order tables, install the Operations module upgrade application.
   c. If you process any Financial tables, install the Financials module upgrade application.
4. Log in to Setup Administrator.
5. Choose Tools|Start Logging. Select a name and location for your log file and save.

6 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.

e 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.

f 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 the box next to *Existing Dimension objects will be* and select Altered.
- On the left, select Dimension View. On the right, click the box next to *Existing Dimension View objects will be* and select Recreated.
- On the left, select Stage Table. On the right, click the box next to *Existing Stage table objects will be* and select Recreated.
- On the left, select Permanent Stage Table. On the right, click the box next to *Existing Permanent Stage table objects will be* and select Altered.
- On the left, select Fact Table. On the right, click the box next to *Existing Fact table objects will be* and select Altered.

g Click OK.

7 Repeat for any other modules that you must upgrade.

a **Order Management.** Change the Application Directory to the `metadata /order_management` subdirectory and install the application `upgradeOM`.

Note Watch for errors here when trying to create an index on fact_om_order. If you see this, create the index manually in the next step.

b **Operations.** Change the Application Directory to the `metadata/operations` subdirectory and install the application `upgradeOps`.

c **Financials.** Change the Application Directory to the `metadata/financials` subdirectory and install the application `upgradeFin`. The upgrade application installations take some time because the load process alters and re-creates various objects.

8 If you have errors during the Order Management module upgrade that a unique index on fact_om_order could not be created, create it manually as non-unique so that the upgrade jobs run properly. Without the index, the OM Postprocess job may take much longer than usual.

a Go to fact_om_order in the BI DWD. Right-click on the table and choose Display Indexes.

b Double-click on fact_om_order_idx_A and clear the Unique check box. Click OK.

c Right-click on fact_om_order_idx_A and choose Create Index.

d If you also have an error on upgrade while attempting to create a fact_om_order_history index, you do not need to address it at this time because the upgrade job drops and re-creates the indexes.

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

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

This enables an extra Tech Notes tab in your Data Warehouse Designer.

10 Start the Scheduler. Make sure that all jobs are still suspended.

11 If the upgrade affects any customized tables, reapply your customizations. For a list of new and modified tables, see “Data Warehouse Tables Changed” on page 112.

12 If any customizations were 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:

- CLEANUP_COMMON_TABLES
- CLEANUP_FIN_TABLES
- CLEANUP_OPERATIONS_TABLES
- CUSTOMER_PROCESS
- DAILY_COMMON_PERM_CHAINED
- DAILY_COMMON_PROCESS_CHAINED
- 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_EE_PERM_EXTRACT
- DAILY_FIN_SE_AP_PROCESS_CHAINED
- DAILY_LOAD_TR_HIST
- DAILY_OM_PERM_CHAINED
- DAILY_OM_PROCESS_CHAINED
- DAILY_OP_INV_PROCESS_CHAINED
- DAILY_OP_PERM_CHAINED
- DAILY_OP_PO_PROCESS_CHAINED
- HIST_COMMON_PERM_CHAINED
- HIST_COMMON_PROCESS_CHAINED
- HIST_EAM_PROCESS_CHAINED
- HIST_FIN_EE_AP_PROCESS_CHAINED
- HIST_FIN_EE_AR_PROCESS_CHAINED
- HIST_FIN_EE_GL_PROCESS_CHAINED
- HIST_FIN_EE_PERM_CHAINED
- HIST_FIN_EE_PERM_EXTRACT
- HIST_FIN_SE_AP_PROCESS_CHAINED
- HIST_LOAD_TR_HIST
- HIST_MODULES_LOAD_FINISH
- HIST_OM_PROCESS_CHAINED
- HIST_OP_INV_PROCESS_CHAINED
- HIST_OP_PERM_CHAINED
- HIST_OP_PO_PROCESS_CHAINED
- INITIAL_JOB_SETUP
- ITEM_PROCESS
- POSTPROCESS_FIN_EE_AP_SUMMARY
- POSTPROCESS_FIN_EE_AR_SUMMARY
- POSTPROCESS_OP_PO_DAILYONLY_SUMMARY
- POSTPROCESS_OP_PO_HISTONLY_SUMMARY
- PREPROCESS_COMMON_DAILYONLY_LOADS
- PREPROCESS_COMMON_EE2010UP_LOADS
- PREPROCESS_COMMON_EEONLY_LOADS
- PREPROCESS_COMMON_LOADS
- PREPROCESS_COMMON_TRUNCATE
- PREPROCESS_FIN_EE_LOADS
- PREPROCESS_FIN_TRUNCATE
- PREPROCESS_OM_DAILYONLY_LOADS
- PREPROCESS_OM_LOADS
- PREPROCESS_OM_TRUNCATE
- PREPROCESS_OP_LOADS
- PREPROCESS_OP_TRUNCATE

13 Set a new parameter for each Source Connection:

INITIAL_JOB_SETUP_CONNECTION_xx_VERSION

Set this to the ERP version number for each source. *Use only the whole number* and do not add any point values. For example, if you have SE 2010, enter 2010. If you have EE 2011.1, enter 2011. Do NOT enter 2011.1. If you have more than two sources, add parameters for each source.

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

15 Run the upgrade jobs for BI 3.8:

   a **Common Module.** Edit your INITIAL_JOB_SETUP_CONNECTION_xx_DAILY_MODULES parameter values to ensure there are no spaces in the comma-separated list. For example, if your DAILY_MODULES parameter currently lists "COM, OM, OP" change it to "COM,OM,OP". Then, run the UPGRADE_37_to_381_QBI_1233 job to populate several new fields in the dim_item table.

   b **Order Management Module.** Run the UPGRADE_37_to_381_ OM job.

   c **Operations module.** Run the UPGRADE_37_to_381_QBI_2171 job to create the dim_po_buyer dimension. Also, if desired, set the new parameter INV_FUTURE_DEMAND_MONTHS from the default value (12) to your desired horizon.

   d **Financials SE Module.** Run the UPGRADE_37_to_381_ FINSE job.

16 Run the Daily load.

17 If you earlier set the index on fact_om_order to non-unique, re-create the index as a Unique index:

   a Go to fact_om_order in the BI DWD and right-click the table and choose Display Indexes.

   b Double-click on fact_om_order_idx_A and check the Unique box. Click OK.

   c Right-click on fact_om_order_idx_A and choose Drop Index.

   d Right-click on fact_om_order_idx_A and choose Create Index.

Data Warehouse Tables Changed

The following section lists the Data Warehouse tables that were changed, added, or deleted.

Note The tables on the Modified list may have been changed structurally. For example:

- Columns added or deleted
- Indexes added or deleted
- Changes made to business display names
- Changes to visibility of particular columns in the BI Portal
- Any combination of these modifications
- The procedure that populates the table has changed

List of Tables Modified

Common Module Modified Tables

- dim_business_relation
- dim_credit_term
- dim_customer
- dim_date_qad
- dim_daybook
- dim_ee_invoice_status
- dim_ee_saf
- dim_item
- dim_layer
- load_ar_mstr
- load_in_mstr
- load_ld_det
- load_pt_mstr
- load_tr_hist
- load_tr_hist_history
- load_tr_hist_list
- load_transaction_type
- perm_code_master
- perm_item_master
- perm_shop_cal
- perm_transaction_hist
- stage_ar_master
- stage_business_relation
- stage_calendar_interval_week
- stage_code_master
- stage_credit_term
- stage_customer
- stage_date_qad
- stage_daybook
- stage_eam_date_qad3
- stage_item
- stage_item_costs
- stage_item_list
- stage_item_master
- stage_item_site
- stage_layer
- stage_project
- stage_se_ar_master
- stage_se_cost_center
- stage_se_customer
- stage_supplier
- stage_transaction_hist

Financials Module Modified Tables

- dim_ap_transaction_type
- dim_ar_transaction_type
- extract_gl_balance
- load_acd_det
- load_bgd_det
- load_gen_led_report
- load_gen_led_report_line
- load_gen_led_report_line_calcs
- load_gen_led_report_line_range
- load_gltr_hist_history
- ods_gen_led_rep
- ods_gen_led_rep_line
- ods_gen_led_rep_line_calcs
- ods_gen_led_rep_line_range
- stage_gen_led_rep_code_comb
- stage_gen_led_rep_line_ran
- stage_gen_led_rep_line_ran1
- stage_gen_led_rep_line_ran2
- stage_gen_led_rep_line_range
- stage_gen_led_report_line
- stage_gen_led_report_line1
- stage_gen_led_report_line2
- stage_gen_led_report_line3
- stage_gen_led_report_line4
- stage_gen_led_report_line5
- stage_gen_led_report_line6
- stage_se_ap_snap1
- stage_se_ar_snap1
- work_ap_change_list
- work_ap_change_list1
- work_ap_change_list2
- work_ap_change_list3
- work_ap_change_list4
- work_ap_change_list5
- work_gl_trans_change_list

Operations Module Modified Tables

- dim_buyer
- fact_inv_mth_balance
- fact_inv_transaction
- fact_op_item_efficiency
- fact_po_order
- fact_po_order_history
- perm_mrp_detail
- permsup_mrp_detail
- stage_inv_bal_mth_date
- stage_inv_mth_balance_hist_calc
- stage_inv_trx_chg_mths
- stage_mrp_calculations
- stage_mrp_calculations1
- stage_mrp_calculations2
- stage_po_order_hist
- stage_po_order_hist1
- stage_po_order_hist5
- stage_po_receipt3

Order Management Module Modified Tables

- extract_om_transaction_hist
- fact_om_booking
- fact_om_invoice
- fact_om_order
- fact_om_order_history
- fact_om_order_performance
- fact_om_shipment
- perm_om_invoice_line
- stage_om_booking_orig_order
- stage_om_booking_transaction
- stage_om_booking_transaction1
- stage_om_booking_transaction2
- stage_om_booking1
- stage_om_invoice
- stage_om_invoice_line_list
- stage_om_invoice_line_ranged
- stage_om_invoice_order_line
- stage_om_invoice_order_list
- stage_om_invoice_transaction
- stage_om_invoice_transaction1
- stage_om_invoice2
- stage_om_invoice3
- stage_om_ord_perf
- stage_om_ord_perf_curr
- stage_om_ord_perf_curr1
- stage_om_ord_perf_date
- stage_om_ord_perf_date1
- stage_om_ord_perf_full
- stage_om_ord_perf_ship
- stage_om_ord_perf1
- stage_om_order
- stage_om_order_header
- stage_om_order_header_all
- stage_om_order_hist_end_date
- stage_om_order_hist_revalue
- stage_om_order_hist_revalue2
- stage_om_order_hist_revalue3
- stage_om_order_hist_revalue4
- stage_om_order_inv_header
- stage_om_order_inv_line
- stage_om_order_inv_line1
- stage_om_order_inv_line2
- stage_om_order_line_all
- stage_om_order_snap
- stage_om_order_snap1
- stage_om_order_snap2
- stage_om_order_snap3
- stage_om_order_transaction
- stage_om_order_transaction1
- stage_om_order_transaction2
- stage_om_order_transaction3
- stage_om_order_transaction4
- stage_om_order_transaction5
- stage_om_order1
- stage_om_order3
- stage_om_shipment
- stage_om_shipment_transaction
- stage_om_shipment1
- stage_om_shipment2
- stage_om_shipment3
- work_om_change_list1

Tables Added to Existing Modules

Common Module Added Tables

- load_com_mstr
- load_comd_det
- load_ee_pti_det
- stage_pti_det

For the 3.8.1 release, all EE financial related tables and jobs were deleted and replaced.

Order Management Module Added Tables

- load_tr_hist_remarks
- stage_om_order_transaction6

Financials Module Added Tables

For the 3.8.1 release, all EE financial related tables and jobs were deleted and replaced.

Operations Module Added Tables

- dim_po_buyer
- stage_po_buyer
- stage_po_buyer_code_master
- stage_po_buyer_po_header

List of Tables Deleted

For the 3.8.1 release, all EE financial related tables and jobs were deleted and replaced.

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 in progress. All jobs must be completed.
3. Put all scheduled jobs On Hold until the upgrade is complete. Shut down the scheduler.
4. **IMPORTANT: Make a backup of the current warehouse database and the current portal database.**
5. Before you begin, note 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, place any custom jobs at the end of the chains. Take note of your custom jobs here and ensure that all jobs appear in the rebuilt job chain.

Important This upgrade replaces 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.
7. At the “Define FileGroups” dialog, click OK.
8. 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 are not overwritten.
3. In the SQL Server Management Studio, run the upgrade script: *TOMCAT_HOME\webapps\your_bi_appname\install\sqlserver\v36tov37.sql*.
4. Start Tomcat.

   Note If necessary, clear the user's browser cache before 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

1. 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 you need.

   - Perform the Common upgrade first. You can do the other upgrades in any order.
   - All users must 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.
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* 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. Change the value if necessary:

- On the left, select Dimension. On the right, click the box next to *Existing Dimension objects will be* and select Altered.
- On the left, select Dimension View. On the right, click the box next to *Existing Dimension View objects will be* and select Recreated.
- On the left, select Stage Table. On the right, click the box next to *Existing Stage table objects will be* and select Recreated.
- On the left, select Permanent Stage Table. On the right, click the box next to *Existing Permanent Stage table objects will be* and select Altered.
- On the left, select Fact Table. On the right, click the box next to *Existing Fact table objects will be* and select Altered.
- Click OK.

5 Repeat for any other modules that you must upgrade.

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

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

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

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

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

7 Start the Scheduler. Make sure that all jobs are still suspended.

8 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 126.

9 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
- 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

- 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
- 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

10 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 parameter 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 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 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 fails and indicates that you must increase this number. The default value is 3.
- **TR_HIST_MAX_VALUE.** This parameter is the maximum number allowed for the tr_hist.tr_trnbr field, or in other words, the number after which the tr_trnbr rolls 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 parameter 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 forces the tr_hist extraction to scan the whole table for records that were processed within the last TR_HIST_PROCESS_DAYS. The default value is 5000.
- **TR_HIST_PCT_OF_MAX_REF.** This parameter 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 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.

11 If you already installed the BI 3.6.1 or BI 3.6.2 patch, skip to Step 12.

If you are upgrading directly from BI 3.6, run the Upgrade jobs for the tr_hist.tr_trnbr rollover 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.

12 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.

13 There are two new parameters for each connection:

   - INITIAL_JOB_SETUP_CONNECTION_xx_DAILY_MODULES
   - INITIAL_JOB_SETUP_CONNECTION_xx_HIST_MODULES.

   The upgrade installs 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. The INITIAL_JOB_SETUP job uses these parameters to determine the Daily and History job chains, respectively. By default the parameters for the 01 and 02 connections contain all ERP modules (COM, OM, OP, FIN). Remove any modules that you do not use from this list.

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

15 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.

16 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 takes longer than normal because all inventory transactions are being reprocessed to correct errors and the fact_inv_mth_balance table is being repopulated.

17 Run the Daily load. If you had set the INV_PROCESS_DAYS value to a large number, 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 references 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. 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 were changed, added, or deleted. The tables 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 the 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_account_currency
- dim_approver
- dim_base_currency
- dim_buyer
- dim_channel
- dim_corporate_currency
- dim_cost_center
- dim_credit_term
- dim_currency
- dim_customer
- dim_customer_bill_to
- dim_customer_ship_to
- dim_customer_sold_to
- dim_discount_cost_center
- dim_discount_sub_account
- dim_due_date
- dim_effective_date
- dim_entered_date
- dim_entity
- dim_item
- dim_last_receipt_date
- 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
- dim_shipment_date
- dim_site
- dim_sub_account
- dim_supplier
- dim_transaction_currency
- dim_transaction_type
- dim_unit_of_measure
- load_ct_mstr
- load_ctd_det
- load_cu_mstr
- load_ee_Address
- load_ee_BudgetGroup
- load_ee_Company
- load_ee_CostCentre
- load_ee_Creditor
- load_ee_Currency
- load_ee_Debtor
- load_ee_Division
- load_ee_DomainProperty
- load_ee_Domains
- load_ee_DomainSharedSet
- load_ee_PaymentCondition
- 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_credit_term
- stage_currency
- stage_customer
- stage_customer_ar_list
- stage_customer_bill
- stage_customer_bill1
- stage_customer_dim_bill
- stage_customer_dim_ship
- stage_customer_dim_sold
- stage_customer_invoice_list
- stage_customer_master
- stage_customer_sales_list
- stage_customer_ship
- stage_customer_ship1
- stage_customer_sold
- stage_customer_sold1
- stage_date_qad
- stage_domain
- stage_eam_entity
- stage_ee_account
- stage_ee_address
- stage_ee_ic_creditor
- 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_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_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_trans_scrapi
- stage_op_transaction
- stage_op_transaction_merge
- stage_op_transaction0
- stage_op_transaction1
- 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_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
- stage_om_booking1
- stage_om_invoice_line_list
- stage_om_invoice_line_ranged
- stage_om_order_line
- stage_om_order_line_all
- stage_om_order_snap

- 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_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_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
- load_ee_State
- load_ee_Usr
- 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_invoice_status
- 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
- 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 that none of the loads are in progress. **All jobs must be completed**.
3. Put all scheduled jobs On Hold until the upgrade is complete. Shut down scheduler.
4. Make 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 do not have Sales or 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 replaces 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. Unzip the metadata for the Common module, plus any other modules you have installed, into a directory. The Metadata files are now segregated by module, so you only install the modules that you need.
   - All users must 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.

Perform the Common upgrade first; you can do the other upgrades 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 asks "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 the box next to "Existing Dimension objects will be" and select Altered.
- Select Dimension View on the left - on the right, click the box next to "Existing Dimension View objects will be" and select Recreated.
- Select Stage Table on the left - on the right, click the box next to "Existing Stage table objects will be" and select Recreated.
- Select Permanent Stage Table on the left - on the right, click the box next to "Existing Permanent Stage table objects will be" and select Altered.
- Select Fact Table on the left - on the right, click 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 before recreating it; if the index is new, the drop fails. Note any errors in the upgrade output other than the index errors. Consult QAD Support before continuing.

6 Repeat for any other modules that require upgrading.

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 metdata/operations subdirectory, and install application **upgradeOps v36**. You may see an error here about the inability to create index dim_po_line_status_idx_A. This error 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 take some time because the load process alters/re-creates various objects. If any of the tables that were modified had included customizations, reapply those customizations to your upgraded tables when you have installed all pertinent upgrade applications. Verify that all other customizations are not affected.

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

9 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 safely allows 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 approximately 90 days. However, if backdating is never more than 45 days, enter 45. The 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 were modified between Today and Today minus COM_PROCESS_DAYS. The 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 the 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 safely allows 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 approximately 90 days. However, if backdating is never more than 45 days, enter 45. The 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 days open or closed between Today and Today minus OP_LOAD_LOOKBACK_DAYS. The default value is 20.
- OP_MOVING_AVG_WEEKS: The number of weeks on which to base the Moving Average calculations. The 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. The default value is B for both.

- OP_PROCESS_DAYS: The number of days of OP to extract and process from permanent staging tables. The 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 produces 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 is rewritten 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 before running the Operations Upgrade job; after the job completes, the parameter is removed, ensuring that 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, reapply your customizations. You can find a list of new and changed tables at the end of the Upgrade instructions.

18 Resume normal processing.

To Add a New Source

If you want to add a 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 an INITIAL_JOB_SETUP_CONNECTION_xx_RUN (if it does not exist) and set its value to N.

5 Add an INITIAL_JOB_SETUP_CONNECTION_xx_TYPE (if it does not exist). Set its value to EE or SE, depending on your type of financials system for that source.

6 Make sure that 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 were 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 that 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` to make 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 are not overwritten.

3 Start Tomcat.

Verification

1 Connect to your BI Portal. Click 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 before reconnecting to the BI Portal to ensure that they are running the latest version of the application.

Update Portal Model

1. Log in to the Portal as an Administrator.
2. Go to the Administration menu and choose Model Administration. Click Rebuild Model.
3. After the model is rebuilt, if you have any custom data models, open each model and save it to ensure that it picks up the 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 physical media or QAD Store product download) 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 physical media or QAD Store product download 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 the Import tab.
4. Check the "Update existing records" box.
5. Click the radio button in front of "File on server."
6. Enter the file name `qad_common_dashboards.zip` in the box and click "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 were changed, added, or deleted. The tables 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 the visibility of particular columns in the BI Portal, or any combination of these modifications.

Tables Modified

Common Module Modified Tables

- dim_account
- dim_credit_term
- dim_customer
- dim_customer_bill_to
- dim_customer_ship_to
- dim_customer_sold_to
- dim_date_qad
- 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_cu_mstr
- load_dom_mstr
- load_ee_BusinessRelation
- load_ee_Creditor
- load_ee_Currency
- load_ee_Debtor
- load_ee_Domains
- 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_shop_cal
- load_si_mstr
- load_sp_mstr
- load_SYSCOLUMNS
- load_tr_hist
- load_tr_hist_history
- load_tr_hist_list
- 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_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_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

- load_bdg_det
- load_ck_mstr
- load_ckd_det
- load_ee_PostingHist
- load_ee_PostingLine
- 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
- fact_inv_transaction
- fact_po_order
- fact_po_order_history
- fact_po_order_performance
- fact_po_order_receipt
- load_inv_trans_type_cnterr
- load_inv_trans_type_cnttr
- 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_inv_trans_type_sales
- load_inv_trans_type_supplier
- load_inv_transaction_type
- load_is_mstr
- load_ld_det
- 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_cus3
- stage_inv_transaction_con_cus4
- stage_inv_transaction_con_cus5
- stage_inv_transaction_con_sup
- stage_inv_transaction_con_sup1
- 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_perf1
- 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_oee_periods
- dim_production_line
- dim_release_date
- dim_start_date
- dim_work_center
- dim_work_center_machine
- extract_op_oee_dates
- extract_op_transaction_hist
- fact_op_equipment_effectiveness
- fact_op_item_efficiency
- fact_op_transaction
- fact_wo_order
- fact_wo_routing_transaction
- load_cal_det
- load_dpt_mstr
- load_ln_mstr
- load_op_hist
- load_shft_det
- load_wc_mstr
- load_wo_mstr
- load_wo_mstr_hist
- load_wod_det
- load_wod_det_hist
- load_wr_route
- load_wr_route_hist
- perm_op_hist
- perm_work_center_master
- perm_work_order_detail
- perm_work_order_master
- perm_work_order_routing
- stage_calendar_period
- stage_exception_period_hrs
- stage_op_equip_effect
- stage_op_equipment_effectiveness
- stage_op_hist
- stage_op_item_efficiency
- stage_op_item_efficiency1
- stage_op_item_efficiency2
- stage_op_item_moving_avg
- stage_op_oee_periods
- stage_op_oee_periods_month
- stage_op_oee_periods_week
- stage_op_prod_line_period_hrs1
- stage_op_prod_line_period_hrs2
- stage_op_prod_line_wc_period
- stage_op_resource_avail_hrs
- stage_op_resource_avail_keys
- stage_op_tr_hist_rjct_wo
- stage_op_tr_hist_rjct_wo1
- stage_op_trans_scrapi
- stage_op_transaction
- stage_op_transaction_merge
- stage_op_transaction0
- stage_op_transaction1
- stage_op_transaction2
- stage_op_work_center_period_hrs1
- stage_op_work_order_detail_scrap
- stage_op_work_order_detail_scrap2
- stage_op_work_order_lot
- stage_op_work_order_lot_scrap
- stage_prod_line_hrs
- stage_prod_line_period_hrs
- stage_prod_line_period_hrs_ma
- stage_wo_mstr
- stage_wo_order
- stage_wo_order0
- stage_wo_routing_transaction
- stage_wo_routing_transaction0
- stage_wod_det
- stage_work_center_period_hrs
- stage_work_center_period_hrs_ma
- stage_work_center_site
- stage_work_center_site_hrs
- stage_wr_route
- work_op_wo_routing_costs
- 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`* to make 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 are not 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 before 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 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 in progress. All jobs must be completed.
3. Put all scheduled jobs On Hold until the upgrade is complete. Shut down scheduler.
4. Make 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 replaces 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. Completely reload the fact_po_receipt, fact_po_order_history, and fact_po_order_performance tables from the permanent Data Stores. Corrections made to the SE Financials module require the reprocessing of SE AP and AR data. Also reload the fact_om_order_history table. 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 prompts you to proceed; click OK.
   b. The next dialog box is for Application Load Properties:
      - Select Dimension on the left; on the right, click the box next to Existing Dimension objects will be and select Altered.
      - Select Dimension View on the left; on the right, click the box next to Existing Dimension View objects will be and select Recreated.
      - Select Stage Table on the left; on the right, click the box next to Existing Stage table objects will be and select Recreated.
      - Select Permanent Stage Table on the left; on the right, click the box next to Existing Permanent Stage table objects will be and select Altered.
      - Select Fact Table on the left; on the right, click the box next to Existing Fact table objects will be and select Altered.
      - Click OK.
4. The upgrade application installation takes some time because the load process alters/re-creates 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 situation is OK because they are new indexes that do not yet exist in the database. The upgrade script tries to drop each index before recreating it; if the index is new, the drop fails. Note any other errors in the upgrade output other than those index errors listed above. Consult QAD Support before continuing.
6. Start the Scheduler, but make sure that all jobs are still suspended.

7 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. The 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 is still 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 is still considered In Full. The 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 is still considered In Full. If the quantity received is less than the quantity ordered minus10%, the order line is not considered In Full.
- PO_RET_QTY_TOLERANCE: This parameter is the Percentage of Returns Quantity used for calculating the Defect Rate. The 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 add new records and not update previously calculated records. The default value is N.
- SALES_SNAPSHOT_FREQUENCY: Set to M for monthly snapshots, W for weekly snapshots, or B for both monthly and weekly. The 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 produces 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 were 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 are created):

- DAILY_COMMON_PROCESS_sourcename
- DAILY_PERM_sourcename (CAUTION: do NOT deleted 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 is 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 any jobs that occurred after the snapshot jobs (which you noted in the Preparation section) to the end of the DAILY_ and HIST_ load lists.

13 This is a good time to review the Parameters and their definitions in the BI 3.5 Installation Guide. Check your parameter settings; you will want to consider changing some, such as various PROCESS_DAYS and SNAPSHOT_NUM_DAYS values, to make sure that 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 that you truncated earlier. Run these for each source. Depending on the amount of history data in your system, these jobs can 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, 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.

The upgrade is complete. Resume normal processing.

To Add a New Source

If you want to add a 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 an INITIAL_JOB_SETUP_CONNECTION_xx_RUN (if it does not exist) and set its value to N.

5 Add a INITIAL_JOB_SETUP_CONNECTION_xx_TYPE (if it does not exist). Set its value to EE or SE, depending on your type of financials system for that source.

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 were 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 that 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 Rebuild Model.

After the model is 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 physical media or QAD Store product download) 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 physical media or QAD product download 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 the Import tab.
4. Select the Update existing records box.
5. Click the radio button in front of File on server.
6. Enter `BI35_deprecated_items.zip` in the box and click Import.
7. Repeat for the `qad_standard_items.zip` file.

Chapter 4

Onsite Data Validation

This chapter includes the following information about validating that BI is working properly:

Introduction 146

EAM Validation Reports 146

Financials EE Accounts Payable Validation Reports 147

Financials EE Accounts Receivable Validation Reports 148

Financials EE General Ledger Validation Reports 149

Financials SE Accounts Receivable Validation Reports 150

Financials Accounts Payable Validation Reports 151

Inventory Validation Reports 151

Overall Equipment Efficiency Reports 152

Order Management Validation Reports 153

Purchasing Validation Reports 155

Automated Order Management Validation 155

Running Validation of BI Against ERP Reports 156

Introduction

Onsite data validation reports are used to validate the target QAD BI fact table data against the source QAD Enterprise Applications data. This validation is recommended at initial installation and for upgrades so that business users can confirm that the ETL jobs have completed and are operating as expected.

Run the report on the QAD EA source system according to the instructions and then run the corresponding report in QAD BI. Compare the totals and if any numbers do not match, look for the specific discrepancies.

EAM Validation Reports

These reports are located in the Reports menu, under the QAD Standard|EAM|Plant Maintenance|EAM Validation folder.

They are also available as links from the EAM Administration dashboard.

EAM Equipment Downtime Transaction Table

Report name	All Downtime
Usage	Compares EAM Equipment Downtime Transaction table (fact_eam_equip_down_tran) to the All Downtime browse. This report takes the Site and a range of Start to End Months as parameters. Set up a filter in the EAM browse with the Start Date within a range that matches the range selected in the BI report. Set the EAM Browse Filter's Site to the Site used in the BI report.
Validation	• Down Time (BI) = Hours Down (EAM) • Cost Equipment Down (BI) = Cost (EAM) If you export the EAM results to Excel, you can use the SUM function on the columns so that you do not have to do line-by-line matching.

EAM Inventory Transaction Table

Report name	Inventory History
Usage	Compares EAM Inventory Transaction table (fact_eam_inv_transaction) to the Inventory Transaction History browse. This report takes the Site and a range of Start to End Months as parameters. Set up a filter in the EAM browse with the Effective Date within a range that matches the range selected in the BI report. Set the EAM Browses Site to the Site used in the BI report.
Validation	• Quantity (BI) = Quantity (EAM) • Total Cost (BI) = Total Cost (EAM) If you export the EAM results to Excel, you can use the SUM function on the columns so that you do not have to do line-by-line matching.

EAM Labor Transaction Table

Report name	Labor History
Usage	Compares EAM Labor Transaction table (fact_eam_labor_transaction) to the Labor History browse. This report takes the Site and a range of Start to End Months as parameters. Set up a filter in the EAM browse with the Date within a range that matches the range selected in the BI report. Set the EAM Browses Site to the Site used in the BI report.
Validation	Total Cost (BI) = Total Cost (EAM) The Time should also match. Time in the EAM Browse is in hh:mm notation whereas Time in the BI Time column is in number of hours. For example, 1:30 in the EAM browse is equal to 1.5 hrs in the BI report. If you export the EAM results to Excel, you can use the SUM function on the columns so that you do not have to do line-by-line matching.

Financials EE Accounts Payable Validation Reports

These reports are located in the Reports menu, under the QAD Standard|Financials|EE Accounts Payable|EE AP Validation folder.

They are also available as links from the Financials Administration dashboard.

EE AP Supplier Invoice Fact Table

Report name	28.1.1.3 EE AP Supplier Invoice Fact Validation
Usage	Compares EE AP Supplier Invoice Fact table (fact_ee_ap_supplier_invoice) to the 28.1.1.3 Supplier Invoice View browse. In the browse, set the following: • Open filter equal to 'yes' • Voucher filter NOT equal to "0"(exclude unmatched vouchers) • Year/GL Period filter to a range for the same Financial Month range you use in this report; for example, for all of 2013, use the Year/GL Period range of 201301 to 201312.
Validation	Base Open Amount (BI) = BC Balance(EE)

EE AP Supplier Invoice History Fact Table

Report name	28.18.2 EE AP Supplier Invoice History Validation
Usage	Compares EE AP Supplier Invoice History Fact table (fact_ee_ap_supplier_history) to the 28.18.2 Supplier Invoice Activity View browse. In the browse, set the following: • Type filter equal to 'initial' to verify the BI Base Open Amount or 'activity' to verify the BI Base Allocated Amount • Invoice Year/GL Period filter to a range for the same Financial Month range you use in this report. For example, for all of 2013, use the Year/GL Period range of 201301 to 201312. **Note:** You cannot query further than your initial EE_AP_HISTORY_DATE_MIN parameter setting.
Validation	• **Initial**. Base Amount (BI) = BC Debit - BC Credit • **Activity**. Base Allocated Amount (BI) = (BC Debit - BC Credit)

EE AP Supplier Invoice Snapshot Fact Table

Report name	28.17.10 EE AP Supplier Invoice Snapshot Validation
Usage	Compares EE AP Supplier Invoice Snapshot Fact table (fact_ee_ap_invoice_snapshot) to the 28.17.10 Supplier Aging Analysis History report. In the report, set the Aging Type to "Months", and To GL Year/To GL Period filter and Date for Aging Calculation to the same GL period you choose in this report. **Note:** Items that are open but are older than your AP_EE_HISTORY_DATE_MIN or your AP_EE_SNAPSHOT_HIST_START_DATE do not appear in the BI report. If any of these items are in your EE report, you must subtract them from the total on the EE report side.
Validation	Base Open Amount (BI) = Open Amount (BC). The aging 'buckets' (for example, 1 month overdue, 2 months overdue, and so on) should also match. **Note:** For older versions of the EE report, these may not be correct and may not match.

Financials EE Accounts Receivable Validation Reports

These reports are located in the Reports menu, under the QAD Standard|Financials|EE Accounts Receivable|EE AR Validation folder.

They are also available as links from the Financials Administration dashboard.

EE AR Customer Invoice Fact Table

Report name	27.1.1.3 EE AR Customer Invoice Fact Validation
Usage	Compares EE AR Customer Invoice Fact table (fact_ee_ar_customer_invoice) to the 27.1.1.3 Customer Invoice View browse. In the browse, set the following: • Open filter equal to 'yes' • Voucher filter NOT equal to "000000000" (exclude unmatched vouchers) • Year/GL Period filter to a range for the same Financial Month range you use in this report; for example, for all of 2013, use the Year/GL Period range of 201301 to 201312.
Validation	Base Open Amount (BI) = BC Balance(EE)

EE AR Customer Invoice History Fact Table

Report name	27.18.2 EE AR Customer Invoice History Validation
Usage	Compares EE AR Customer Invoice History Fact table (fact_ee_ar_customer_history) to the 27.18.2 Customer Invoice Activity View browse. In the browse, set the following: • Type filter equal to 'initial' to verify the BI Base Open Amount or 'activity' to verify the BI Base Allocated Amount • Invoice Year/GL Period filter to a range for the same Financial Month range you use in this report. For example, for all of 2013, use the Year/GL Period range of 201301 to 201312. **Note:** You cannot query further than your initial EE_AR_HISTORY_DATE_MIN parameter setting.
Validation	• **Initial**. Base Amount (BI) = BC Debit - BC Credit • **Activity**. Base Allocated Amount (BI) = (BC Debit - BC Credit)

EE AR Customer Invoice Snapshot Fact Table

Report name	27.17.7 EE AR Customer Invoice Snapshot Validation
Usage	Compares EE AR Customer Invoice Snapshot Fact table (fact_ee_ar_customer_snapshot) to the 27.17.7 Customer Aging Analysis History report. In the report, set the Aging Type to "Months", and To GL Year/To GL Period filter and Date for Aging Calculation to the same GL period you choose in this report. **Note:** Items that are still open but are older than your AR_EE_HISTORY_DATE_MIN or your AR_EE_SNAPSHOT_HIST_START_DATE do not appear in the BI report. If any of these items are in your EE report, subtract them from the total on the EE report side.
Validation	Base Open Amount (BI) = Open Amount (BC). The aging 'buckets' (for example, 1 month overdue, 2 months overdue, and so on) should also match. **Note:** For older versions of the EE report, these may not be correct and may not match.

Financials EE General Ledger Validation Reports

These reports are located in the Reports menu under the QAD Standard|Financials|EE General Ledger|EE GL Validation folder.

EE GL Balance Fact Table

Report name	EE GL Balance Fact Validation
Usage	Compares EE GL Balance Fact Table (fact_ee_gl_balance) to the "Trial Balance View" in the ERP, matching Entity, Balance Year, and Balance Period. In the ERP, search on Entity Code equals any value and Year/GL Period equals any value, such as, 201209.
Validation	• Base Open Amount (BI) = Opening Balance BC(EE) • Base Closing Amount (BI) = Closing Balance BC(EE)

EE GL Transaction Fact Table

Report name	EE GL Transaction Fact Validation
Usage	Compares EE GL Transaction Fact Table (fact_ee_gl_transaction) to "25.15.2.10 GL Transactions View Extended " in the ERP, matching Entity, Posting Year, and Posting Period. In the ERP, search on Entity Code equals any value and GL Period equals any value, such as, 201209.
Validation	• Debit Base Posting Amount (BI) = BC Debit (EE) • Credit Base Posting Amount (BI) = BC Credit (EE)

CFO Dashboard Fact Table

Report name	EE GL CFO Balance Fact Validation
Usage	Compares CFO Dashboard Fact Table (fact_gen_led_report_line) to the "Trial Balance View" in the ERP, matching Entity, Balance Financial Month (YYYYMM) and report line. In the ERP, search on Entity Code equals any value and Year/GL Period equals any value, such as, example 201209. In the Portal Report, the correct Report Line must also be selected. Each CFO dashboard is configured to match the General Ledger Chart of Accounts into groupings that are specific to a business. The Report Line determines the groupings.
Validation	Base Closing Amount (BI) = Closing Balance BC (EE)

Financials General Ledger Validation Reports

These reports are located in the Reports menu, under the QAD Standard|Financials|SE General Ledger|SE GL Validation folder. They are also available as links from the Financials Administration dashboard.

GL Transaction Validation

Report name	25.13.21 GL Transaction Validation
Usage	Compares GL Transaction Fact table to the 25.13.21 Transaction Register.
	In the source (ERP) report, set Entity range to/from to a single Entity. Set Beginning Effective Date start to the first day of a financial period. Set the Ending Effective Date to the last day of a financial period.
Validation	Base Debit Amount (BI) total = Debit Amount total (ERP)
	Base Credit Amount (BI) total = Credit Amount total (ERP)

GL Trial Balance Validation

Report name	25.15.4 GL Trial Balance Validation
Usage	Compares GL Balance Fact table (fact_gl_balance) to the 25.15.4 Trial Balance Summary.
	In the source (ERP) report, set Entity range to/from to a single Entity. Set Beginning Date start to the first day of a financial period. Set the Ending Date to the last day of a financial period.
	Select Summarize Sub-Accounts and Summarize Cost Centers check boxes.
Validation	Base Close Amount (BI) = Ending Balance (ERP)

Financials SE Accounts Receivable Validation Reports

These reports are located in the Reports menu, under QAD Standard|Financials|SE Accounts Receivable|SE AR Validation. They are also available in the Dashboards menu under QAD Standard|Financials|Administration and as links from the Financials Administration dashboard.

SE AR Invoice Fact Table

Report name	27.6.6 SE AR Payment Validation Report
Usage	Compares BI SE AR Payment Fact table to ERP's report 27.6.6 Payment Register. Choose an Effective Date range in the ERP report to cover the same dates as are covered by the Financial Months start and end selected in the BI report.
Validation	After verifying the workspace/entity for the report, leave all search criteria fields in their default blank values to get a complete list of all customers with open AR invoices. You can compare the Total Open Amount from the footer of each customer Bill To with the BI value open_amount_base.
	Base Report Total (in ERP) Equals AR Payment Fact - Base Amount (in BI)

SE AR Invoice Fact Table

Report name	27.14 SE AR Invoice Validation Report

Usage	Compares SE AR Invoice Fact Table (fact_ar_invoice) to "27.14 Customer Account Statement Print" in the ERP, matching workspace and entity and customer bill to number.
Validation	After verifying the workspace/entity for the report, leave all search criteria fields in their default blank values to get a complete list of all customers with open AR invoices. You can compare the Total Open Amount from the footer of each customer Bill To with the BI value open_amount_base. Total Amount Open (in ERP) = AR Invoice Fact - Base Open Amount (in BI)

Financials Accounts Payable Validation Reports

These reports are located in the Reports menu, under QAD Standard|Financials|SE Accounts Payable|SE AP Validation folder.

They are also available as links from the Financials Administration dashboard.

AP Voucher Fact Table

Report name	28.9.11 SE Payment Validation
Usage	Compares SE AP Check Fact table to the 28.9.11 Payment Browse. In the Browse, set the Effective Date From and To values to a range for the same Financial Month range you use in this report (i.e., for all of 2013, use the effective date range of 201301 to 201312); you can add a Sum to the Amount field to compare totals.
Validation	Transaction Amount (BI) = - Amount(SE)

AP Voucher Fact Table

Report name	28.3.1 SE Voucher Register Validation
Usage	Compares AP Voucher Fact table (fact_ap_voucher) to the 28.3.1 Voucher Register. In the source (ERP) report, set Entity range to/from to a single Entity. Set Effective Date start to the first day of a calendar month. Set the Effective end to the last day of a calendar month. Select Print Unconfirmed check box. Set Summary/Detail to Summary. Accept all other defaults.
Validation	Base Amount Total (BI) = Amount (ERP) Base Amount Applied (BI) = Amount Applied (ERP)

Inventory Validation Reports

These reports are located in the Reports menu, under the QAD Standard|Operations|Inventory|Inventory Validation folder.

They are also available as links from the Operations Administration dashboard.

Inventory Balances Fact Monthly Snapshot Table

Report name	3.21.2 Inventory Transaction Validation
Usage	Compares Inventory Transaction Fact table to the Transactions by Item Browse. In the source (ERP) browse, add a filter for Date range to the first day of a calendar month through the last day of a calendar month. The range may span multiple months. Also add a filter for the same Item Number used in the BI validation report.
Validation	Quantity (BI) = Loc Quantity Changed (ERP)

Inventory Balances Fact Monthly Snapshot Table

Report name	1.5.23 Inventory Balances Fact Monthly Snapshot Validation
Usage	Compares Inventory Balances Fact Monthly Snapshot table (fact_inv_mth_balance table) to the 1.5.23 Inventory Valuation as of Date report. This report takes the Domain, a Product Line, and the End of Month date (as of Date) as parameters. Set the following: • ERP report's Effective Date to the BI report's End of Month Date • Set the Project Line to and from values to the Product Line used in the BI report. Do not include zero quantity. The BI report does not display the 0 quantity records. Include the Customer and Supplier Consigned records. • Choose the Base Currency for the Domain or the Corporate currency. The BI report displays both.
Validation	End Value OH (BI) = Report Total (ERP) • If you choose the Domain's Base Currency Code in the ERP report, use the Base End Value OH in the BI report. • If you choose the corporate currency in the ERP report, use the Corp End Value OH in the BI report. **Note:** If you want to perform your validation using a range of Product Lines or a range of item numbers, you can copy the query used by this report and modify it to search using your preferred criteria. You can then make a report from that query.

Overall Equipment Efficiency Reports

These reports are located in the Reports menu, under QAD Standard|Operations|Overall Equipment Efficiency|OEE Validation folder.

They are also available as links from the Operations Administration dashboard.

Work Order Validation

Report name	16.2 Work Order Validation
Usage	Compares Work Order Fact table to the Work Order Browse. In the source (ERP) browse, add a filter for Order Date range to the first day of a calendar month through the last of a calendar month. The range may span multiple months. Also add a filter for the same Item Number used in the BI validation report.
Validation	Quantity Open (BI) = Quantity Open (ERP) Quantity Completed (BI) = Quantity Completed (ERP) Quantity Rejected (BI) = Quantity Rejected (ERP)

Work Order Routing Validation

Report name	16.13.14 Work Order Routing Validation
Usage	Compares Work Order Routing Transaction Fact table to the Work Order Routing Browse. In the source (ERP) browse, add a filter for Start Date range to the first day of a calendar month through the last of a calendar month. The range may span multiple months. Also add a filter for the same Item Number used in the BI validation report.
Validation	Quantity Ordered (BI) = Quantity Ordered (ERP) Quantity Completed (BI) = Quantity Completed (ERP)

Operation Transaction Validation

Report name	16.20.13.8 Operation Transaction Browse
Usage	Compares Operations Transaction Fact table (fact_op_transacton) to the Operation Transaction Browse (sfbr001.p). In the source (ERP) browse, add a filter for Date range to the first day of a calendar month through the last of a calendar month. The range may span multiple months. Filter out the “MOVE” transaction type; for example, add a filter where Type does not equal MOVE.
Validation	Quantity Total (BI) = Quantity (ERP) The Quantity Total (BI) = the SUM of quantity_good + quantity_rejected + quantity_rework + quantity_scrap

Order Management Validation Reports

These reports are located in the Reports menu, under QAD Standard|Order Management|Order Management Validation Reports folder.

Compares OM Booking Fact Table

Report name	7.15.13 OM Booking Validation Report
Usage	Compares OM Booking Fact table (fact_om_booking, standard [non-scheduled] orders) to 7.15.13 Booking Transaction Browse. Set the Effective Date Range in the browse to cover the month or months chosen in the BI report range. Set the Item Number filter in the browse to the same item number selected in the BI report. You can add SUMs for the Quantity Required and Net Price to the browse to allow for comparison of the totals in the BI report against the sums in the browse.
Validation	Add a SUM summary to the Qty Required and Price columns to compare to the BI Report total, or you can compare line-by-line. **Note**: If you want to perform your validation on a range of items, you can copy the query used by this report and modify it to search using your preferred criteria. You can then make a report from that query. However, if you include multiple items in the browse, you cannot add a summary, and you would have to compare line-by-line.

OM Booking Fact for Scheduled Orders Table

Report name	7.5.10 OM Scheduled Order Booking Validation Report
Usage	Compares OM Booking Fact for Scheduled Orders table (fact_om_booking, scheduled orders) to 7.5.10 Schedule Report. In the Schedule Report, set the Create Date range to the dates that cover the month range chosen in the BI report. Set the Schedule Type to 3 and Active/History to Active.
Validation	Line-by-line comparison **Note:** EE environments can also use the Required Ship Schedule Browse.

OM Shipping Fact Table

Report name	7.15.17 OM Shipment Validation Report
Usage	Compares OM Shipping Fact table (fact_om_shipment) to 7.15.17 Shipping Transaction Browse. Set the Effective Date Range in the browse to cover the month or months chosen in the BI report range. Set the Item Number filter in the browse to the same item number selected in the BI report.
Validation	Add a SUM summary to the Loc Qty Chg and Price columns to compare to the BI Report total, or you can compare line-by-line. **Note:** If you want to perform your validation on a range of items, you can copy the query used by this report and modify it to search using your preferred criteria. You can then make a report from that query. However, if you include multiple items in the browse, you cannot add a summary, and you would have to compare line-by-line.

OM Invoice Fact Table

Report name	7.13.7 OM Invoice Validation Report
Usage	Compares the OM Invoice Fact table (fact_om_invoice) to the 7.13.7 Invoice History Browse. Set the Invoice Date Range in the browse to cover the month or months chosen in the BI report range. Set the Item Number filter in the browse to the same item number selected in the BI report.
Validation	Add SUMs for the Quantity Order/Invoiced/Backordered (in EE, Invoice Amount and BC Invoice Amount) and compare the totals in the BI report against the sums in the browse. In the BI Report, Qty Ordered - Qty Shipped should equal the Qty Backordered in the Browse. **Note:** If you want to perform your validation on a range of items or a range of invoices, you can copy the query used by this report and modify it to search using your preferred criteria. You can then make a report from that query. However, if you include multiple items in the browse you cannot add a summary, and you would have to compare line-by-line. Using an invoice range may cause the selection drop-down lists for choice of invoice to be large and difficult to use.

OM Order History Fact Table

Report name	7.1.2 Sales Order Browse
Usage	Compares the OM Order History Fact table (fact_om_order_history, standard [non-scheduled] orders) to the 7.1.2 Sales Order Browse. Set the Due Date Range in the browse to cover the month or months chosen in the BI report range. Set the Item Number filter in the browse to the same item number selected in the BI report.
Validation	Add SUMs for the Quantity Order/Open to the browse and compare the totals in the BI report against the sums in the browse. **Note:** EE environments can use the Sales Order Detail Browse instead. If you want to perform your validation on a range of items, you can copy the query used by this report and modify it to search using your preferred criteria. You can then make a report from that query. However, if you include multiple items in the browse, you cannot add a summary, and you would have to compare line-by-line.

OM History Fact for Scheduled Orders Table

Report name	7.5.10 OM Scheduled Order History Validation Report
Usage	Compares OM History Fact for Scheduled Orders table (fact_om_order_history, scheduled orders) to 7.5.10 Schedule Report. In the Schedule Report, set the Create Date range to the dates that cover the month range chosen in the BI report. Set the Schedule Type to 3 and Active/History to Active.
Validation	Line-by-line comparison

Purchasing Validation Reports

These reports are located in the Reports menu, under the QAD Standard|Operations|Purchasing|Purchasing Validation folder.

They are also available as links from the Operations Administration dashboard.

Purchase Order History Fact Table

Report name	5.8 Purchase Order History Validation
Usage	Compares Purchase Order History Fact table (current records) to the Purchase Order Browse.
	In the source (ERP) browse, add a filter for Due Date range to the first day of a calendar month through the last of a calendar month. The range may span multiple months. Also add a filter for the same Item Number used in the BI validation report.
Validation	Quantity Open (BI) = Quantity Open (ERP)

Purchase Order Receipt Fact Table

Report name	5.13.5 Purchase Order Receipt Validation
Usage	Compares Purchase Order Receipt Fact table (fact_po_receipt table) to the 5.13.5 Purchase Receipt report. This report takes the Domain and a Start Month and End Month as parameters. You can set the start and end months to the same value to run the report for a single month. Set the ERP report's Receipt Date range to cover the same time frame as the BI report's Start and End months. Clear the "Unmatched Only" check box.
Validation	Base Standard Received Value total (BI) = Ext GL Cost total (ERP)
	Base Actual Received Value total (BI) = Ext PO Cost total (ERP)
	Note: Because of rounding differences, some lines may be off by a penny or so. Over the whole report, the difference may be several cents. If required, do a line-by-line comparison to ensure that the differences are due to rounding error.
	Base PO Price Variance total (BI) = PO-GL Var total (ERP)

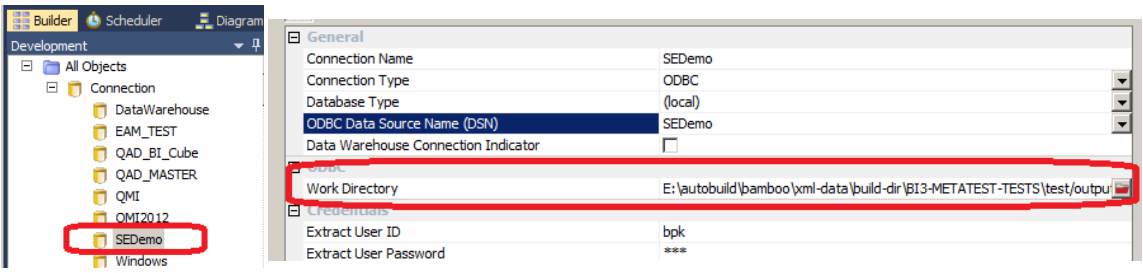
Automated Order Management Validation

You can schedule the job VALIDATION_OM to run at regular intervals and compare a spreadsheet generated from the ERP Invoice History Browse to the BI values. You can use this capability to validate a large quantity of data that would be difficult to compare visually.

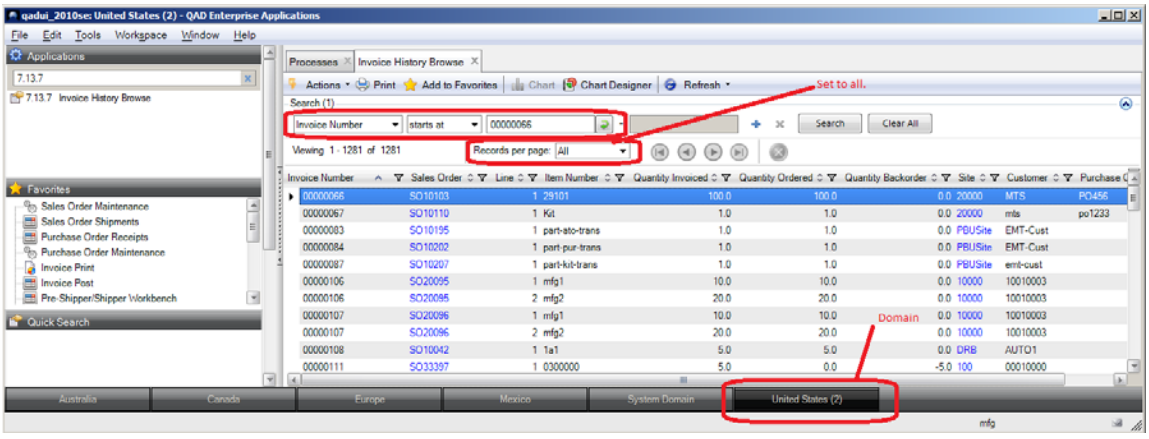
Running Validation of BI Against ERP Reports

Standard Edition – Invoice History

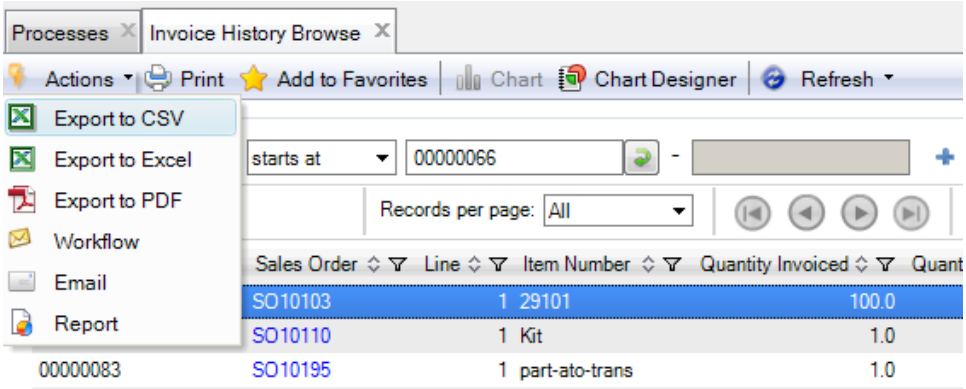
1 Before generating any reports, determine in which directory you will place the output. This could be different for each source if you are on a multi-source system and have defined different work directories for each Connection. You can find these work directories by opening the Connection in question in the DWD and looking at the Work Directory. In the example below, the SEDemo connection was opened to find the work directory.



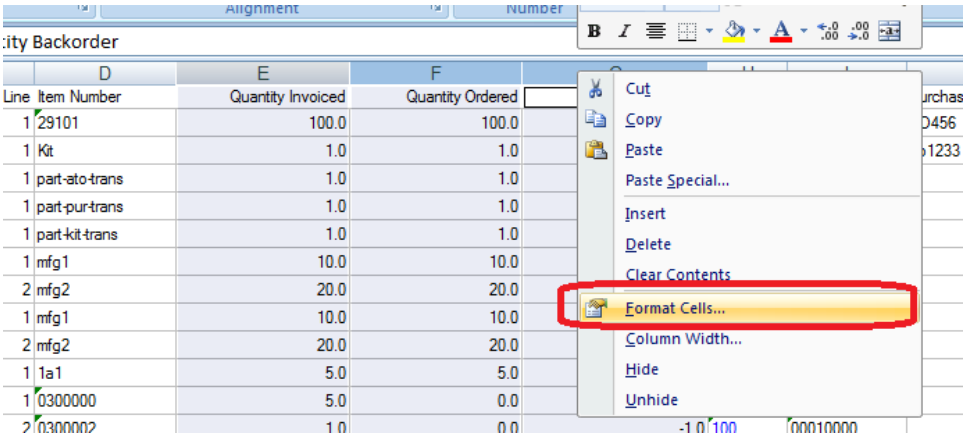
2 In the ERP side, open the Invoice History Browse (also known as 7.13.7). Pick which Domain this query will be run against, then pick a starting Invoice number and set the results sets to run to All. Run the query for all invoices greater than that Invoice number.



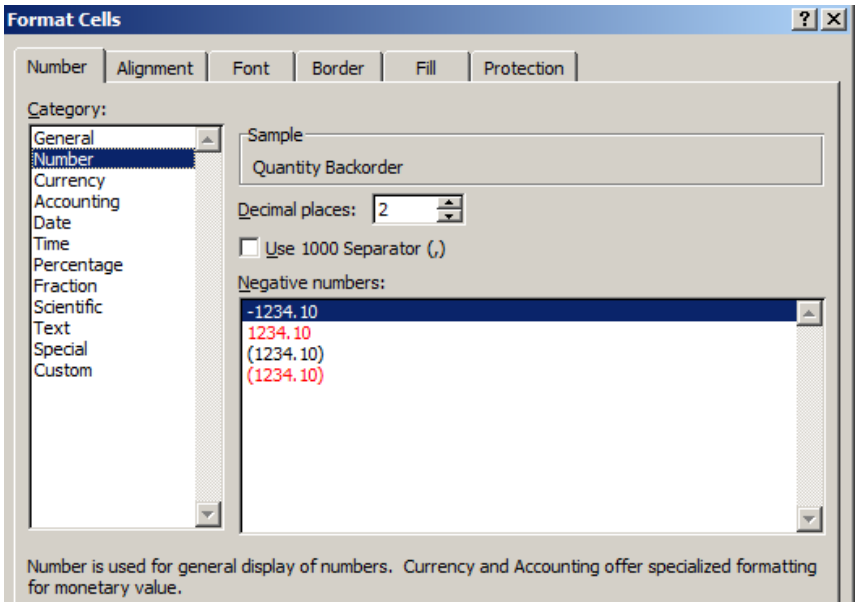
3 Under Actions, run the browse to output the results to Excel (exporting to `.csv` has an issue with commas in numeric fields that makes it difficult to properly load those records).



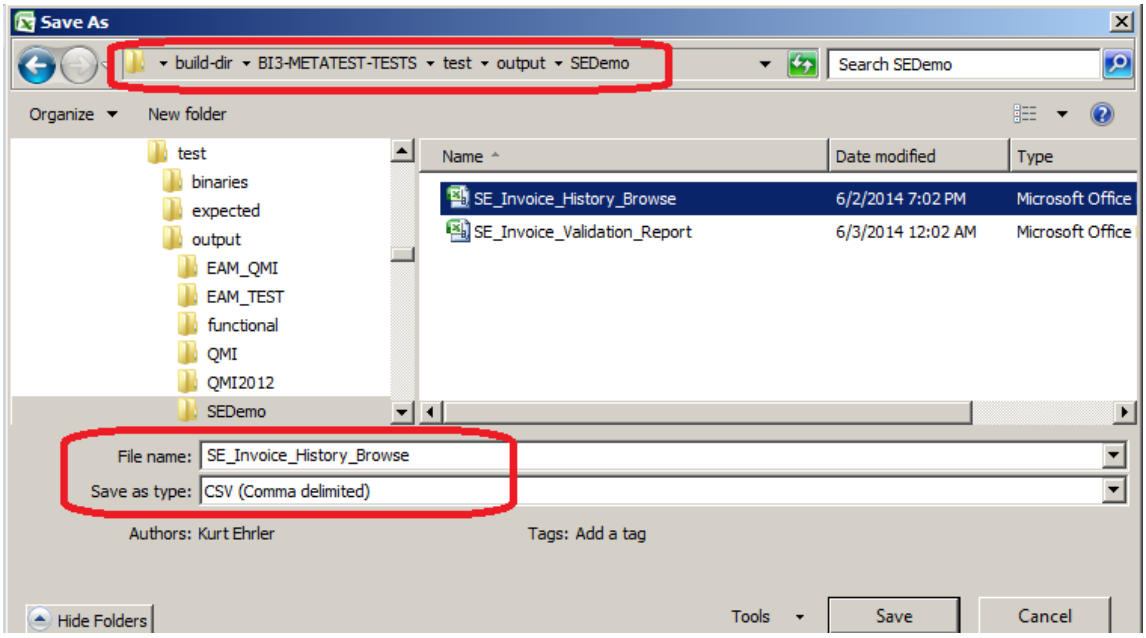
4 When the Excel worksheet opens, reformat the numeric columns by highlighting the tops of the columns and choosing Format Cells. In this case, the columns are E, F, and G.



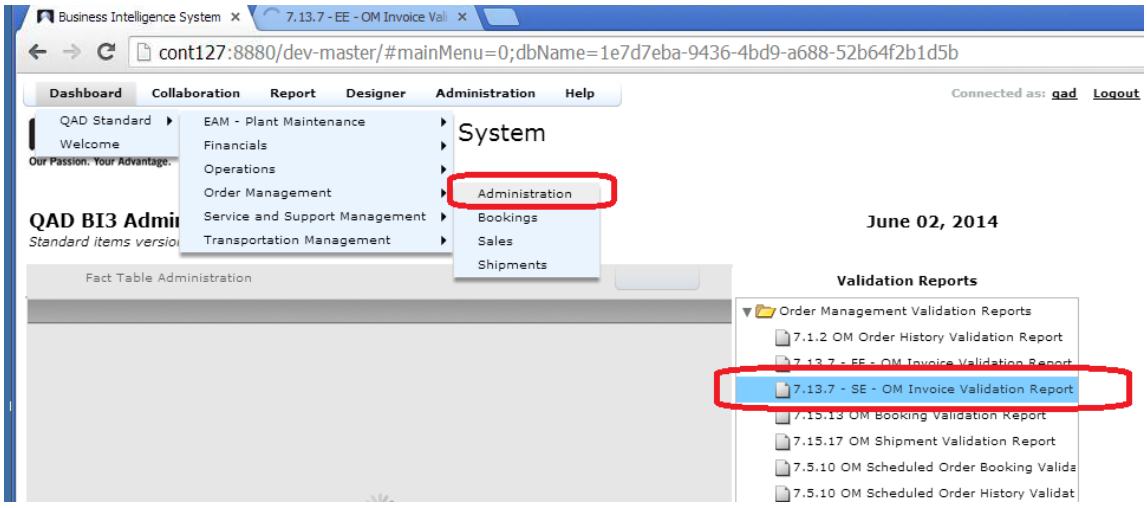
5 Here, set the format to Number and make sure that the 1000 separator box is not checked. The top option with the negative sign would be the right choice and the number of decimal places can be set as is appropriate. In this case the number is being left as 2.



6 Place the resulting file in the data source connection's work directory (as defined in the data source's DWD connection properties). Name the file `SE_Invoice_History_Browse.csv`. Be sure to close the Excel window once you have saved the `.csv` file.

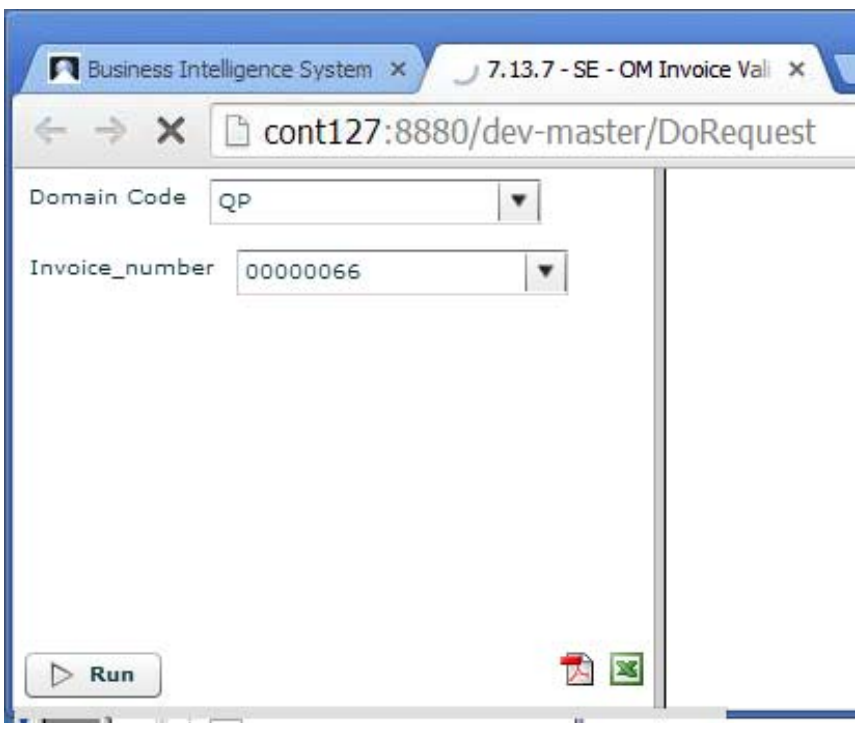


7 Now go to the Portal and go to the Dashboard, QAD Standard, Order Management, Administration and pick the SE – OM Invoice Validation Report.

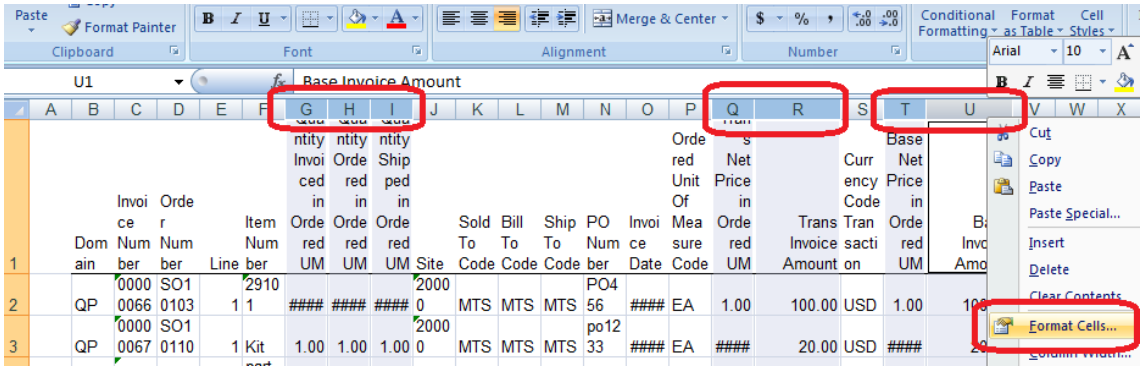


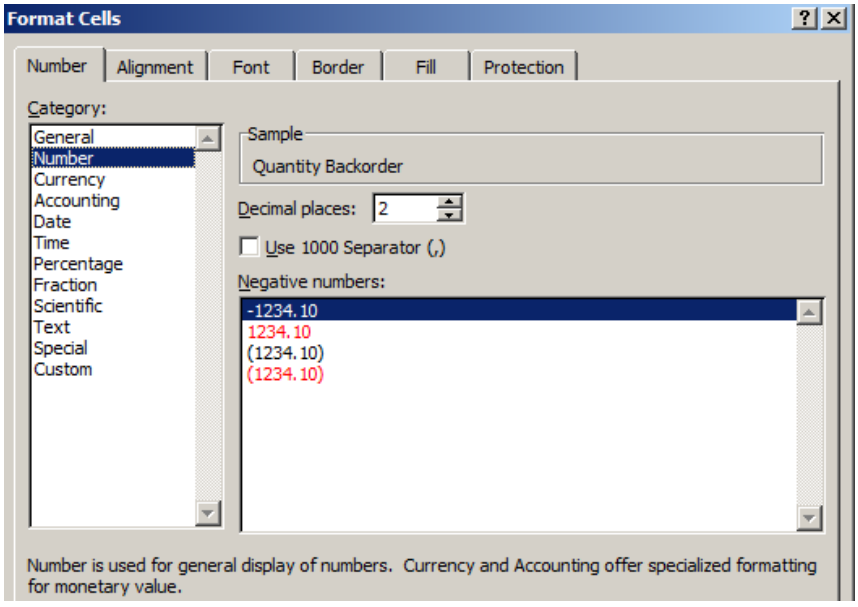
8 Run the report using the same parameters that were used to run the Browse in the ERP. Choose the Excel option to generate output to Excel. You may need to turn off your pop-up blocker. A separate window opens and an `export.xsl` file is saved to your User/Downloads directory.

Note Sometimes it is not obvious that anything is happening because the download window is in the background. Check to see if a second browser window exists if you cannot find your download.

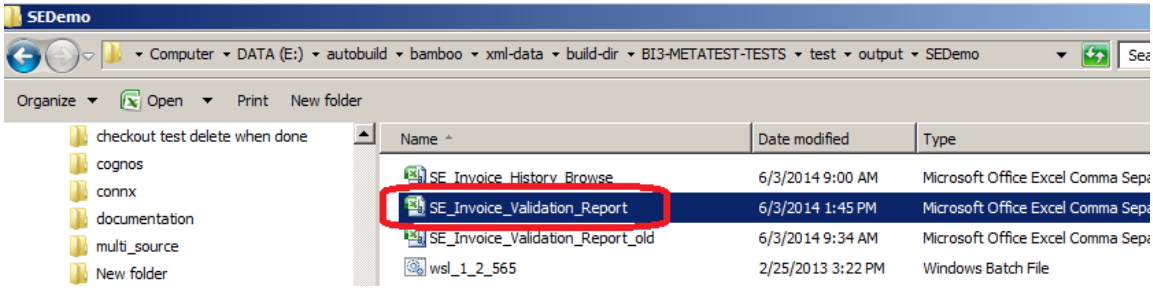


9 Open the `export.xls` file (or whichever version is the one you need) and change the formatting on columns G, H, I, Q, R, T and U to not have a comma separator at the thousands mark.

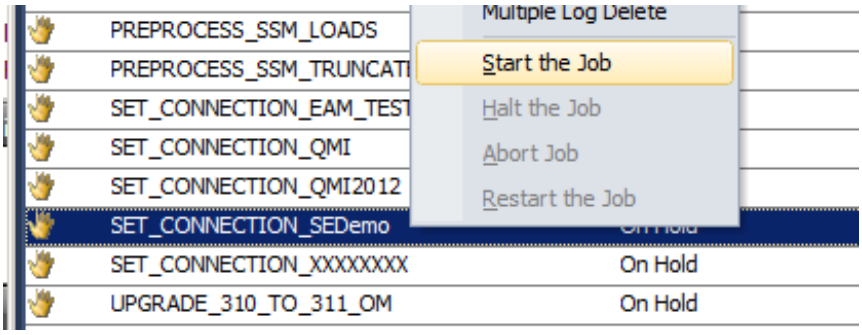




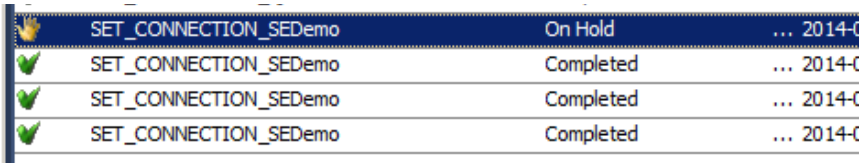
10 Save the output to the same Work Directory the browse was saved to. Save it as `SE_Invoice_Validation_Report.csv`.



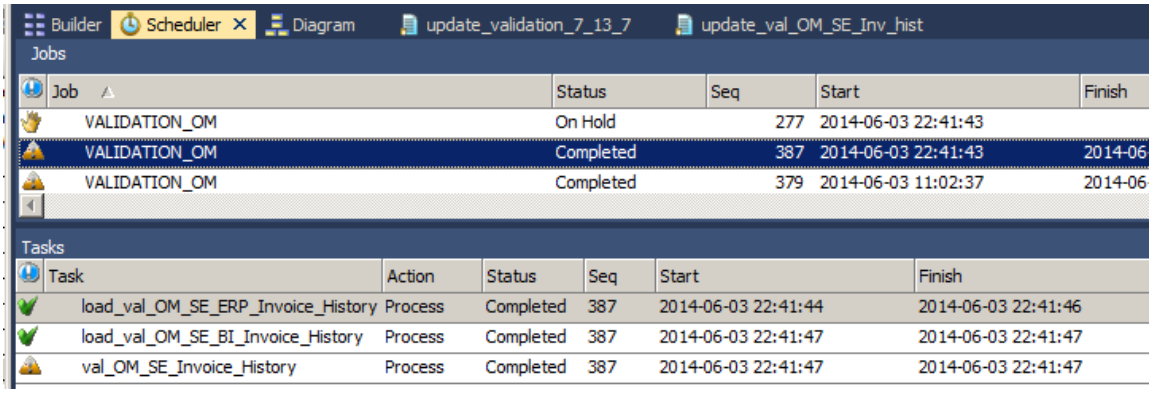
11 In the DWD, go to the scheduler. Make sure that no other jobs are about to run. Run the SET_CONNECTION job for the source the reports were just run on.



12 Confirm that the SET_CONNECTION job has completed.



13 Now run the validation job that will compare the results of the ERP browse against the portal results.



Appendix A

Sizing Guidelines

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

Introduction 164

Business Intelligence Deployment Options and Terminology 164

CPU Sizing 168

Disk and IO Subsystem 169

Network Considerations 171

Example Sizing 172

Reference: Estimation of Initial Database Size 173

Reference: Network Sizing Details 176

Introduction

This section 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 section are provisional in the sense that they were produced in the absence of a formal capacity planning process.
- **Interim:** This sizing section 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 section are the most accurate available to QAD. Use this document for practically sizing BI deployments.
- **Caveats to BI Sizing:** The sizing estimates given here are the best available. However, the absence of a formal capacity planning process means that certain estimates in this document may need updating as more accurate information becomes available. Carry out the sizing 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.

These estimates are the most accurate available at the time of publication. If you take the following restrictions and caveats into account, the estimates provide guidance for QAD BI sizing:

- This product has not been sized using formal internal sizing/scalability/performance testing. 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.
- 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, remember that since sizing is taking place based on incomplete information, some estimates may not be 100% accurate.

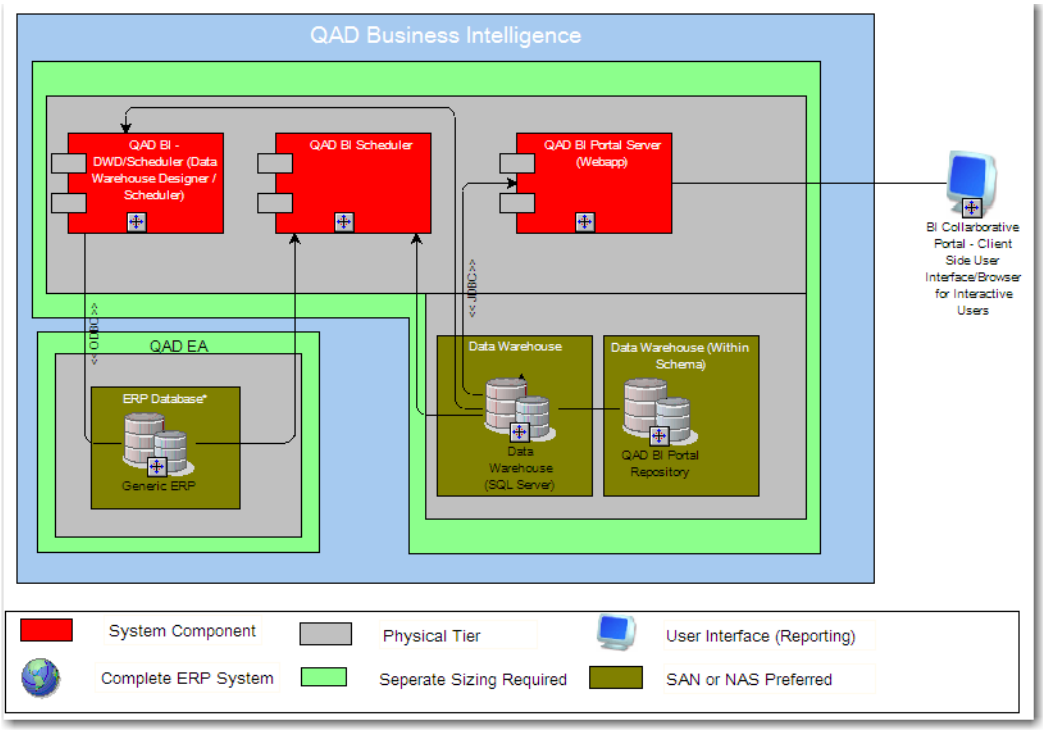
In summary, consider the current sizing estimates as best-estimate sizing guidelines, safe to use in the field, but subject to future updates/correction and also subject to restrictions and caveats.

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 DWD component 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 section, DWD/Scheduler refers to all of the technical components composing the DWD and the DWD Scheduler.
- **QAD BI Collaborative Portal (Client-Side Browser):** This component 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 component 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 the primary server machine.

BI Deployment Options

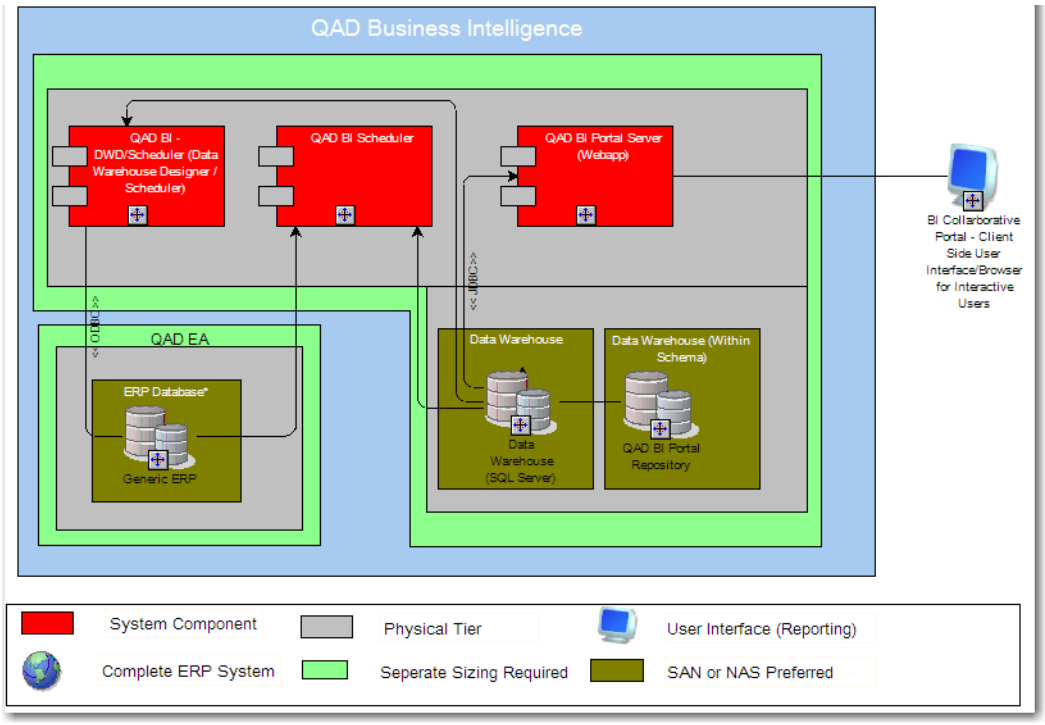
Business Intelligence involves several different components and processes. Before determining a final sizing estimate, size the components and processes separately and appropriately, depending on the deployment option. There are two primary deployment options available:

- **Single Tier:** All BI components are 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 a single- or two-tier deployment.

Deployment Option 1 – Single Tier

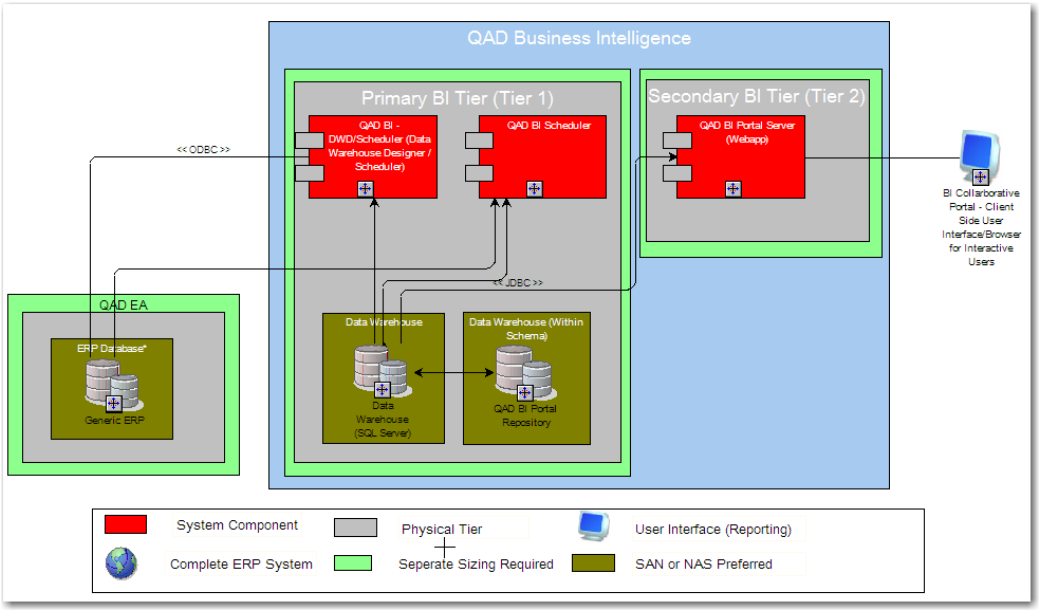
A single-tier BI deployment has all BI components on the same server. See “BI Terminology” on page 165 and the schematic here:



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. Therefore, the size of a BI Portal server is smaller than the BI DWD/Scheduler server. This distinction is made throughout this section.



Sizing Terminology Used Throughout this Sizing Section

Sizing Metrics

Sizing across the following 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 these considerations, size each workload type separately. The methodology used for sizing BI divides each metric into the following areas:

- **Sizing for DWD/Scheduler Workload (Batch Jobs):** This term refers to the workload placed on the server as a result of the DWD/Scheduler automated jobs. Size this workload, which is of a batch-type, separately.
- **Sizing for Concurrent User Workload (Interactive Portal Users):** This term 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. Size this workload separately.

- **Sizing for BI Portal WebApp (Server Side).** You can place the BI Portal WebApp 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

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 *faster* CPU cores (not from *more* CPU cores).

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.

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- 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 for DWD/Scheduler Component

The following factors determine the amount of RAM needed:

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

Sizing Metric	RAM for OS (GB)	RAM for SQL-Server	Total Server RAM[1]
Small Load	4	8	12
Medium Load	4	10	14
Large Load	4	16	20

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.

CPU Sizing for BI Portal WebApp (Server Side)

This type of sizing involves CPU sizing for just 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 1	Recommendation 2
CPU – BI Portal Component (Java WebApp)	1 CPU Core	

Sizing Metric	Recommendation	Comment
RAM – BI Portal Technical Component	2 GB	Allocate 2GB to the BI Portal WebApp. On 64-bit systems and for 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 was 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 consideration was 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. 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 173 for details. Follow these guidelines. **Recommendations** 1. Minimum - A minimum of 500-GB disk space is recommended, but see points 5 and 6. 2. Ideal - Ideally 1 TB of disk space is recommended, but see points 5 and 6. **General Guidelines** 3. Allowance for database growth - 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 176 for the exact details. 4. Allowance for disk expansion – If appropriate and to allow for future project growth, it is recommended that you allow for disk expansion capabilities in the SAN/NAS device. 5. Estimation - “Reference: Estimation of Initial Database Size” on page 173 provides an approximate method of estimating the initial BI database size from source ERP tables. You can use this method 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 size as appropriate.
SAN/NAS?	It is preferable to use external disk arrays or SAN/NAS devices where possible. Configure internal disk drives 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:** Due to the larger IO demands of BI systems, a *minimum* of six disks is recommended. Double the number of disks when using mirroring. **Note 2:** More disks are preferred, because using more disks significantly helps performance and IO rates.
IO Controller	Ensure that 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. Base the final estimate of disk space on best-estimates. See points 2 and 3 below.
2. Use “Reference: Estimation of Initial Database Size” on page 173 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 176 for details.
4. Fast disks – 15k SAS disks.
5. RAID Level spread across multiple disks, recommend six disks minimum.
6. The associated IO controllers must be capable of servicing maximum throughput from each disk or combination of disks in RAID configuration. To prevent a bottleneck at high IO demand rates, ensure that the IO controllers used do not limit transfer rates per disk below the maximum rated physical-disk and RAID transfer rates.

Notes on Database Disk Sizing

Factors Affecting Disk Sizing Estimates

There is currently no detailed information on how to scale the expected size of the BI data warehouse from a source ERP system. The reason for this lack of information is that the size of the BI data warehouse depends on various 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 necessary, refer to “Reference: Estimation of Initial Database Size” on page 173 for an approximate method of estimating the initial BI database size. This estimate is approximate. Only use it to get an order-of-magnitude estimate of 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. Make an allowance 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 three years of historical data and you assume that the same rate of transactions is 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 three years.

Also, daily or weekly monitoring of the BI database size provides ongoing measurement of growth rates, so provide adequate storage 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 176:

- 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 176 for details about estimating the impact of network considerations in each of these areas.

Network Recommendations

Take into account the sizing notes from this table and review “Reference: Network Sizing Details” on page 176 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 178 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 that these rates are *minimum bandwidth per user* 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. Make very careful measurement 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 176 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	Two cores	
B	CPU – Concurrent User Workload (Remote Portal Users)	20 concurrent users = 6 cores	
C	CPU – BI Portal WebApp	One core	
D	CPU Total	6 + 1 = 7 cores	Max. (A,B) + C – This is a *minimum requirement*
E	RAM - DWD/Scheduler	Assume “Medium” Load – 10+ 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 169)	
H	RAM - Total	10+ + 2 + 4 GB = 16 GB	As per “RAM Sizing for DWD/Scheduler Component” on page 168, the 4 GB extra is for the OS
I	Disk Sizing	Carry out initial approximate sizing to create an *approximate* estimate of the initial DW size based on ERP table data. See “Reference: Estimation of Initial Database Size”.	500 GB is a recommended minimum. However, obtain *disk sizing estimates* for all new implementations by using the referenced guidelines.
J	IO Subsystem	15 k rpm SAS disks RAID 10 on a SAN/NAS device Minimum of six 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 connection between the DW server and the BI Portal WebApp server.

Reference: Estimation of Initial Database Size

If the BI Data Warehouse does not exist, use the following information to obtain an approximate initial expected BI database size.

Dependence of Initial Database Size on BI Modules

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 depends on the modules and submodules used. This section describes how to estimate 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 to estimate the initial data warehouse size.

Note There is no guarantee that this estimate is an accurate initial data warehouse size. It is provided as a way to generate an order-of-magnitude estimate on the initial data warehouse size.

Note SQL-Server uses a large amount of extra disk space for the temporary manipulation of transformational data. You must allow for this. The following estimate only considers an approximation for the BI database, excluding requirements for SQL-Server temporary files.

Determine the size of the following tables in the source QAD ERP database and use that information with the Size Factor to estimate 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*		

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

Module	Table	Source	Size Factor**	Source ERP Table Size	(Size Factor) * (Source Size)
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		
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**		
	xmtpasp0		1/10*** x 1/3**		
	xmtpesp0		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

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 the relevant BI module does not use all transactions in the specified table. Take this into account in size estimations. For example, in the BI Financials Module, the ar_mstr table has a size factor of 2/3. This factor indicates 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 many years (*n* years, say *n*=10 years). For example, if the data warehouse will be populated with only three years worth of data (*y*=3), apply an additional size factor of *y/n* (in this case 3/10) to the relevant source ERP tables. Include this extra size factor in the table above and in the calculations to determine the estimated initial size of the data warehouse.

***** TABLE NOTE 3:** The 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 three years, expect the data warehouse size growth over the coming three 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 three years of activity, and the initial size is 250 GB, then expect the data warehouse size to at least double over the coming three 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 parameter refers to the data transfer rate, measured in bits per second (bps), kilobits 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 term is a measure of the time delay when transferring data from one point on a network to another point on the network. It is usually measured 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 arrangement is in contrast to a LAN (Local Area Network) where 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.

Importance of Latency and Bandwidth

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

- A large volume of data may be transferred between the ERP system and the BI server. The volume could be hundreds of MBs or multiple GBs, depending on database size, transaction volumes, and so on. Therefore, 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. Bandwidth and latency can limit the transfer rate of large data transfers. For this reason, sizing and allocating appropriate bandwidth is vitally important.

Bandwidth Requirements

Figures quoted here the peak bandwidth requirements, but note that this rate 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 MB/(3 MB/s) = 333 seconds = 5 min. 33 sec.

WAN, 200 ms: 1000 MB /(0.4 MB/s) = 2,500 sec. = 41 min. 36 sec.

In reality, the peak rates are sustained only briefly and average transfer rates are much lower—some average transfer rates may be 10 times lower. These figures are only provided 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. Processed data is then 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 hundreds 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

Reserve 100 kB/s bandwidth minimum per user, bearing in mind the following caveats:

- **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.
- **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 results in a decreased network transfer rate, extending communication time, and decreasing bandwidth requirements, at the cost of overall responsiveness.

Appendix B

Customization Guidelines

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

Overview 180

Customization Challenges 180

QAD BI 3.x Customization Methodologies 183

Customization Needs for Additional Columns 185

Identifying Every Appearance of a Column in Existing Code 192

Naming Conventions for Custom Changes to Packaged QAD BI Modules 192

Guidelines for Creating Custom Path Custom Tables 193

Adding Additional Columns to a Load Table 195

Adding Additional Custom Columns to Pre-perm Stage Tables 197

Adding Additional Custom Columns to perm/dim/fact Tables 199

Adding Custom Path Stage/Extract Tables After the Perm Tables 201

Additional Customization Topics 206

Overview

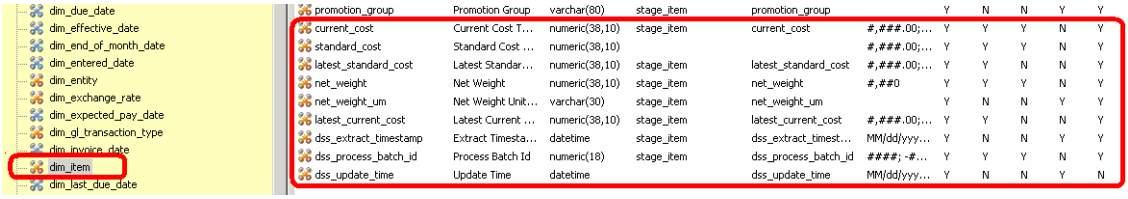
Most customers that purchase the QAD BI modules invariably want additional information that falls outside of the baseline code of the warehouses. This comprehensive guide to QAD-recommended best practices for BI 3.x customizations helps you get the additional information you need. The QAD BI Data Warehouse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

Customization Challenges

The DWD tool makes the addition of more columns or tables to a data warehouse fairly straightforward. This feature is one of the primary selling points of its creator, WhereScape. One of the great challenges in data warehousing is that once a data warehouse is built, it is difficult to modify. WhereScape's approach eliminates much of the difficulty with getting these changes right. However, the original intent of the tool is for a single customer to be able to modify their data warehouse. QAD is building out its BI solution to package warehouses into different modules that are implemented for many different customers. Therefore, it is a much more challenging environment for releasing bug patches and upgrading code. Each customer may have a different set of custom work. Installing patches or upgrades to tables and procedures with custom work can wipe out the custom 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.



promotion_group	Promotion Group	varchar(80)	stage_item	promotion_group		Y	N	N	Y	Y
current_cost	Current Cost T...	numeric(38,10)	stage_item	current_cost	#,###.00;...	Y	Y	Y	N	Y
standard_cost	Standard Cost ...	numeric(38,10)			#,###.00;...	Y	Y	Y	N	Y
latest_standard_cost	Latest Standar...	numeric(38,10)	stage_item	latest_standard_cost	#,###.00;...	Y	Y	N	N	Y
net_weight	Net Weight	numeric(38,10)	stage_item	net_weight	#,##0	Y	Y	Y	N	Y
net_weight_um	Net Weight Unit...	varchar(30)	stage_item	net_weight_um		Y	N	N	Y	Y
latest_current_cost	Latest Current ...	numeric(38,10)	stage_item	latest_current_cost	#,###.00;...	Y	Y	N	N	Y
dss_extract_timestamp	Extract Timesta...	datetime	stage_item	dss_extract_timest...	MM/dd/yyy...	Y	N	N	Y	Y
dss_process_batch_id	Process Batch Id	numeric(18)	stage_item	dss_process_batch_id	####; -#...	Y	Y	Y	N	Y
dss_update_time	Update Time	datetime		dss_update_time	MM/dd/yyy...	Y	N	N	Y	N

Here is the dim_item table layout for a customer with a 3.3 customized version of the modules that has the easily identifiable custom columns that begin with an x_ prefix. QAD's preferred prefix for custom columns now is xx_. This layout also has some other columns that do not match the 3.4.1 dim_item table. These columns were 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)			#,##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_descrip...	Promotion Grou...	varchar(80)	stage_item	promotion_group_d...	
dim_entity	supplier_description	Supplier Descrip...	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		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                          current_cost
, 0                                     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
, x_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 ))
```

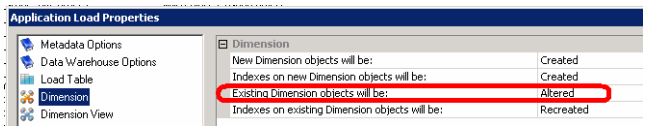
A patch is then applied using the Administrator tool to fix some issue with the table or to upgrade the table to the 3.4.1 version of the code. You can see that it initially says that existing objects are replaced.

Object	Status
Application load begins	D:\kurt_work\custom_tests\app_obj_Patch_Test_1.wst
get_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_om_order3	Will replace existing object
update_stage_om_order2	Will replace existing object
update_stage_om_order1	Will replace existing object
custom_stage_om_order1	Will replace existing object
update_fact_om_order_history	Will replace existing object
custom_fact_om_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_om_order3	Will replace existing object
stage_om_order2	Will replace existing object
stage_om_order1	Will replace existing object
fact_om_order_history	Will r
dim_item_idx_0	Will r
dim_item_idx_A	Will r
stage_om_order1_idx_A	Will r
fact_om_order_history_idx_1	Will r
fact_om_order_history_idx_10	Will r

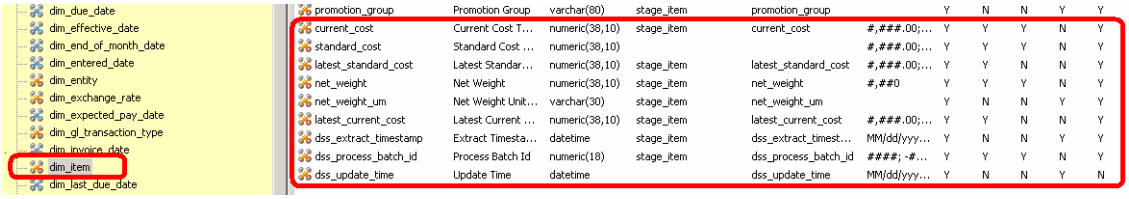
Setup Administrator

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

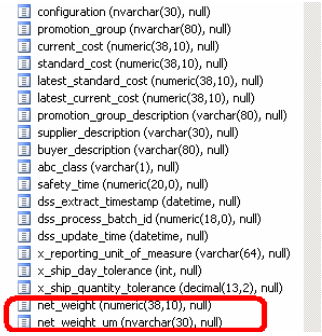
However, the default setting for each object during application installation ensures that the changes alter the dim, perm, and fact tables and the updates proceed successfully.



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.



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. 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.



Reviewing the affiliated procedure in the DWD shows that all of the logic referencing the custom columns was removed. Also, the new logic only reflects the columns that exist in the 3.4.1 version that was applied.

```
, configuration                          configuration
, promotion_group                        promotion_group
, current_cost                           current_cost
, 0                                      standard_cost
, latest_standard_cost                   latest_standard_cost
, net_weight                             net_weight
, net_weight_um                          net_weight_um
, latest_current_cost                    latest_current_cost
, dss_extract_timestamp                  dss_extract_timestamp
, dss_process_batch_id                   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 Because the new columns were added to the table and the table was not dropped, the custom columns still contain their data. However, the new procedure no longer updates the new columns.

As you can see, implementing patches and upgrades to tables that have any custom work done to them is not a straightforward process. It likely requires substantial follow-up work to get the customizations back in place in the DWD meta data. The underlying SQL table can remain unchanged. Ideally, it is for this reason that custom work is not directly done on standard release tables in the data warehouse. This reason is also why, at the least, the footprint of the customized work mostly falls outside of the baseline code.

QAD BI 3.x Customization Methodologies

Adding custom columns to our existing standard infrastructure presents potential problems. Whenever possible, you must perform customizations in a way that allows for easy patch and upgrade installation.

Custom Path Methodology (Preferred)

To avoid the pitfalls of changing our existing baseline tables at customer sites, it is much simpler if the custom requirements are met without ever touching the existing infrastructure. This is nearly a possibility! You can resolve this issue by creating a custom path for the customization that has its own separate objects. This custom path 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 are 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, add the custom column back into the DWD, but not the actual database. If cubes are used and the new customization desires them, add the additional customization as 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

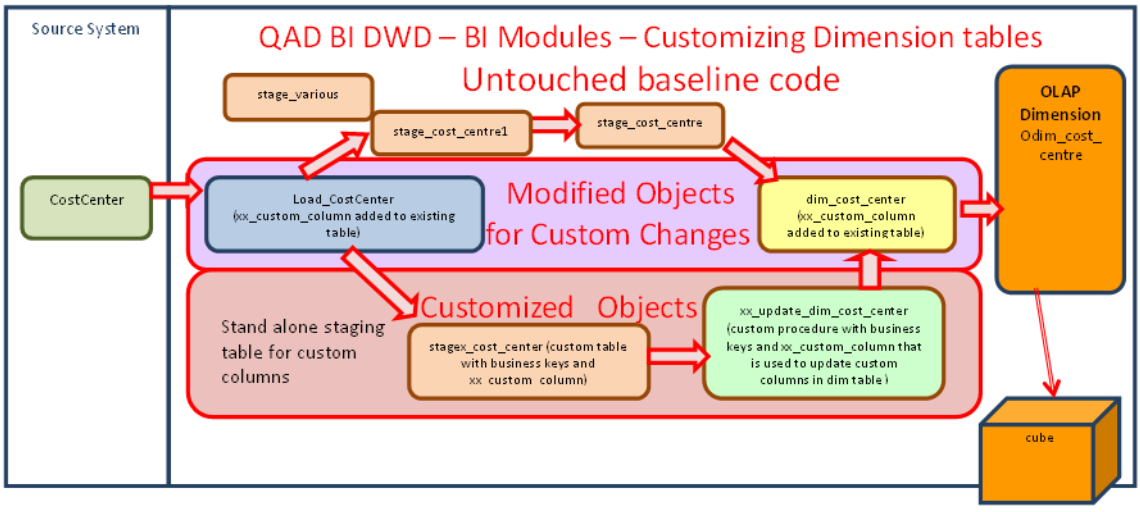


Fig. B.2
DWD Processing Architecture with Customized Objects and Update Procedure for a Fact (not History or Rollup)

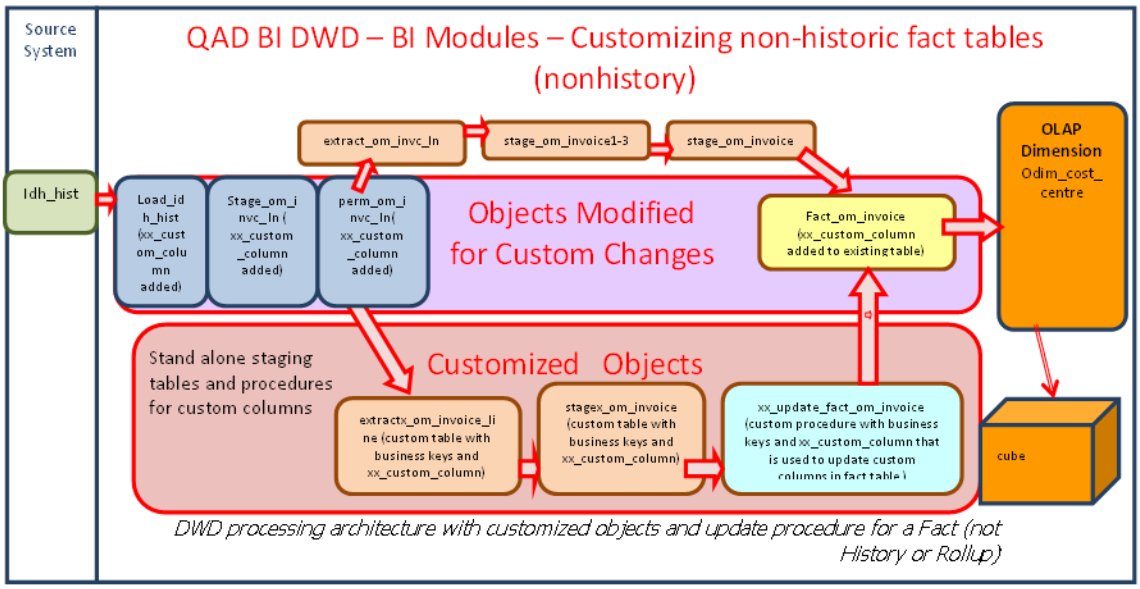
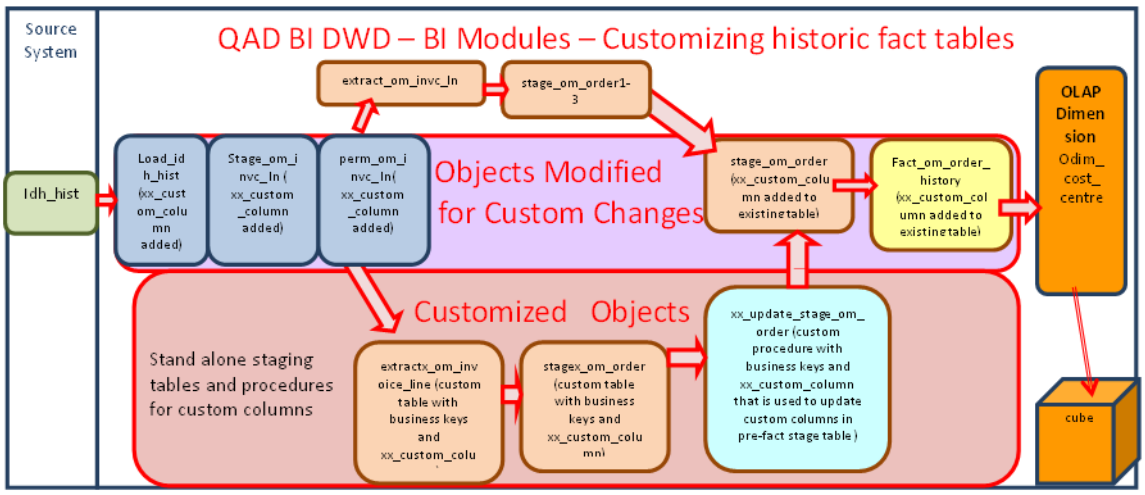


Fig. B.3
DWD Processing Architecture with Customized Objects and Update Procedure for a Fact (History)



Pros and Cons

Pros

- Any existing reports would continue to work because 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 low risk of data loss with this integration method because the core dimension and fact tables only have extra columns added to them by the customizations that the stored procedure populates. If there is a patch or upgrade, restore the columns in the Designer, and re-add the custom column update procedure to any jobs it may have been lost from.

Cons

- This method of customization requires you to add custom columns directly to the dim/fact table. After an application upgrade or patch, reapply custom columns need 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 their normal insert/update processes populate.
- 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 make a parallel table path too difficult to set up. This scenario is true for some fact tables, in particular those fact tables that do a large amount of ranking, like the fact_om_order_history table. If an attempt was made to create a parallel path and it is too difficult to achieve a satisfactory result, adding custom columns to the existing infrastructure is allowed. Future patches and upgrade installations have to take these into account and they will have to be reimplemented when they get disrupted. Inform the customer of this possibility as it requires an additional services engagement to re-implement the customizations.

Pros and Cons

Pros

- Easier to do an initial implementation.
- Ensures that 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 must be reimplemented after the patch is applied. This consideration 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 should be careful and aware of Cartesian joins.

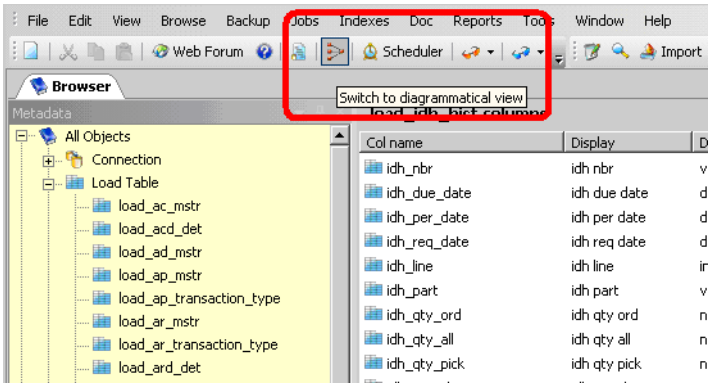
Customization Needs for Additional Columns

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, check if the data is already being pulled into one of the pre-staging tables and if the source table is already being referenced.

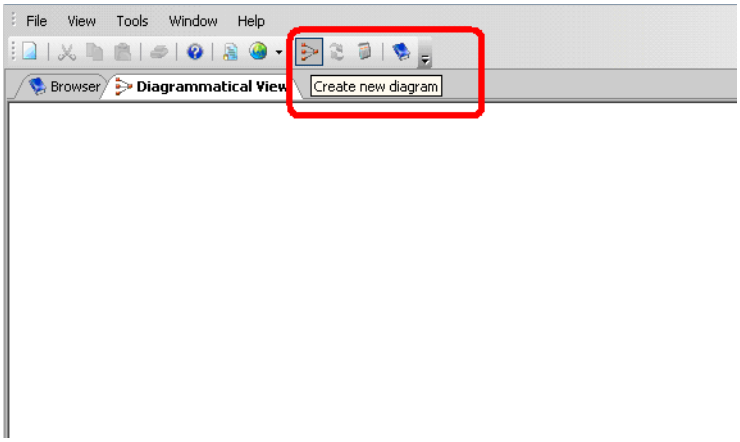
Use Track Back Diagram to Check Tables

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).

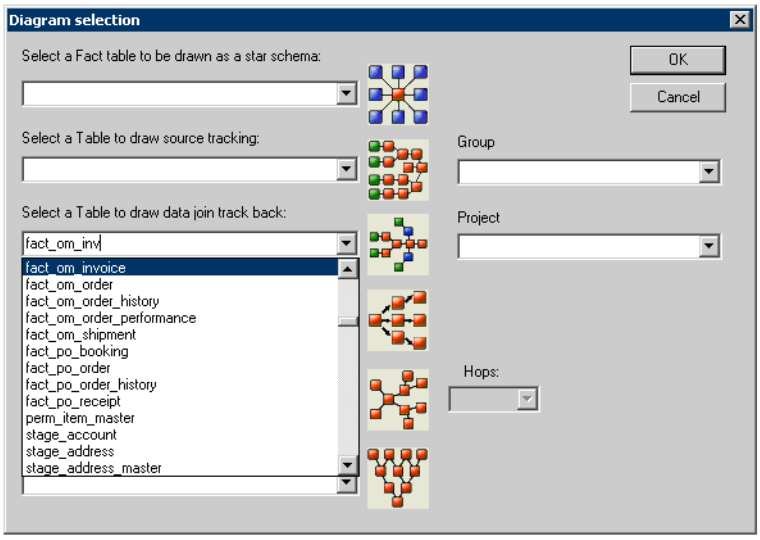


3 If the Diagrammatical View tab was not already open, click the icon a second time.



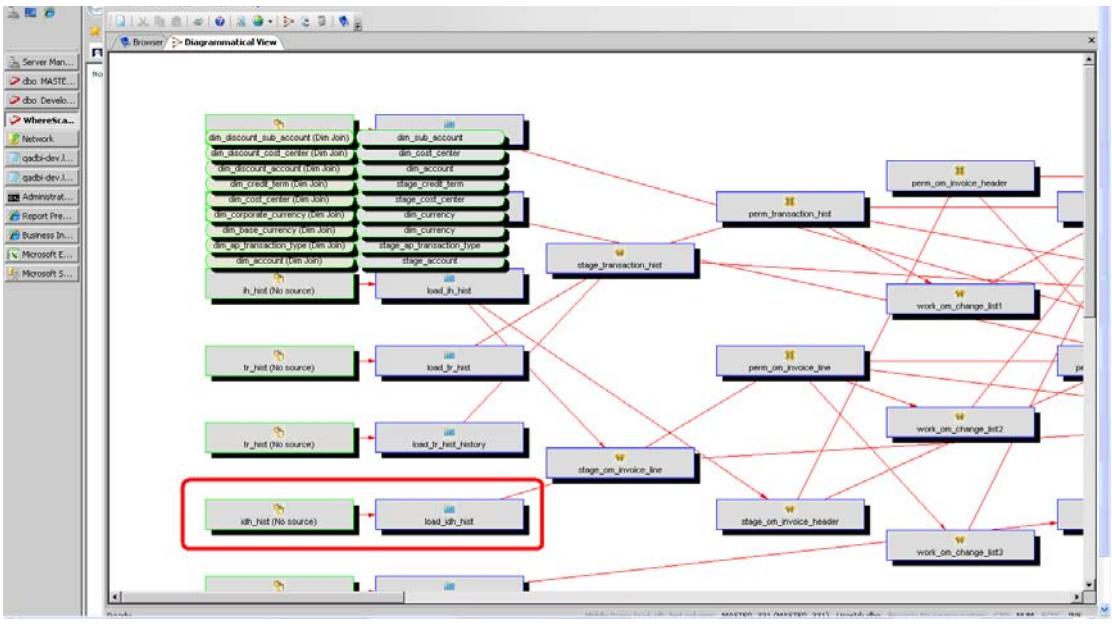
4 Pick the fact table you want to track back on and click OK. You can start typing the name of the table to find it in the drop-down selection list. However, if you type the whole name, you must still select the item from the drop-down list. Otherwise, nothing appears when you click OK.

5 For 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.

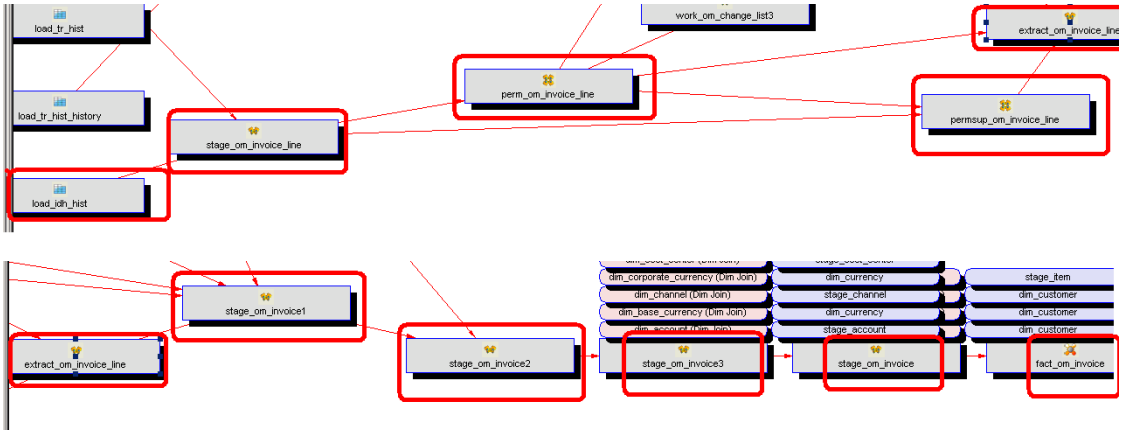


6 A track back diagram appears displaying all the tables that feed the fact_ table.

7 Identify here if the fact_ table already references the source table of the new data.



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 is 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 previous dataflow diagrams are for determining new data based on dates. Therefore, they were not included in the list for modification.
- extract_om_invoice_line
- stage_om_invoice1
- stage_om_invoice2
- stage_om_invoice3
- stage_om_invoice
- fact_om_invoice

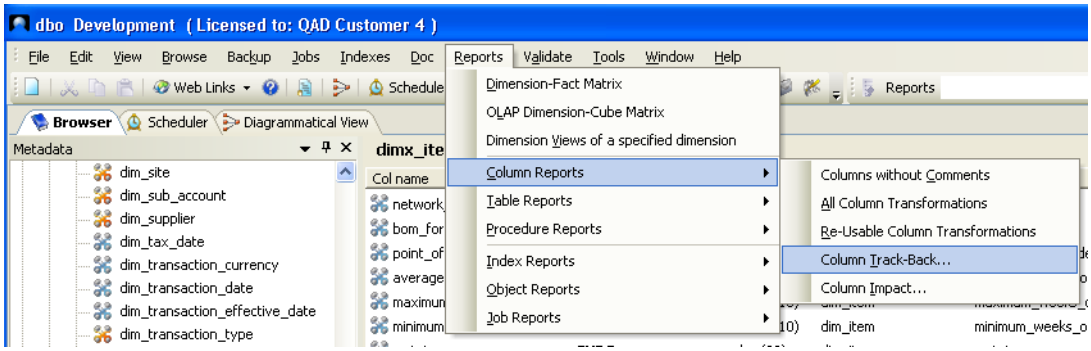
As a matter of practice, it is helpful to type this list into notepad or some other text editor for reference as you work through the customization.

Double Checking the Data Path with a Column Track Back Report

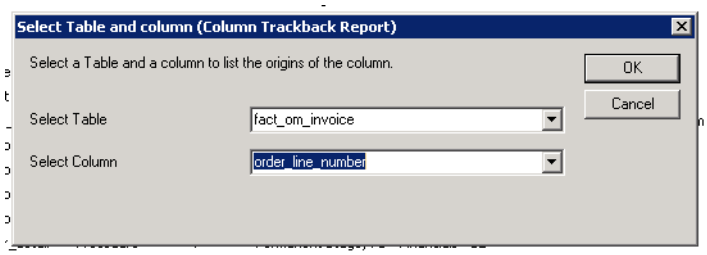
To be extra sure your table list is complete, you can do a track back on a column name from the fact table. This check is possible if you know that the column is from the same source table as the one that the custom column would be added from.

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

Note Before 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.

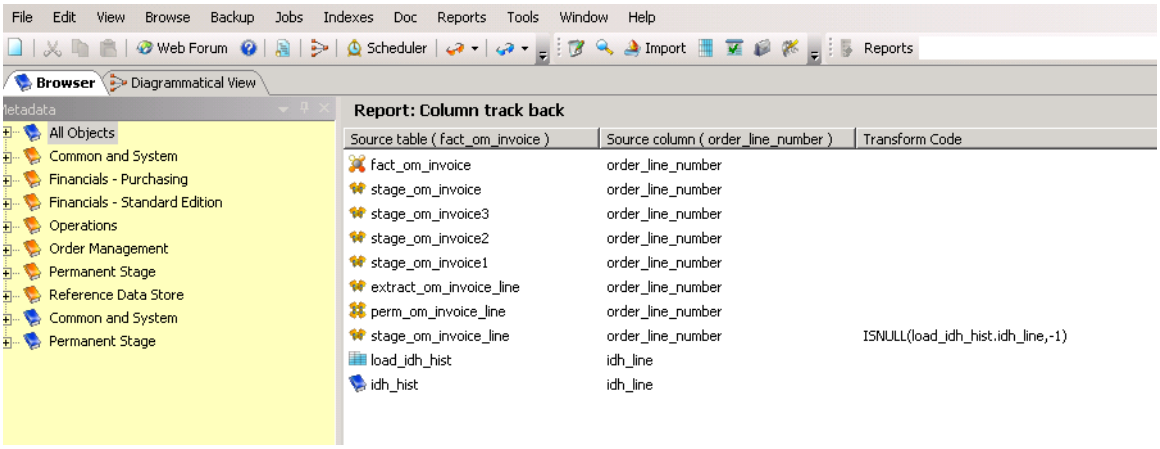


2 You are prompted with a box. Select the name of the fact_ table and a column you know comes from the source table in question. In this instance, we know that order_line_number comes from the idh_mstr table.



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 before the staging table that the dim_ value keys is defined in.

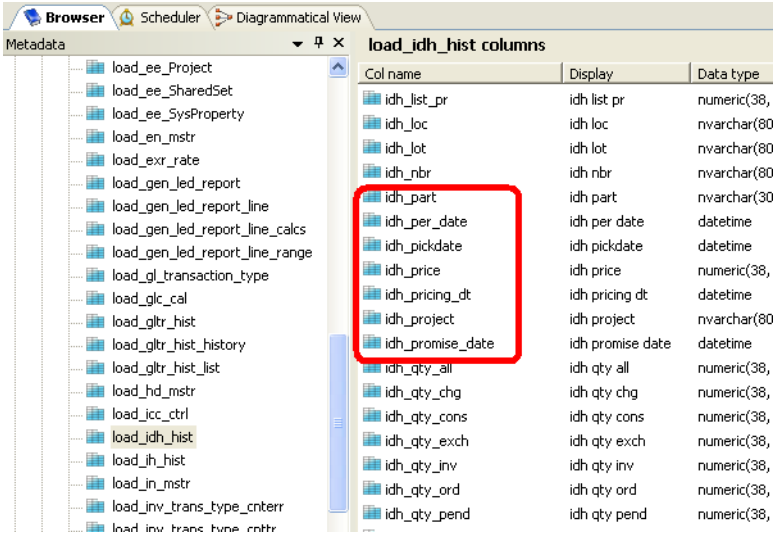


4 Aside from showing you all the tables that lead to the fact table, you also see the column names in those tables and any transform code that occurs for a column. This table-list closely matches the list we wrote down from the Diagramatic View.

5 We have identified that the QAD table the desired extra data is sourced from as existing in the dataflow as a load_ table. Now, 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 the Col name. In this case, idh_prodline is not included in the load_idh_hist table.



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 the table name, look for the source column to add in the Src column list.

Note If the list is long and it is not immediately obvious if the column is there or not, click the header to sort alphabetically. 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. Most of the BI extracts come directly from the QAD ERP system's Progress databases. There are only two changes or transformations that should occur for these columns. The first is the cleanup of blank spaces that were inadvertently left in the Progress table. The other is the removal of the "|" symbol from the data because it disrupts the data load.

Note There are some exceptions to this rule.

The following is what the transformation looks like:

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

Use this logic for any source table character (char,varchar) type column that gets added to the load table.

stage_ tables

Stage tables are used to do various things that transform the data in some way. The stage tables that come right after load tables are typically used to 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, "perm" tables (also called Data Stores) hold data that is permanently maintained in the source system. This arrangement prevents heavy system loading that would occur if such data were extracted during each run. There is a one stage table before the perm where the column names change from the load table; no transformations should occur before 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 arrangement might mean that the work table is looking for any records that were updated in the perm tables in the last ten days. For a historic load, all the records are pulled from the perm table. The determination of the date values to use is derived from parameters such as PO_HISTORY_DATE and PO_PROCESS_DAYS. PO_HISTORY_DATE is the day to start looking back from. PO_PROCESS_DAYS is the number of days to look 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 for pickup 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 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 creates keys in the same sequence if it repopulates. 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.

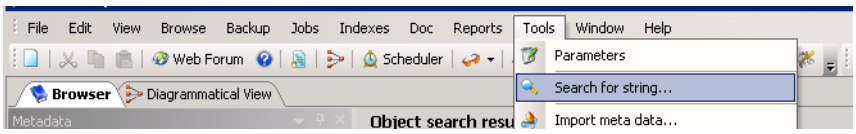
fact_ tables

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 important that fact_ table data not be inadvertently deleted or the table re-created. However, recovering from accidentally doing so is not as painful as it is for perm_ or dim_ tables.

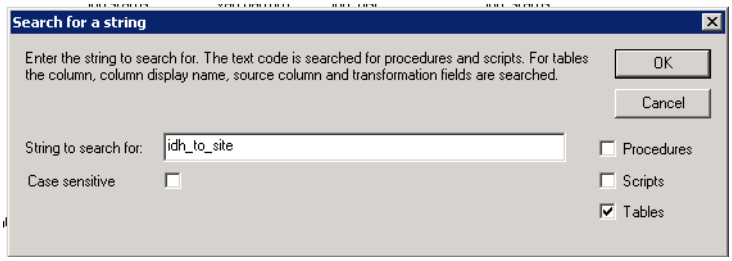
Identifying Every Appearance of a Column in Existing Code

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

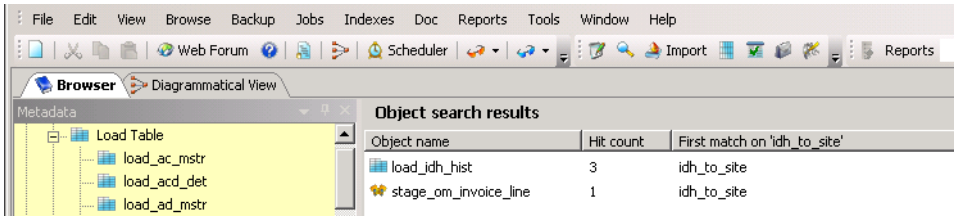
1. If you believe that the column is somewhere in the warehouse and it did not show up through the prior methods, you can search the code for the character string.



2. Depending on what you are looking for, pick the relevant boxes. In this case, only Table is the best choice.



3. The 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 using QAD-defined standards. This practice allows anyone looking at code, such as Support, Services, or customers, to easily identify the custom work and verify that the custom work was applied in a uniform way. We have three different categories of naming conventions, which are outlined below.

Custom Tables

If you create any tables, affix an x 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

If you add columns to the existing tables or new parallel path custom tables that eventually join with existing tables, prefix them with an xx_ to identify them.

Example If you were to add a column called recipient_date to an existing table in the standard release of the module, name it xx_recipient_date. The underscore in xx_ acts like a wild card in the DWD's Search string feature. Therefore, a search for any columns that contain x_ returns any value that contains an x, such as tax_code. By using xx_, you can dramatically narrow down searches to only those columns that contain xx, which should be nearly all customizations.

Custom Columns Not Joined to the Baseline Module Tables

Columns in new custom tables that are not going to be 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 columns 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 recommended customization methodology.

To avoid difficult upgrades and patch installs, custom work should reside in its own set of parallel tables. These parallel tables closely mimic the tables and logic already used to populate the dim_ and fact_ tables. If we already import data from the same source table as the requested custom column, then unless the existing data path is very complex, it is fairly simple to create a series of custom tables to handle the custom column. The new custom tables can follow a path in parallel to the existing stage tables to get the custom data to the destination; namely, a view created from the existing dimension or fact table, combined with the new custom dimension or fact table. Even under this methodology, there are exceptions to the rules.

Load Tables

Ideally, custom data should be loaded in custom tables (to avoid problems with upgrades to BI metadata that might overwrite custom changes). However, if the source of a new custom column is a table that is already loaded in the data warehouse, you can use the same load table and just add your custom column. Make sure, however, that you review the list of modified tables for any upgrades you apply to your metadata, as you may need to restore your custom column to the load

table if the load table is modified by an upgrade. So in an example as provided earlier where idh_prodline is added to the fact_om_invoice table, it should be added as a custom column xx_idh_prodline to the existing load_idh_hist table. If the existing warehouse does not utilize the required new columns from source tables, create a loadx_ table that contains all relevant columns for linking that data to the existing data warehouse (if that is the intent).

Perm Tables and Pre-perm Stage Tables

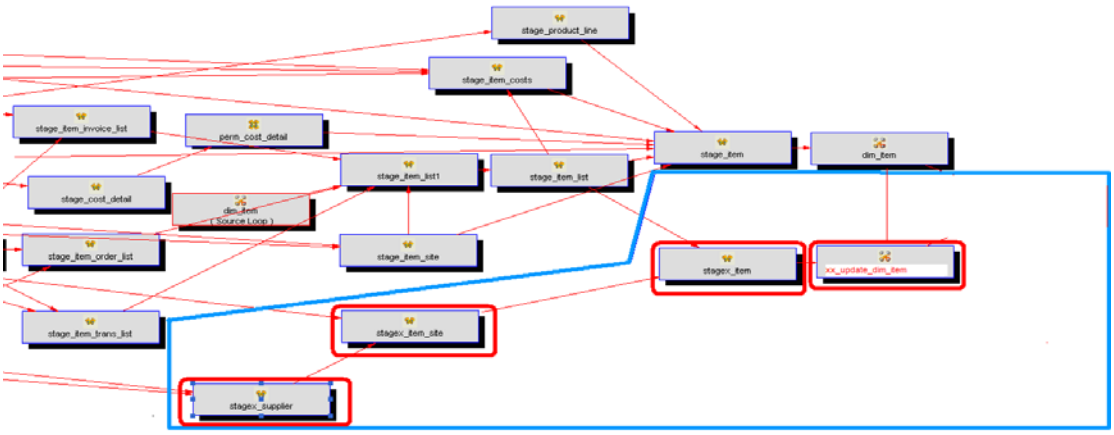
The data from the source table through the load, post-load/pre-perm stage and then perm table in the baseline code is left intact with few or no transformations. Changes to these tables in future releases are limited and require a minimal number of required changes to then re-add custom columns to this handful of tables, if a patch or upgrade disrupts their infrastructure. As a result, 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 must 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

Adding 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. Copy the logic of the WHERE clause that the new table mimics and use it in the new custom path tables logic. This action ensures 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 should be blank. Transform those column values to some dummy value. 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.



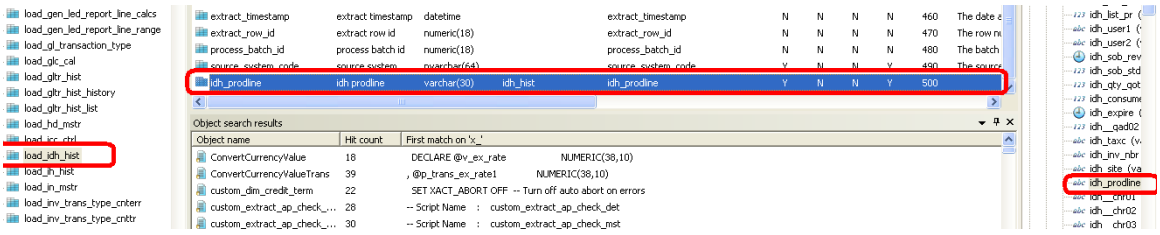
Adding Additional Columns to a Load Table

To add a column to a load table that comes from the same source, click the load table to contain the new column. This action allows you to see all of the columns for that table in the target pane.

Then do the following:

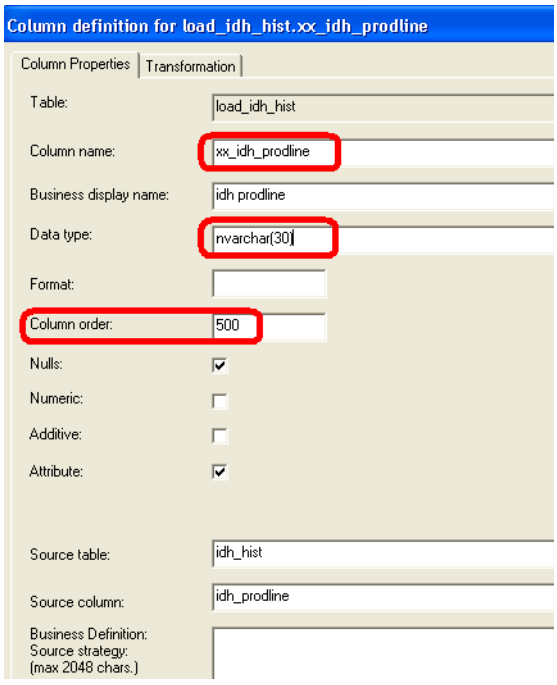
Get Column from Data Source

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



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 a nvarchar datatype.

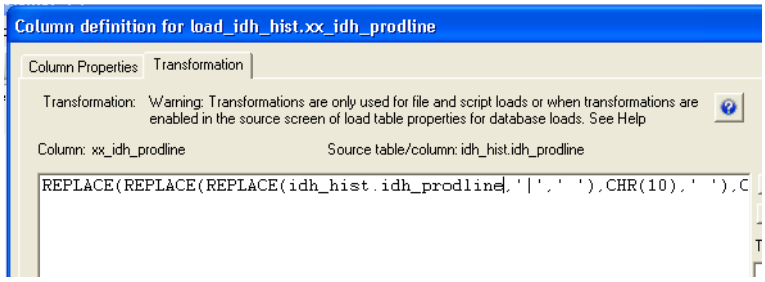


Transform

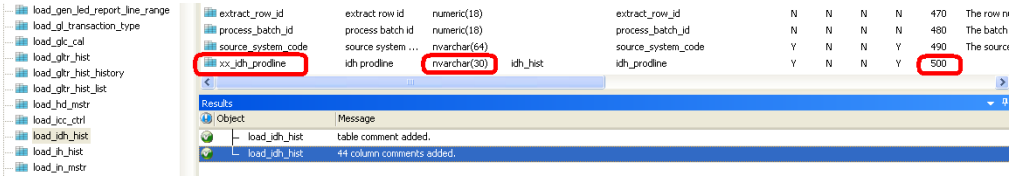
7 If the column is originating from any type of char, varchar, or nvarchar datatype column in the source, perform a transformation on the column to strip out blank spaces and | symbols, which disrupts data loads. Do this cleanup using the following commands, which should be found in any other varchar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

REPLACE(REPLACE(REPLACE(idh_hist.idh_prodline,'|',' '),CHR(10),' '),CHR(13),' ')

This is placed using the Transformation tab.



8 The result is seen below. Scroll to the far right to see the transformation.



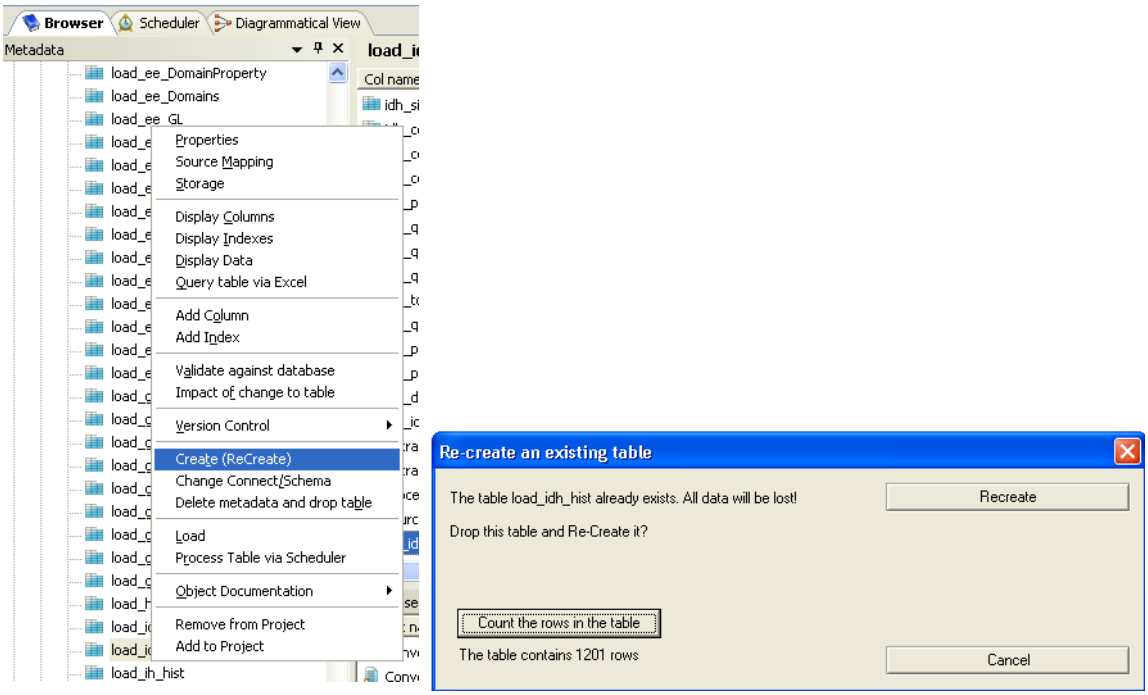
Create (Recreate) the Table

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

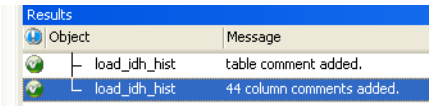
Warning Do not use Create (Recreate) for Dim, Fact, or Datastore/Perm tables unless you are sure that they are empty. For dim, fact, and perm tables, you may need to do a Validate/Modify process instead. See “Modify Table” on page 200 for an explanation of the Modify table process.

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

11 When the window comes up, check the number of records in the table you are about to drop by clicking the Count rows in table button. If you are ready to move ahead, click Recreate.



12 A confirmation that the table re-created correctly appears in the Results pane.



Adding Additional Custom Columns to Pre-perm Stage Tables

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

Then do the following:

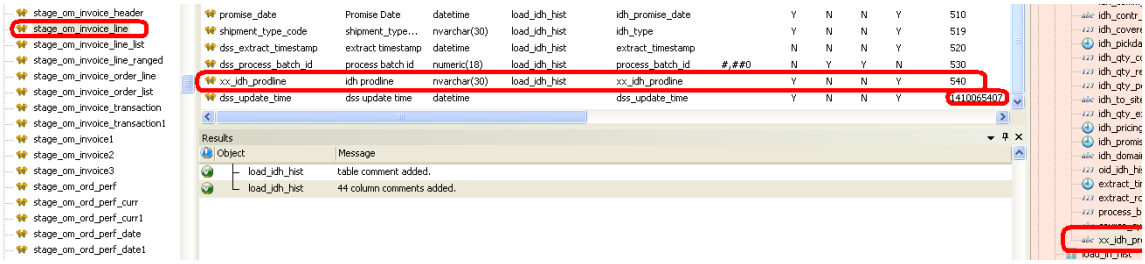
Connect

1 Connect to the Data Warehouse in the Source pane and find the appropriate source table.

2 Double-click the table and find the column to add.

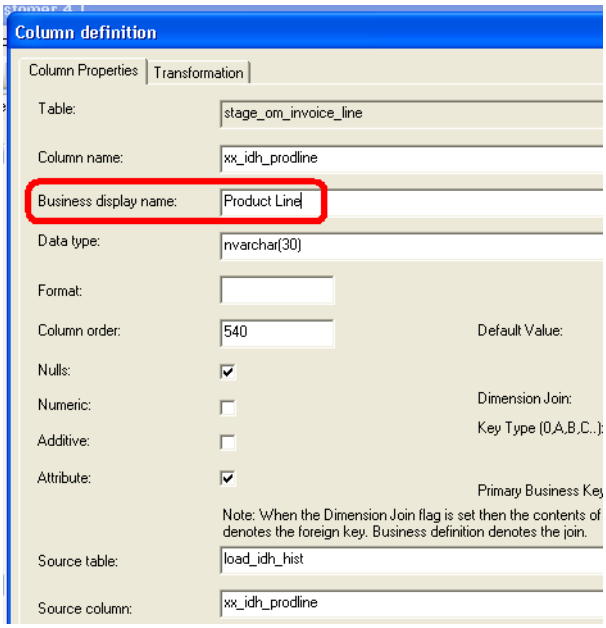
3 Click the column and drag it to the Target pane at the end of the column list.

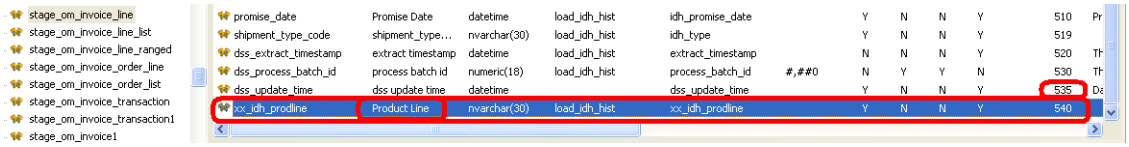
Note The DWD did not place the column at the very end of the column list. It is easier to edit the last column (dss_update_time) and change the column order date for it to 535 so that xx_idh_prodline is last.



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.



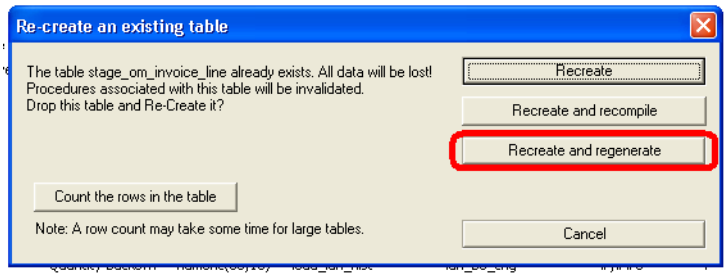


Create (Recreate) the Staging Table

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 are needed.



Adding Additional Custom Columns to perm/dim/fact Tables

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

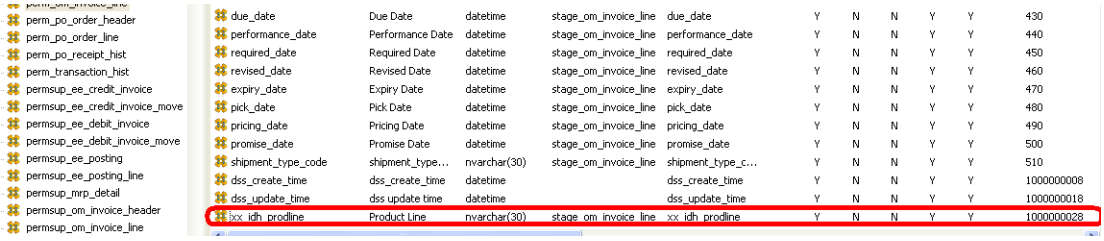
Then do the following:

Data Source

1. Open the Source Pane using the Data Warehouse ODBC connection.
2. Find the source staging table to reference.
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 bottom of the column list.

Placement

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

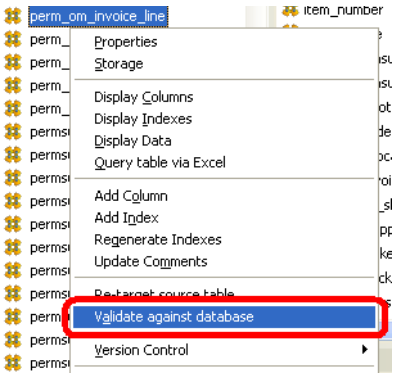


Column Table Source and Transformation

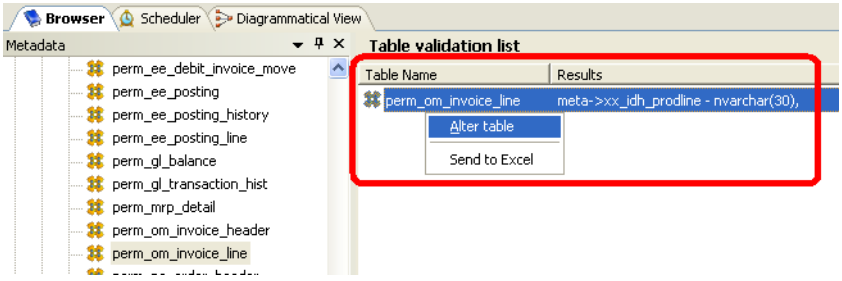
6. If this column was added to a dim_ or a fact_ table and is updated by a procedure, remove the Source Table from the column definition and enter a dummy value in the column Transformation. This ensures that the recompilation of the procedure affiliated with this table does not break.

Modify Table

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



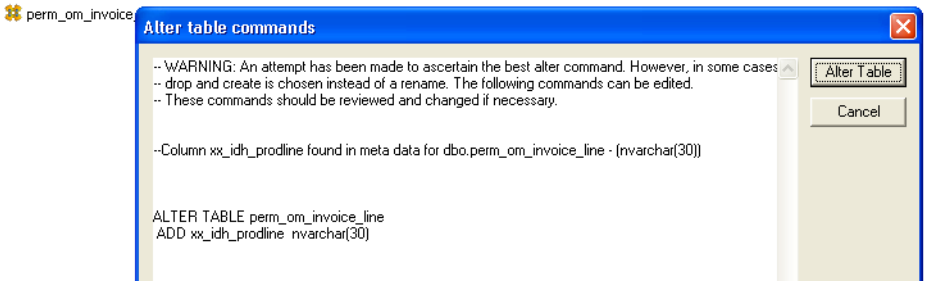
8 A list of new columns appears in the Results part of the Table validation list in the Target pane. Right-click on the list and choose the Alter table option.



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 because sometimes it creates commands that drop all the columns in the table. This is true of dim_ tables.

You can remove any excess incorrect commands. The only commands that you should use here are to ADD columns or possibly modify data types.



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

Rebuild the Procedure

11 Double-click the table that was modified. If there is **not** an indicator to the right of the Regenerate button that the procedure was *MODIFIED*, click Rebuild. Then 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, modify it manually by editing that custom procedure.

Adding Custom Path Stage/Extract Tables After the Perm Tables

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

Then do the following:

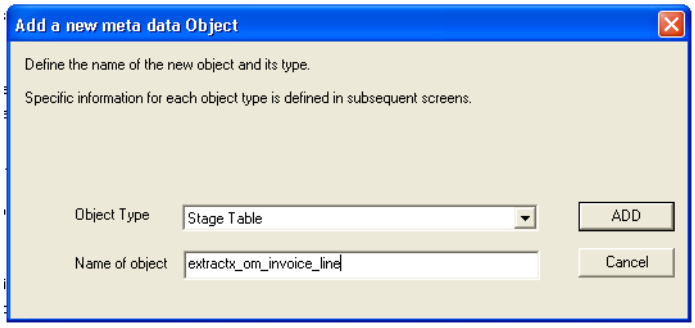
Drag and Drop New Table

1 In the source pane, be sure that 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.



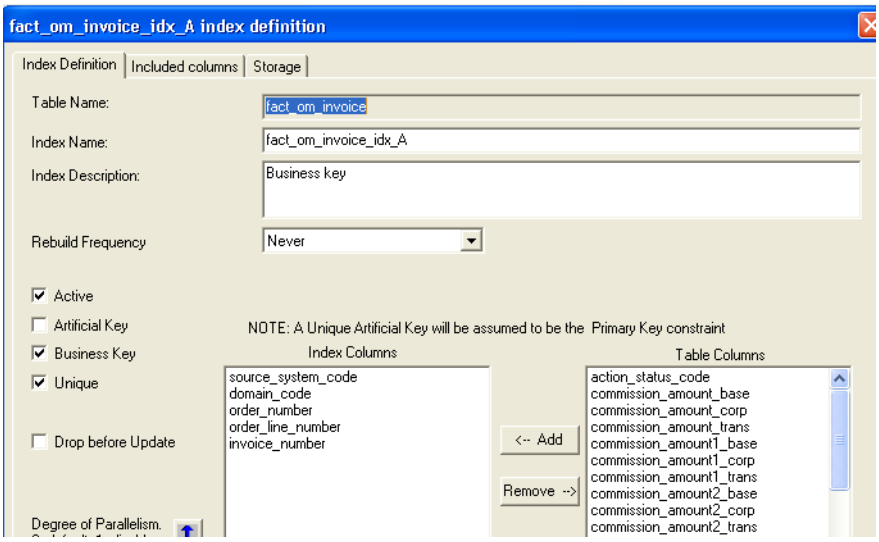
Identify and Set the Columns

5 Identify the components of the primary key that this table uses to later connect with a dim or fact.

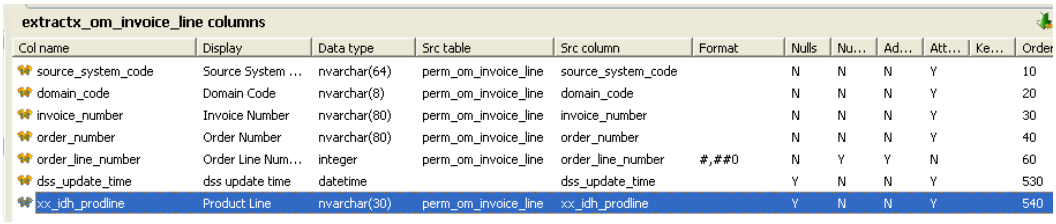
6 Delete all columns except those columns, the new custom column, and any critical date columns at the end of the list.

7 To identify the primary key components, go to the destination table to mimic and Display Indexes.

8 Pick the Business key index from the list.



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

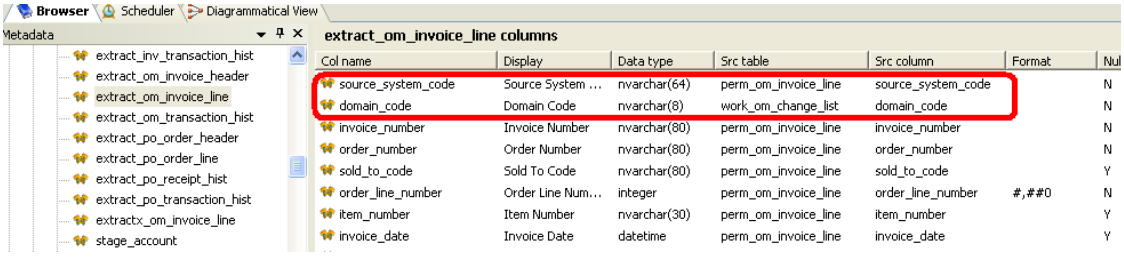


Col name	Display	Data type	Src table	Src column	Format	Nulls	Nu...	Ad...	Att...	Ke...	Order
source_system_code	Source System ...	nvarchar(64)	perm_om_invoice_line	source_system_code		N	N	N	Y		10
domain_code	Domain Code	nvarchar(8)	perm_om_invoice_line	domain_code		N	N	N	Y		20
invoice_number	Invoice Number	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_prodline	Product Line	nvarchar(30)	perm_om_invoice_line	xx_idh_prodline		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 Use in the New Procedure

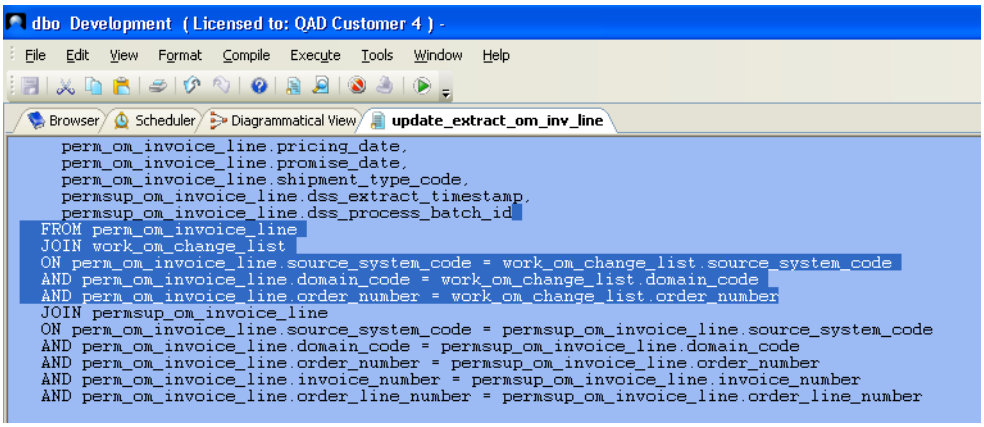
11 Go to the parallel table, if there is one, and view its column sources and its procedure code. Note in the column sources below the inclusion of work_om_change_list.



Col name	Display	Data type	Src table	Src column	Format	Nul
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
sold_to_code	Sold To Code	nvarchar(80)	perm_om_invoice_line	sold_to_code		Y
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 the table name in the Browser pane and choose Code -> View <code name>.

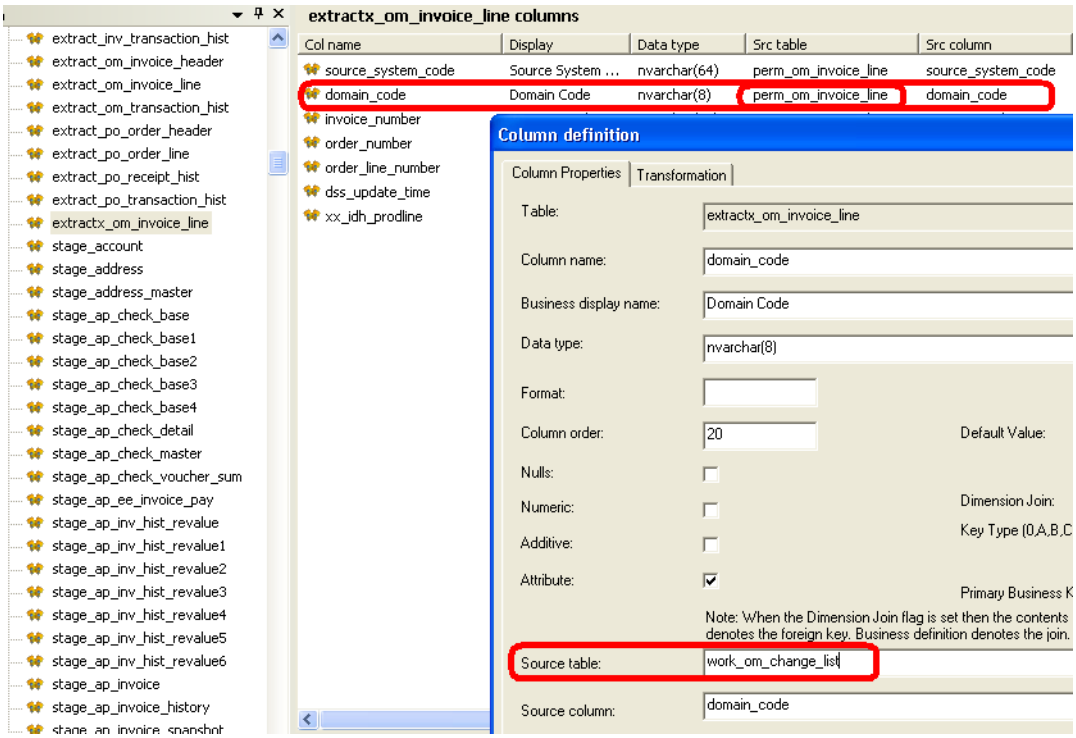
13 In the code screen, scroll down to the FROM/WHERE clause of the query, highlight it, and copy it into a notepad. In this instance, we do not need the permsup_ table data as that data is passed through in the standard modules.

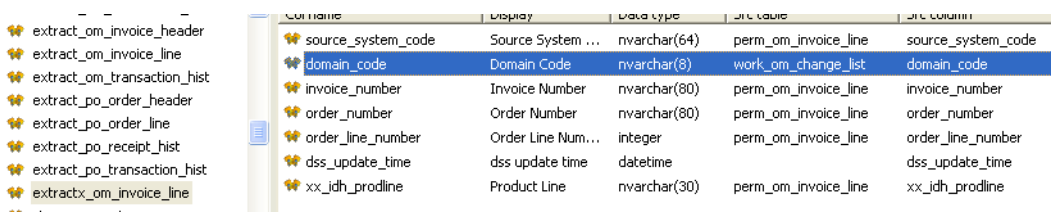


Change the Source Tables

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. At least one of the columns in this new table must be defined with a source from work_om_change_list. Consider each custom table that is set up differently based on what the original baseline table does. It is also possible that there are several baseline tables and the custom path merges all their logic into one table for expediency.





Create (Recreate) the Table

15 Right click the table name and pick Create(Recreate). Create the table.

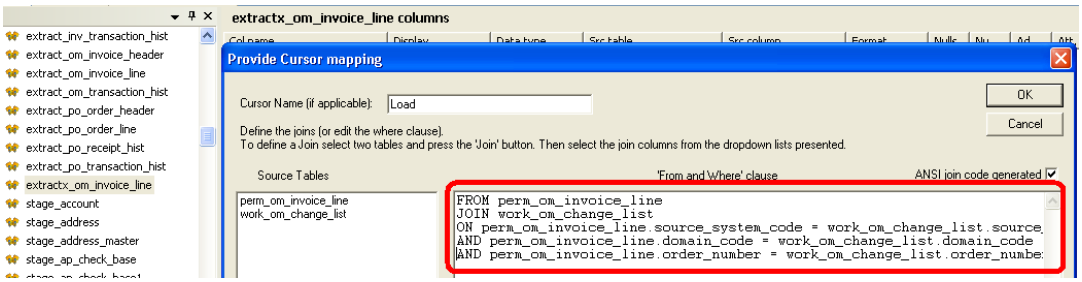
Build the Procedure for the Custom Table

16 Double-click the new custom table name in the Browser pane to open up the Properties window.

17 Click Rebuild.

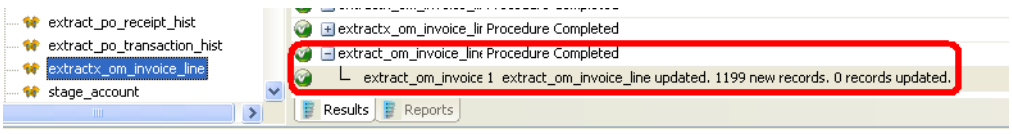
18 Follow the standard settings through the first two screens, unless a parameter is required that you would see in the originating table.

19 At the third screen, paste in the code copied in “Identify the Code to Use in the New Procedure” on page 202. Then click OK.



Load the Table

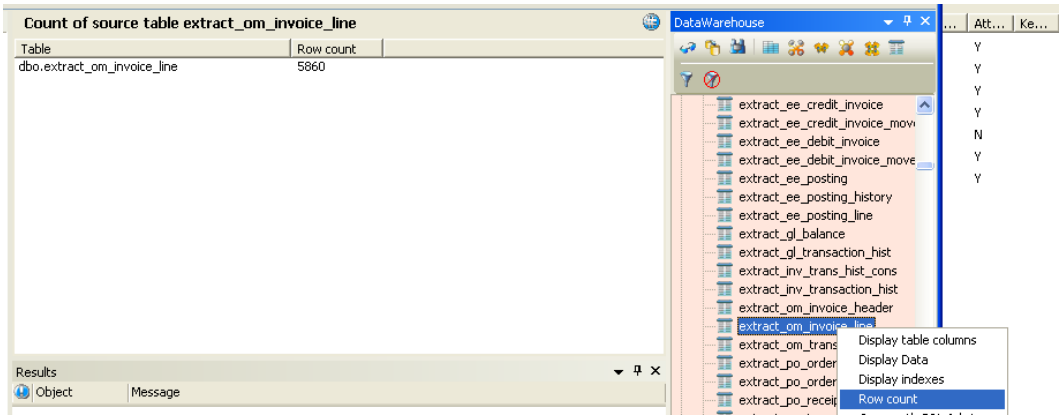
20 With the new procedure compiled, load the table by right-clicking on it in the browser pane and clicking the Execute Update Procedure option. Note the results quantity in the Results pane.



Confirm Results

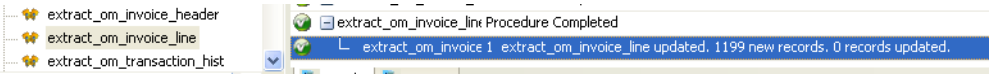
21 Confirm that the number of records matches the correlating original table.

22 Refresh the Source pane and right-click on original and select Display table count. That number should match the number loaded in “Build the Procedure for the Custom Table” on page 204.



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, extract the original procedure query and the parallel procedure query. Start working through them in SQL to see what part of the FROM/WHERE clause is 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

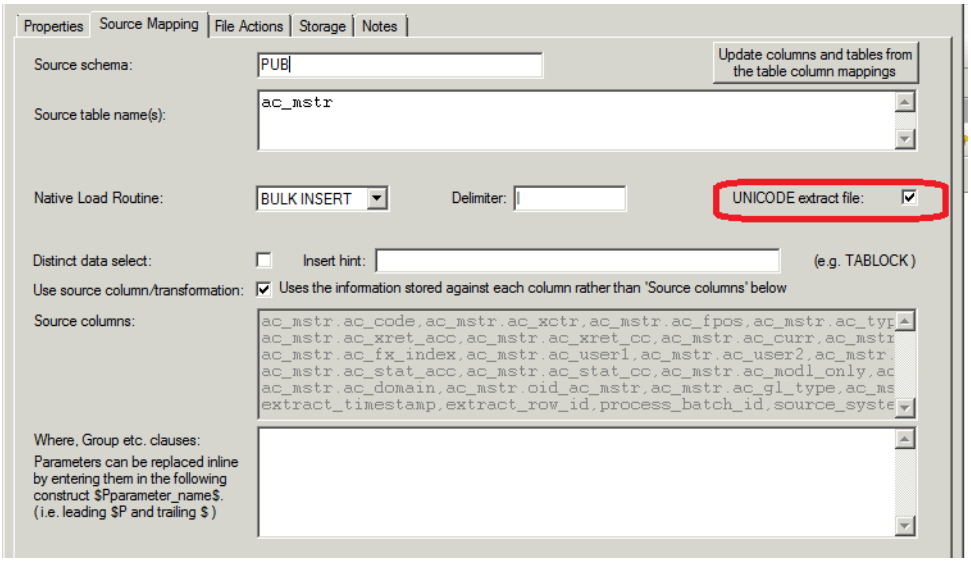
If a customer requires source tables for data that is not in the warehouse, and you do not have matching load tables for these source tables, you can add and integrate them into the warehouse as custom loadx and stagex tables.

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 does not 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, use the wizard 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 necessary to update the existing procedure for that dim/non-history fact.

- **Dealing with Custom/Modified Code**: Carefully add the column 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 Custom or Modified, and
  - the new column should be populated by a prior table; for instance history fact tables.

- **Populating New Custom Columns in *Empty* Perm/Fact Tables:** If custom columns were added to empty perm or fact tables, the tables re-created, and their procedures regenerated, these tables should populate normally as part of the historic load. In some instances, there should be custom tables created to mimic the original historic load for the population of fact tables.
- **Populating New Custom Columns in Already Populated Perm:** If a new custom column is added to a populated perm table, create a script to update that perm table for all of the history records it contains. An exception is the perm_om_invoice_header and perm_om_invoice_line tables, which are repopulated every day. This is an interesting exercise for the large tr_hist and gltr_hist and their correlating perm tables, especially if the customer has archived some of the data already 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 can 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 209.
- **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 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 209.
- **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. If the new tables added to a history fact existed in the perm table, you probably must perform a historic reload from perm that includes the new custom path staging tables. See Appendix C, “Historically Reloading Modules,” on page 209.
- **Cloning Load Tables to Bring in Outside Data Sources (Flat Files or Other Databases):** If you add data from an outside data source to the data warehouse, the customer must 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. Create a post-load procedure 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. The procedure then 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, set up a Group called Custom Work with two projects:
  - Custom Work – Modified Objects
  - Custom Work – New Objects

  Give the projects the extended name, not Modified Objects, or New Objects, because the DWD tool has some issues around confusing same named Projects. Allocate any custom work done in that environment to the appropriate Project that you 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.

Appendix C

Historically Reloading Modules

This chapter describes how to historically reload modules:

Historic Reload Instructions 210

Historic Reload Instructions

Checklist

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

- Is this task a complete reload from the ERP or is it a historic reload from the perm tables? A complete load of the ERP is required only if there is a problem with the data in the relevant perm tables, such as perm_transaction_history. It is a 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, account for extra considerations to ensure that parameters are set properly. This action ensures 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 load all facts for the module? If all of the fact tables need reloading, fewer steps are necessary because you can run the relative historic load job for that module. If only one table needs loading, clone an existing job and remove all of the parts related to the loading of other fact tables. Leave in only those items 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 task 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 Support.

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 step is to populate certain load tables the module loads need, without trying to guess which ones are populated correctly. This is only applicable for a multi-source system. If one source has populated a load table and a different source is run for the modules, the data is not loaded correctly. Specific examples are load_si_mstr and load_en_mstr, which OM, PO, and INV references, 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 is loaded, then go to step 8. If all the fact tables are reloaded, run HIST_<module>_<submodule>_PROCESS_CHAINED. Go to Step 9.

8 If only one fact table is 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 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, you could name the job 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 tasks that are not relevant to the loading of that table. The tasks to remove include the stage and fact table for fact_om_booking, fact_om_shipping, and 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 unclear which tasks to remove, consider removing the non-relevant fact tables from the list so that the historic reload does not try to repopulate them unnecessarily.

e Run the newly created one time job.

f Go to step 10.

9 Assuming all facts should be loaded, then 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 can cause the run to take much longer than normal. Once the daily run has completed, set the INV_PROCESS_DAYS parameter back to a much lower number like 5 or 10.

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

Appendix D

Manual Installation

This document covers the manual installation of the DWD, the BI Modules, and the BI Web Application. Topics include:

Introduction 214

Manual Installation Requirements 214

Manually Installing the Data Warehouse 214

Installation of DWD on Separate Server than SQL Server 222

Manually Installing the QAD BI Analytical Modules 226

Manually Installing the QAD BI Web Application 232

Introduction

This appendix covers installation scenarios that the GUI installer does not support, such as installations where the Data Warehouse Database is on a different server than the Portal database.

Manual Installation Requirements

In addition to the standard product requirements, manual installation requires that Tomcat 5.5, 6, or 7 be installed and configured.

Note Set Tomcat's Java maximum memory pool to at least 1024 MB of RAM. If multiple Web applications are running, or if you are installing dashboards for multiple modules, you may need to increase the Java memory to 2048 MB of RAM.

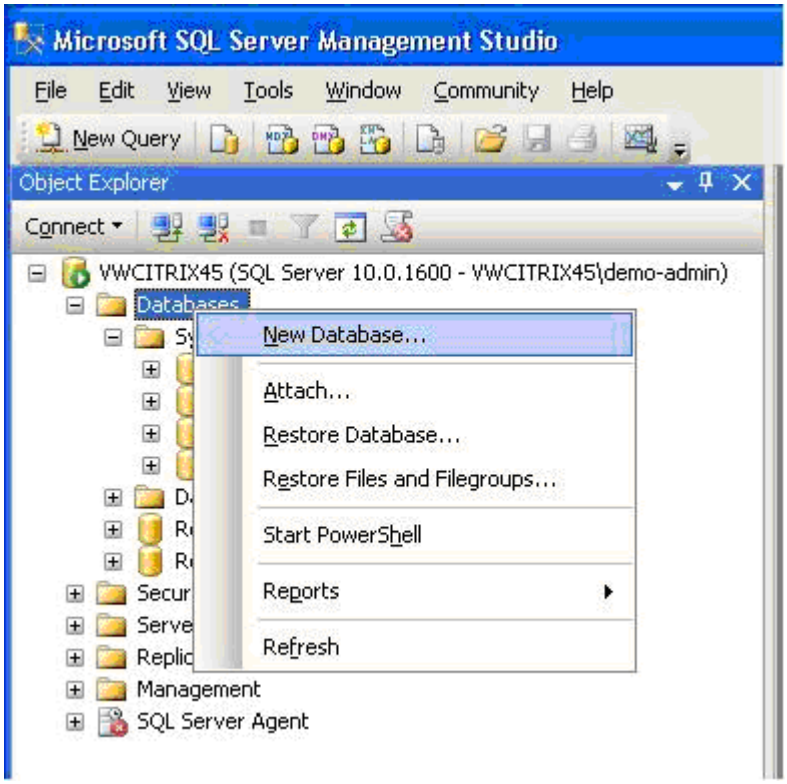
Manually Installing the Data Warehouse

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

Follow these steps to install and configure a new QAD BI Data Warehouse instance manually.

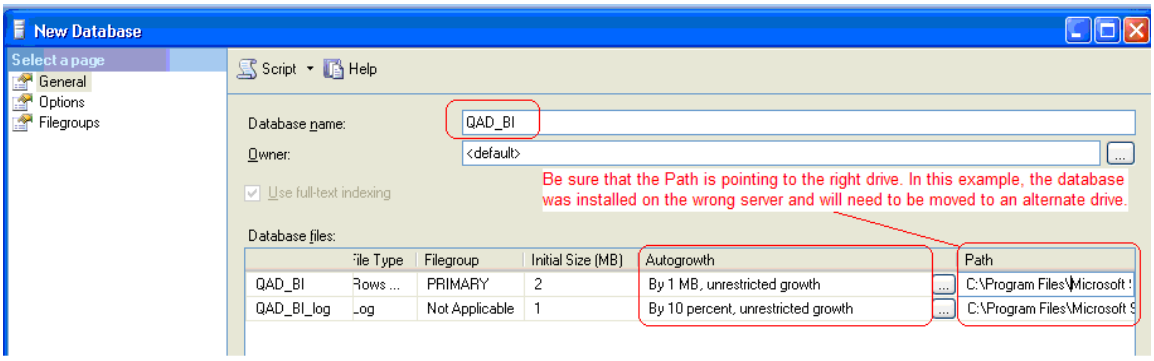
1. Create a database in SQL Server.
   a. From the Microsoft SQL Server Management Studio, right-click on Database, Create new database.

Fig. D.1
Create New Database



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

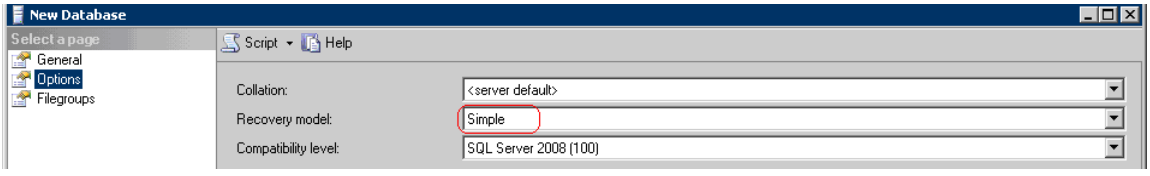
Fig. D.2
Create New Database



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

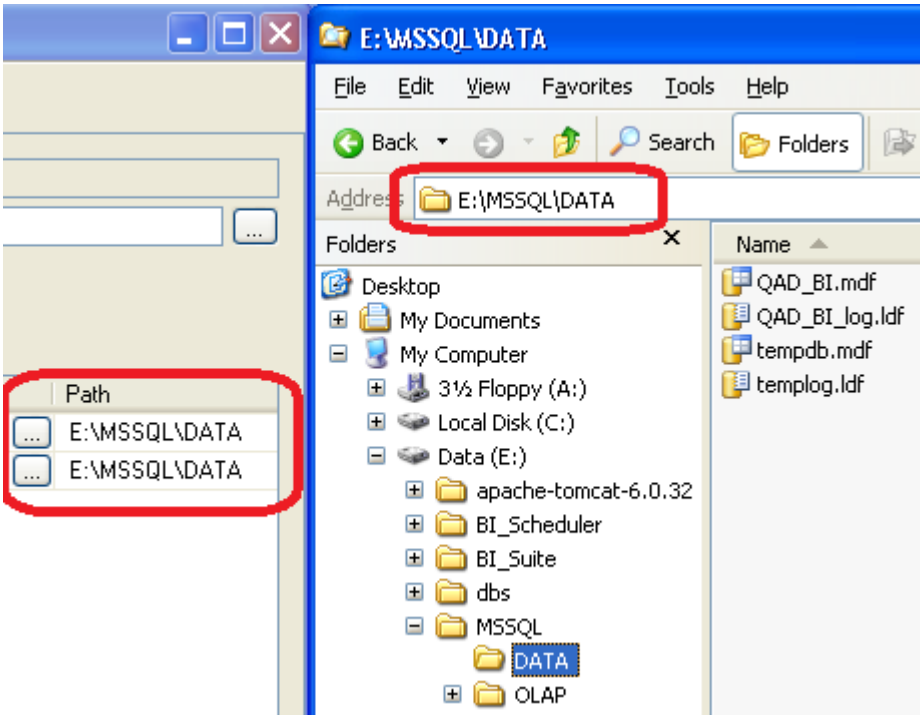
Important Make sure that autogrowth is set in a way that the SQL Server grows the database in appropriately sized chunks. For example, if 1-MB unrestricted growth is set, the database must stop the current process and allocate an additional 1 MB of space every time the database grows by 1MB. If the database grows by 1 GB, because that amount of new data is coming in, the SQL Server must increment the size of the database by 1 MB one thousand times as it tries to load the 1 GB of data. This action severely slows the load process. In this example, a more reasonable setting for autogrowth rate would be 1 GB. Make sure that the database size is set appropriately large as a starting point.

Fig. D.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 folder on the correct server. Once created, enter that directory as the Path for the database.

Fig. D.4
Create New Database 4



d Following up on step c, if the Path was wrong for the database, check the temp database 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 this example; 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 To accept the setting changes, 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.

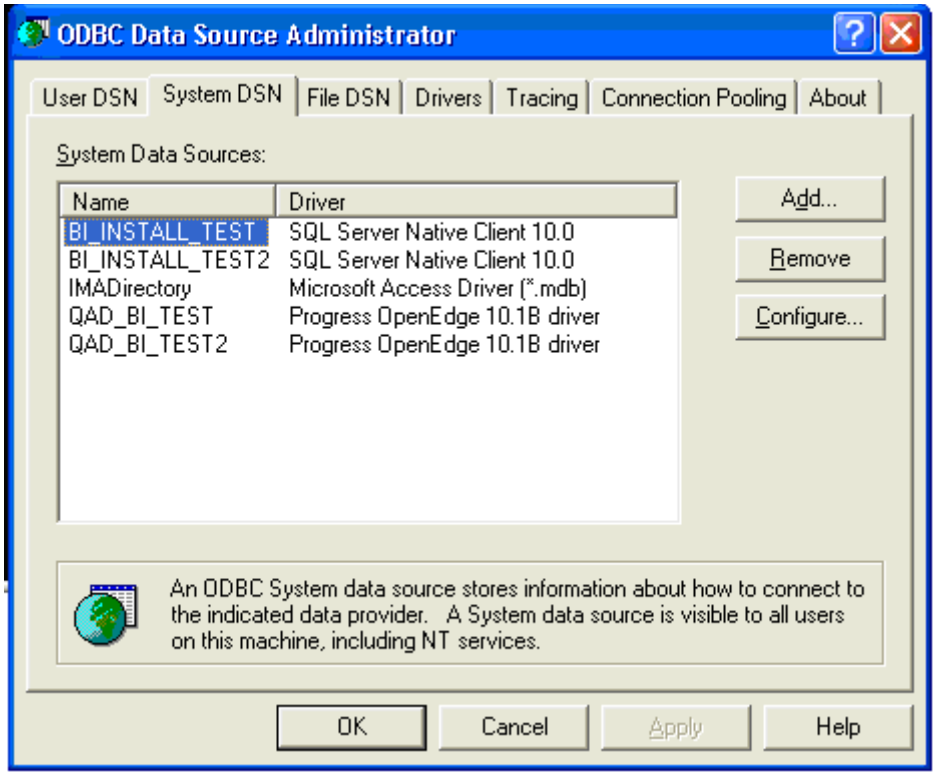
3 Launch the ODBC Data Source Administrator from the QAD BI Administrator so that ODBC connections to the Data Warehouse and to the Progress databases can be created. Using the BI Administrator in the PC Setup tab, select the Validate/Add Odbc Source option. In the resulting ODBC Source screen on the right, right-click and select Add Odbc entry to launch the ODBC Data Source Administrator.

Note Data Warehouse Designer (QAD BI DWD) uses the ODBC to communicate with the Data Warehouse database. It also allows the DWD to communicate with the QAD Progress databases and extract data from them for use in the data warehouse. Persons familiar with the ODBC Data Source Administrator can launch it through other means. However, on a 64-bit system, it is possible that the correct method is not being launched. Use the appropriate 32-bit ODBC driver, `odbcad32.exe`. 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 connection is a System ODBC connection using the SQL Server Native Client driver and Windows Authentication:

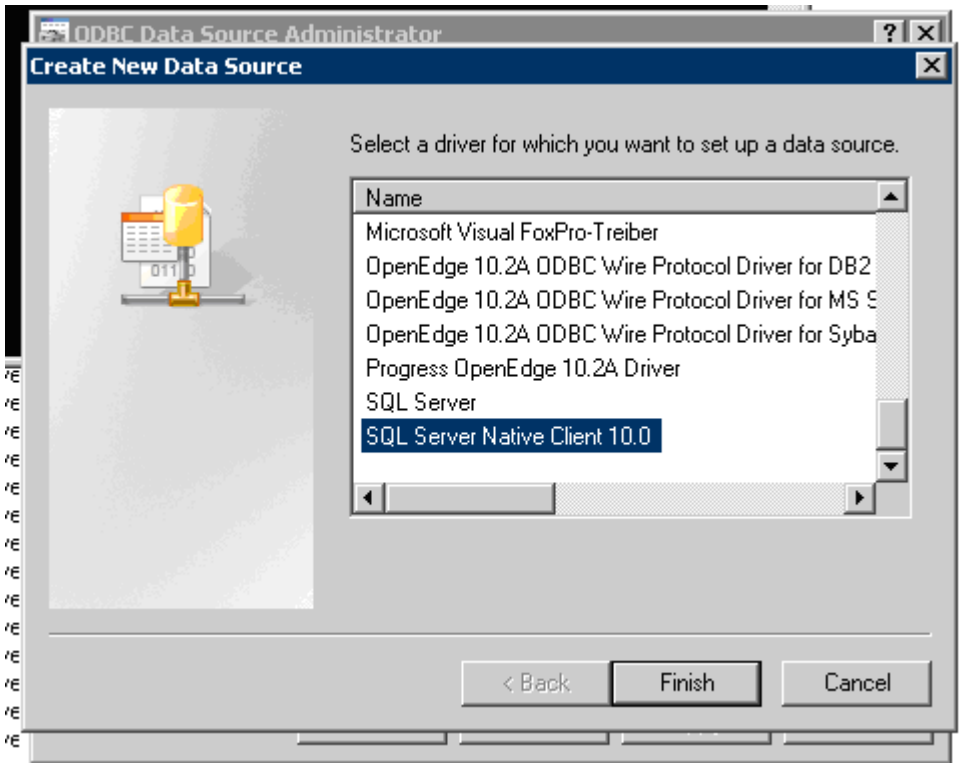
a Select the System DSN tab and click Add.

Fig. D.5
Create ODBC SQL1



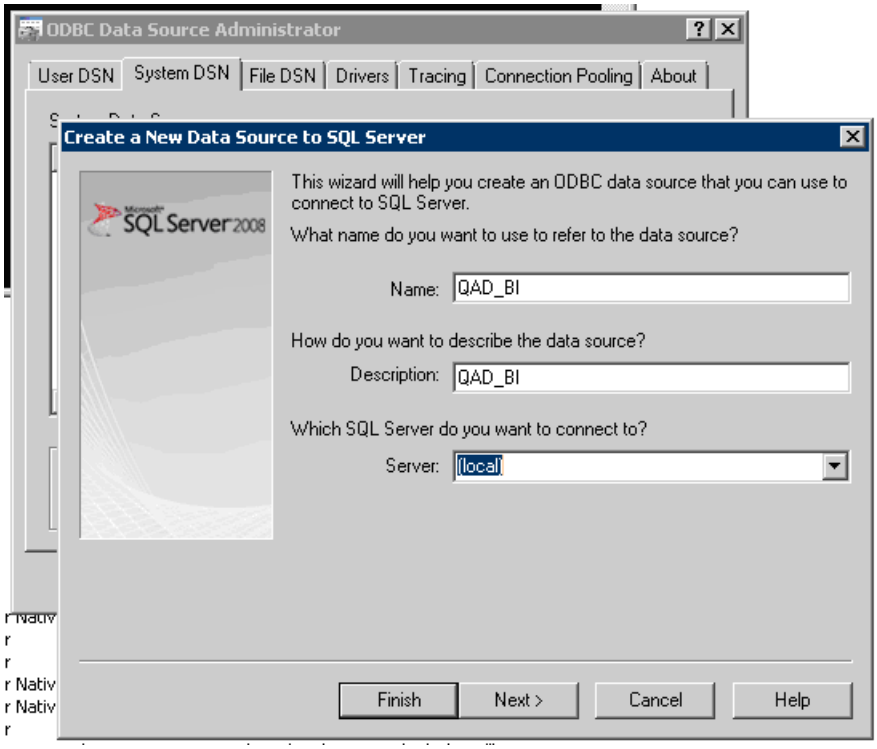
b When Create New Data Source is displayed, select SQL Server Native Client 10.0 and click Finish.

Fig. D.6
Create ODBC SQL 2



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

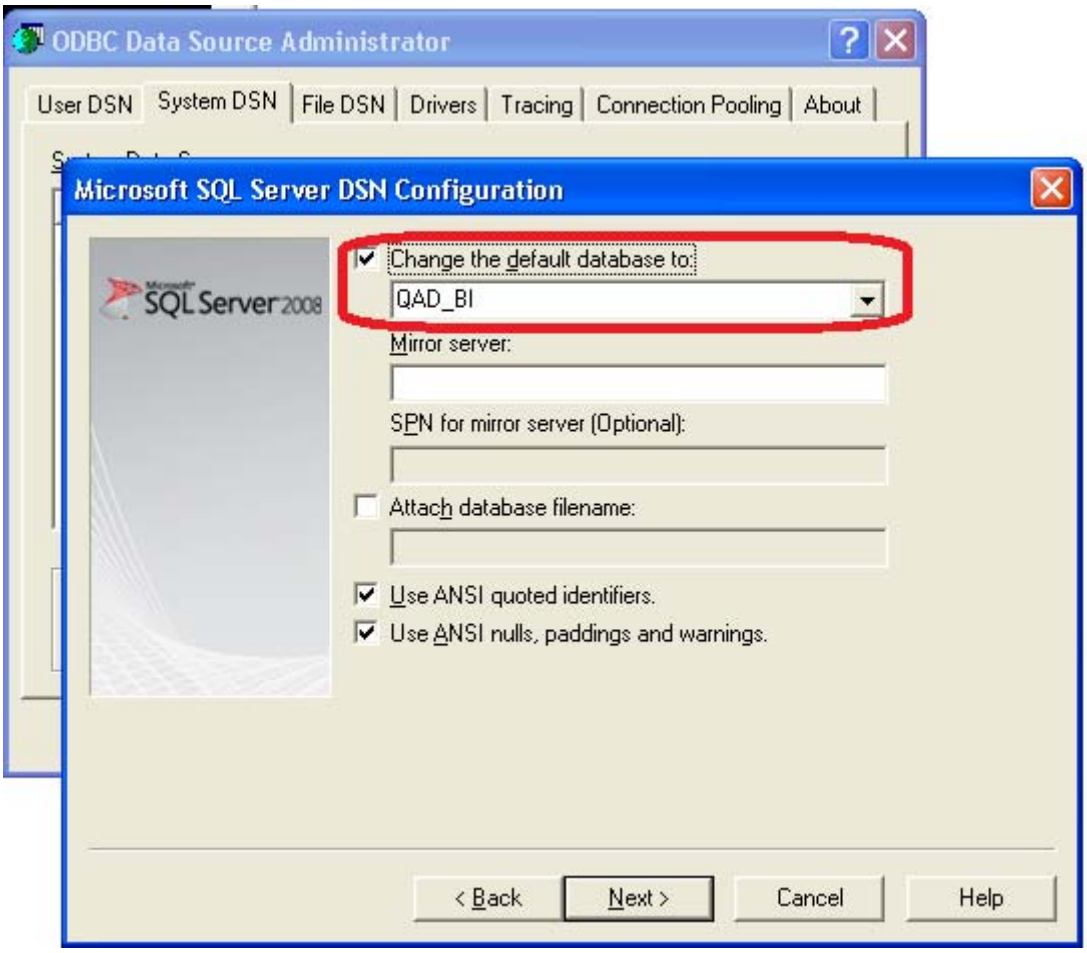
Fig. D.7
Create ODBC SQL 3



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. D.8
Create ODBC SQL 4



f Click Test Data Source.

g Confirm that the database connection works properly. If successful, click OK.

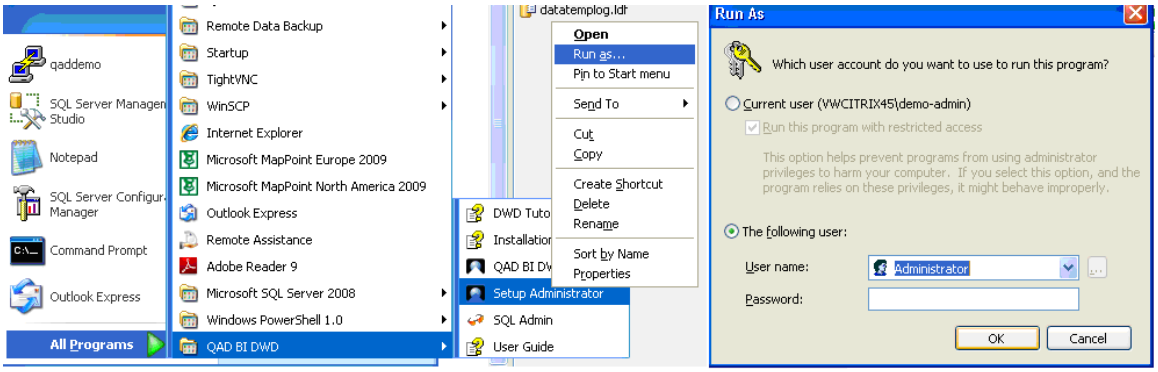
5 If they do not exist, use the Progress ODBC driver to create the ODBC connections to the Progress database that the data warehouse extracts from. The Data Warehouse Engine (QAD BI DWD) uses the ODBC to access each Data Source. You need the following information:

- If they have a preferred connection name
- Host name
- Port number
- Database name
- User ID
- Password

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

6 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.

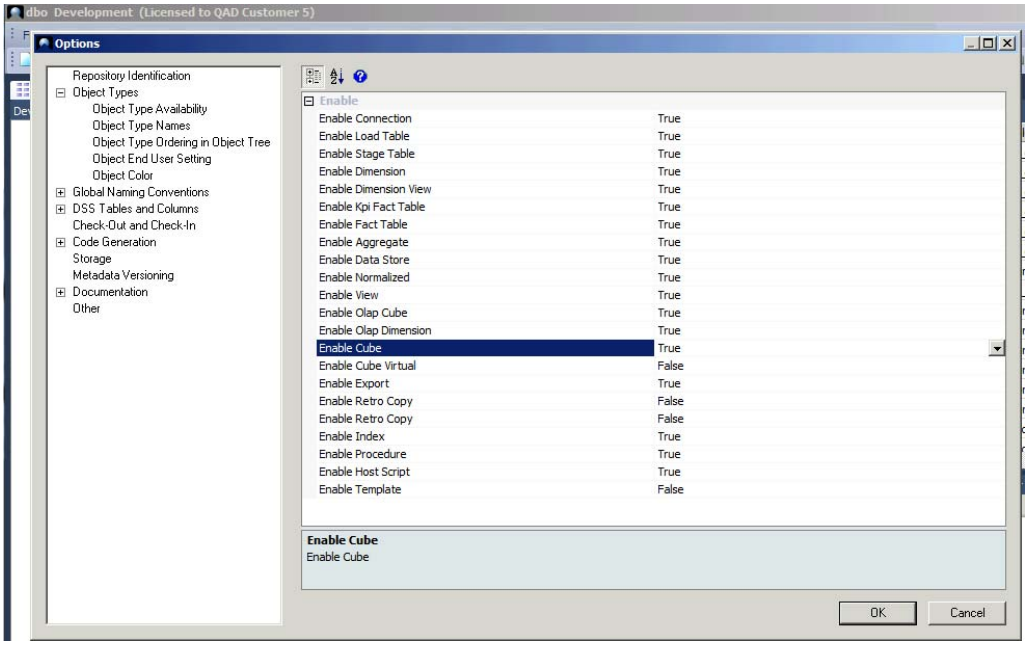
Fig. D.9
Setup Administrator



7 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 must be All Licensed Objects.

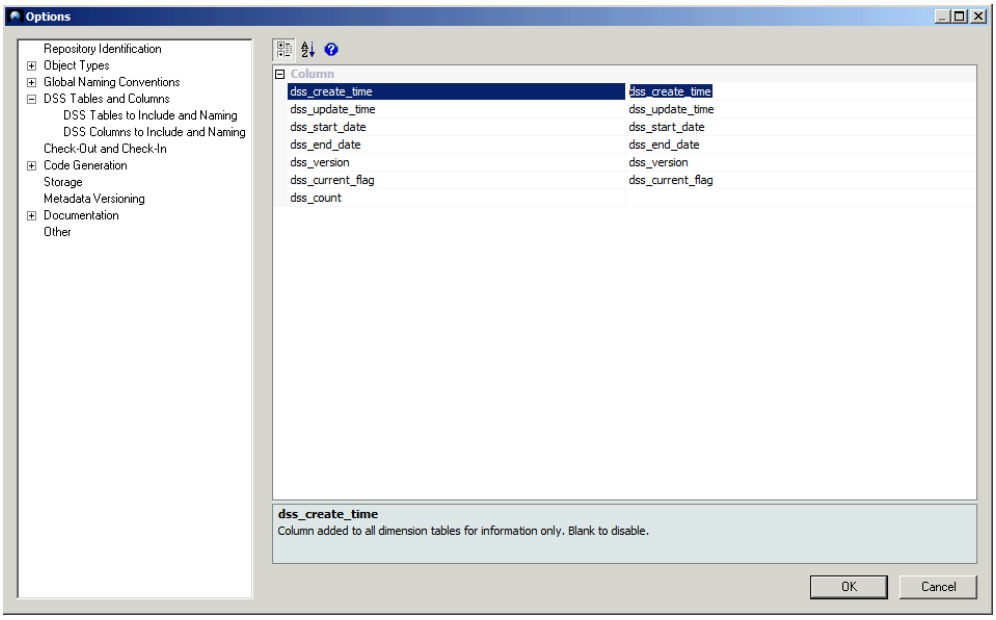
8 When QAD BI DWD is launched after the repository is created, activate Cube objects. From the Tools navigation menu, choose Options|Object Types|Object Type Availability. Click Enable Cube and select True from the drop-down list.

Fig. D.10
Cube



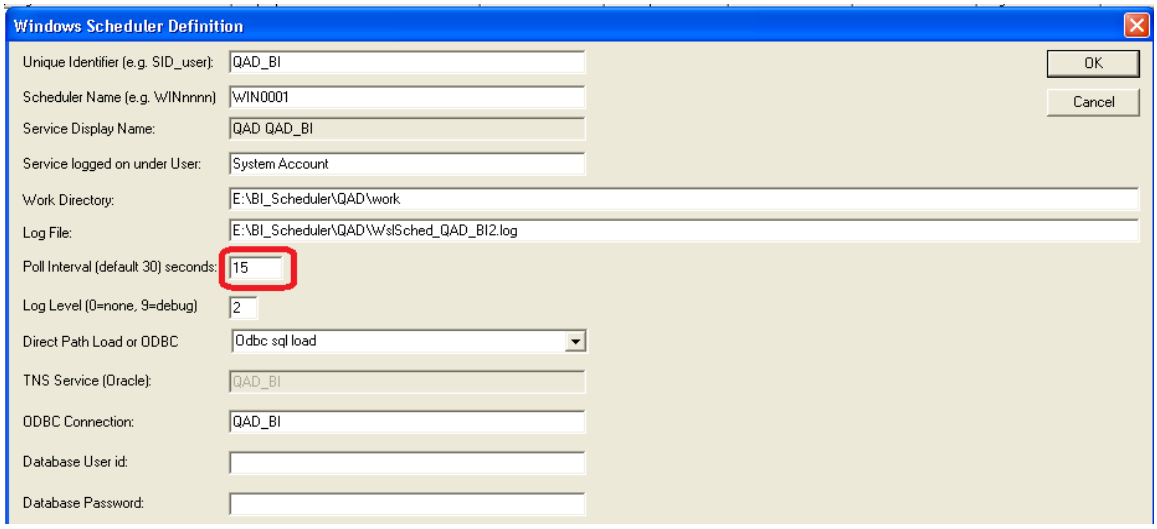
9 While in the Options window, choose DSS Tables and Columns|DSS Columns to Include and Naming. Enter dss_create_time in the dss_create_time field and then click OK to close the Options dialog box.

Fig. D.11
DSS Columns Setting



10 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. If they do not gracefully close quickly after use, any time less than 15 seconds raises the possibility of overloading the scheduler jobs. Thirty seconds is acceptable, but due to the large number of scheduled jobs, reducing this interval is a significant time saver.

Fig. D.12
Poll Interval



Installation of DWD on Separate Server than SQL Server

Accessing Defined Work Directory for the DWD Connection

For the BI jobs to run from the Scheduler, the jobs must access the defined Work Directory for the DWD connection. Because the DWD and Scheduler reside on separate servers, the SQL Server domain user (the Windows user account) that runs the SQL Server service and the BI load jobs must be able to access the DWD Work Directory on the application server.

The SQL Bulk Insert operation used to populate the BI load tables uses the SQL Server service account to access the file share.

Follow these steps to access files from a remote machine:

1. Change the service account from a local account to a domain account in SQL Server Management Studio in the Object Explorer pane or under the specific BI database name:
   - **In the Object Explorer pane**: Under Security|Server Roles|BULKADMIN, add the user in question, such as sqladmin, as a Role Member.

     Or,
   - **Under the specific BI database name**: Under Security|Logins, right-click the username. Select Server Roles and select the bulkadmin check box. This action gives effective permissions to the SQL Server domain user.
2. Add the SQL Server domain account to the DWD share's (server's) permissions for the DWD Work Dir:
   a. On the DWD server, create a Windows user with the same name and password as the account that runs the SQL Server service.
   b. Right-click the DWD Work Dir folder, and go to Properties|Sharing|Advanced Sharing.
   c. Add read and write permission for the new SQL Server service account.

Setting Up a Split Server System

This section provides detailed information on the installation of the data warehouse and the DWD on different servers. Setting up a split server system can be tricky. You may need extra time to get all of the configuration pieces working with each other. Because it is rarely a smooth process, expect it to take at least a half day to get the split server system working properly.

In a split server system, the SQL Server, the data warehouse, and the associated database reside on one server. In this example, this server is called Server A. The DWD and Scheduler reside on another server, which is called Server B.

Follow these steps to set up a split server system:

1. The new BI data warehouse is created in SQL Server on Server A. There are two different approaches to perform the setup:
   - A user is created on the Server A box and affiliated with the database. The same user with the same password is defined on the Server B box.

- The alternate option is to set up a SQL Server user in SQL Server on Server A and use the login/password info for that user in all subsequent component configurations.

2. After creating the SQL Server database, set up a user for the Scheduler/DWD to use. Make sure that the user is given BulkAdmin privileges in order to insert data into the database.

   Note Sysadmin should work, but it depends on how strict the customer is with their security.

3. On Server B, make sure that there is a SQL Server client available with which to connect to the database.

   To make sure that the SQL Server client is available, run `sqlcmd` from a command prompt on Server B. If the system returns a `1>` or if it does not return a command (no error found), then the SQL Server is on the machine.

   If the SQL Server is not on the machine, install the SQL Server Client Services (or SQL Server Management Studio) with access to the Server A data warehouse.

4. Install WhereScape on Server B.

5. Follow these steps to use the Admin tool to set up an ODBC connection to the Server A database:

   a. Launch the ODBC Data Source Administrator from the QAD BI Administrator so that you can create ODBC connections to the Data Warehouse and to the Progress databases.

   b. Using the BI Administrator in the PC Setup tab, select the Validate/Add Odbc Source option.

   c. In the resulting ODBC Source screen on the right, right-click and select Add Odbc entry to launch the ODBC Data Source Administrator.

   d. Select the System DSN tab and click Add.

   e. Enter the name of the new ODBC connection. QAD recommends that you enter the same name as the data warehouse database name. If desired, add a description.

   f. **Important**: In the same screen as step e, choose the SQL server name you are connecting to. The name of the SQL server instance should include a prefix of Server A to identify that you are connecting to the correct machine and correct SQL Server instance.

      If you do not know the Server name, you can get it from customer or from the SQL Server instance on Server A. The server name is listed under Object Explorer.

   g. Click the Next button.

   h. Choose SQL Server authentication instead of Windows Authentication unless the customer chooses to set up all the permissions for the two to work with each other.

   i. Enter the Login ID and Password for the Server A SQL Server user, unless the customer was able to configure Windows Authentication. Click Next.

   j. Select the Change the Default Database To: check box. Then select the data warehouse database from the drop-down list. Click Next.

   k. Click Finish.

   l. Click Test connection and verify that the ODBC connection works properly.

Note If the ODBC connection does not connect, make sure all the ODBC connection parts are set up properly.

m Exit the ODBC administrator tool.

n Install the metadata. See “Manually Installing the QAD BI Analytical Modules” on page 226.

o In the Scheduler step, make sure that the username and password are included in the scheduler setup.

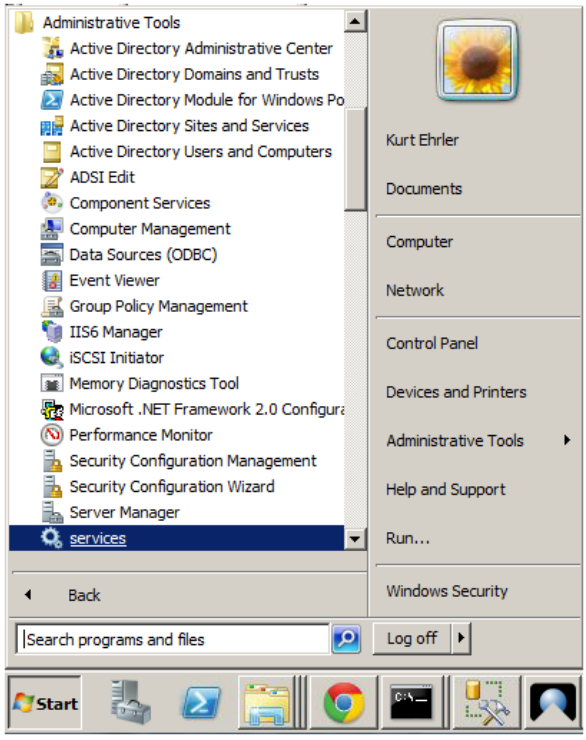
p Manually install the QAD BI modules. See “Manually Installing the QAD BI Analytical Modules” on page 226.

6 If the database server’s OS and database login are the same, set up the login information in an additional location relative to WhereScape, aside from the scheduler. To do this, configure the app box name using Windows Services on Server B:

a Go to Start|All Programs|Administrative Tools|Services.

b Scroll down to WhereScape in the Services window. Right-click it and select Properties.

Fig. D.13
WhereScape Services Window



c Click the Log On tab and select the This Account radio button.

d Enter the domain and username for the OS you will log in to as well as the Password and Confirm password.

7 In DWD, verify that the Data Warehouse connection has the proper login credentials.

8 Once everything is set up, load a table from the Browser pane, by scheduling the load. Also load a table manually. For example, load the load_ad_mstr f table. Verify that you can load the table both manually and from a scheduled load.

Note On a split server system, some configurations allow you to load tables from the scheduler but not manually. If everything is configured properly, you should be able to load tables manually and from a scheduled load.

Troubleshooting the Split Server System

Troubleshooting

Problem	Solution
Scheduled load table job fails	If the scheduled load table job fails, check the messages that explain why it failed. One possible reason for the failure is that there are no Bulk Admin permissions for the user setup in the database on Server A.
Scheduled job never starts	If the scheduled job never starts, it is likely that there is a password failure. Check the scheduler log file to see if it is throwing an invalid username/password error. The scheduler definition as defined in the Admin tool sets the location of the scheduler log.
The audit log generates the following message: “system can not find the path specified message.”	If the database user on Server A does not have read/write permissions to the directory on Server B as defined for the Work directory in the DWD under Connections for the Progress source database to load, a “system can not find the path specified message” is generated in the audit log.
Login failure	An exclamation point (!), and possibly other special characters, in a password will cause a login failure. The failure occurs because the exclamation point is stripped from the password when the bulk insert command is created. If everything appears to be set up correctly and the load is failing, check to see if the password uses special characters. Request an alternate password if this appears to be the problem.
On initial setup, the first attempts to run a scheduler load fail with messages such as sqlcmd not available.	Restart that scheduler and check if it fails again.

Loading Outside Flat Files into the Data Warehouse

If outside source flat files will also be loaded into the data warehouse, their load script requires extra configuration. See the underlined portion below as an example:

```
REM ************************************************************
REM ************* L O A D    T H E   D A T A *****************
REM ************************************************************
echo load of %LOAD_TABLE% > %FILEAUD%
SET /A RESULT_CODE=1
SET RESULT_MSG=Load Completed
:LABEL_LOAD
SET FILE_NAME=NO_MORE_FILES
for %%A in (%LOAD_FILE%) do if "!FILE_NAME!"=="NO_MORE_FILES" SET FILE_NAME=%%A
if "!FILE_NAME!"=="NO_MORE_FILES" goto LABEL_EXIT
echo loading %FILE_NAME%  >> %FILEAUD%
echo --  > %FILECTL%
echo   BULK INSERT  >> %FILECTL%
echo     %LOAD_TABLE% >> %FILECTL%
echo   FROM  >> %FILECTL%
echo     %FILE_NAME% >> %FILECTL%
echo   WITH >> %FILECTL%
echo   ( >> %FILECTL%
```

```
echo        DATAFILETYPE='char' >> %FILECTL%
echo     , CODEPAGE='raw' >> %FILECTL%
echo     , FIELDTERMINATOR = '^|' >> %FILECTL%
echo   ) >> %FILECTL%
echo exit(SELECT @@ERROR) >> %FILECTL%
rem
rem In the following sql statement use either -E for a trusted connection
rem or -Uxxx -Pyyy  where xxx is the username and ppp the password (use if
rem database and Wherescape are on two different machines)
rem
sqlcmd -S %SERVER% -U<username> -P<password> -d %DATABASE% -i %FILECTL% -o
%FILELOG%
SET ERRLEV=%errorlevel%
IF %errorlevel% EQU 0 GOTO LABEL_OKAY
:LABEL_FAIL
echo -2
echo Load failed with return status %ERRLEV% See error log for details
type %FILEAUD% >&2
type %FILELOG% >&2
exit
:LABEL_OKAY
```

Manually Installing the QAD BI Analytical Modules

This section covers the manual installation and configuration for the following QAD Business Intelligence analytical modules:

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

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 being installed, plus the Common set of application files.

Installation Steps

Follow these steps to manually install the QAD Business Intelligence metadata modules.

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

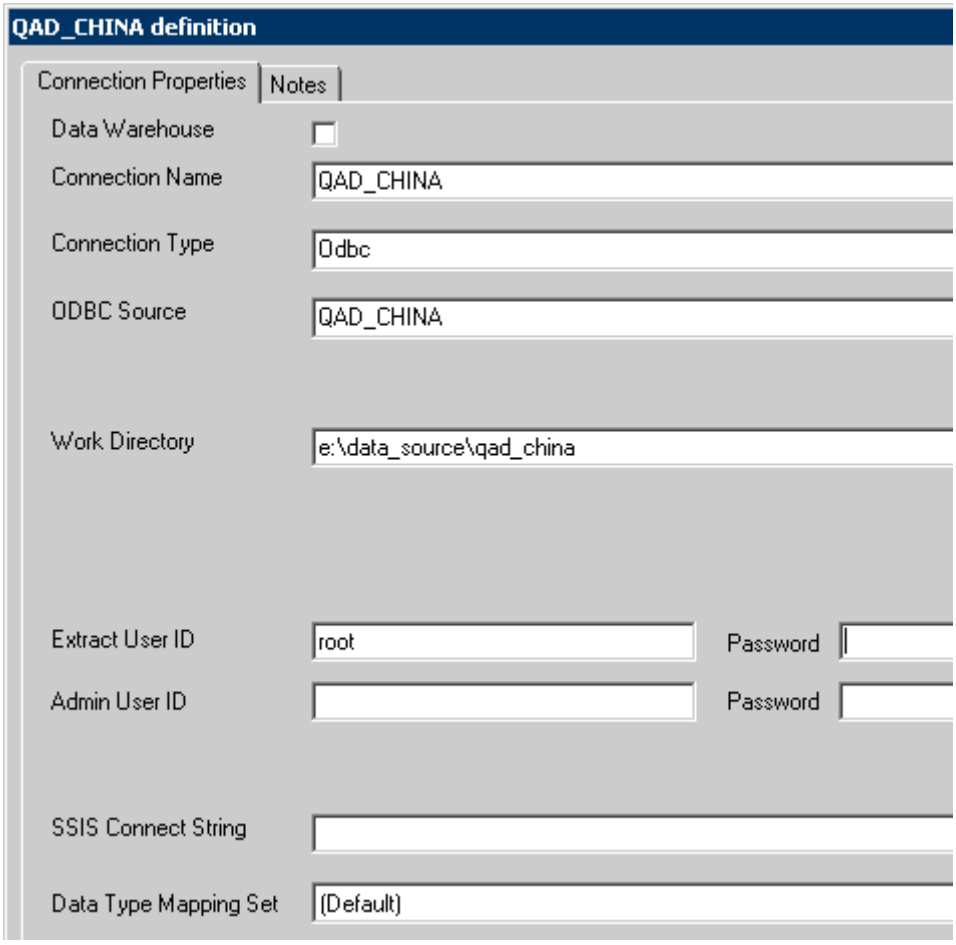
1. Right-click Connection in the QAD BI DWD Metadata Browser and choose New Object.
2. Enter a short descriptive name for the data source; for example, QAD_MAIN, QAD_US, QAD_EMEA, or QAD_ASIA. Click OK.
3. Select a Connection Type of ODBC and select the ODBC connection.
4. Enter a work directory. This directory must be different for each QAD Data Source. Create this directory in the file system. You 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. D.14
Object Connection Definition



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 Install the Common module before installing any of the analytical modules except TMS. The Common module contains core components that all modules use, 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 must 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. There is a SQL script named `misc.sql` in this directory. In the SQL Server Management Studio, open `misc.sql`, change the database to your Data Warehouse, and execute the `misc.sql` script. This step adds a 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 You can see these objects from the DWD under Tools|Options|Documentation.

Install BI Common Module

5 If you are installing any of these analytical modules, the Common module is required:

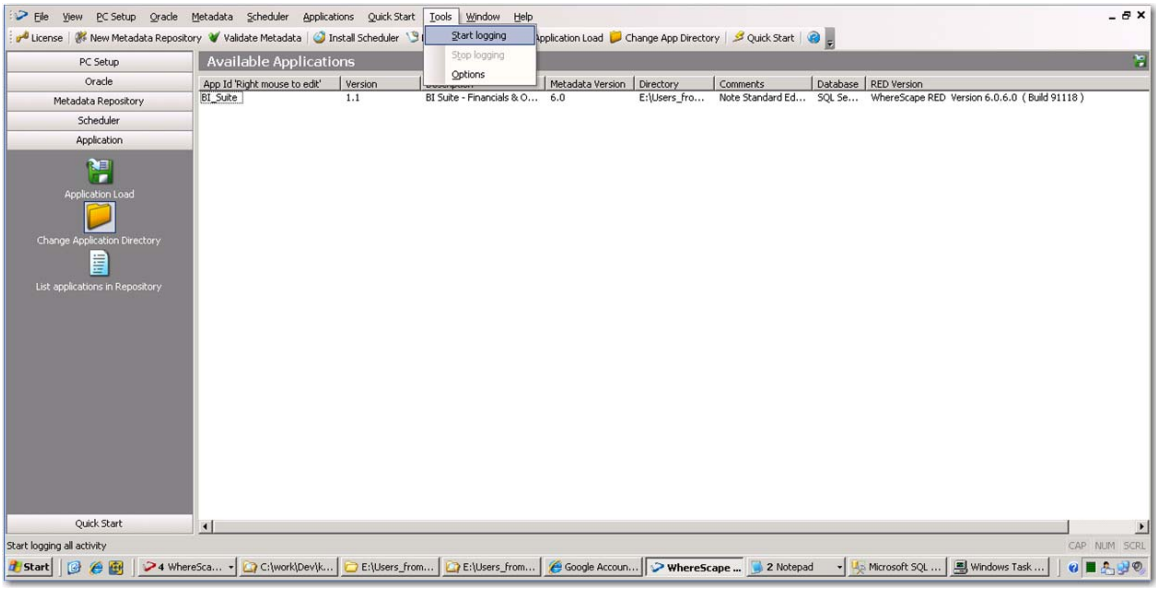
- Order Management (OM)
- Operations (OP)
- Financials (both SE and EE) (FIN)
- EAM

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.

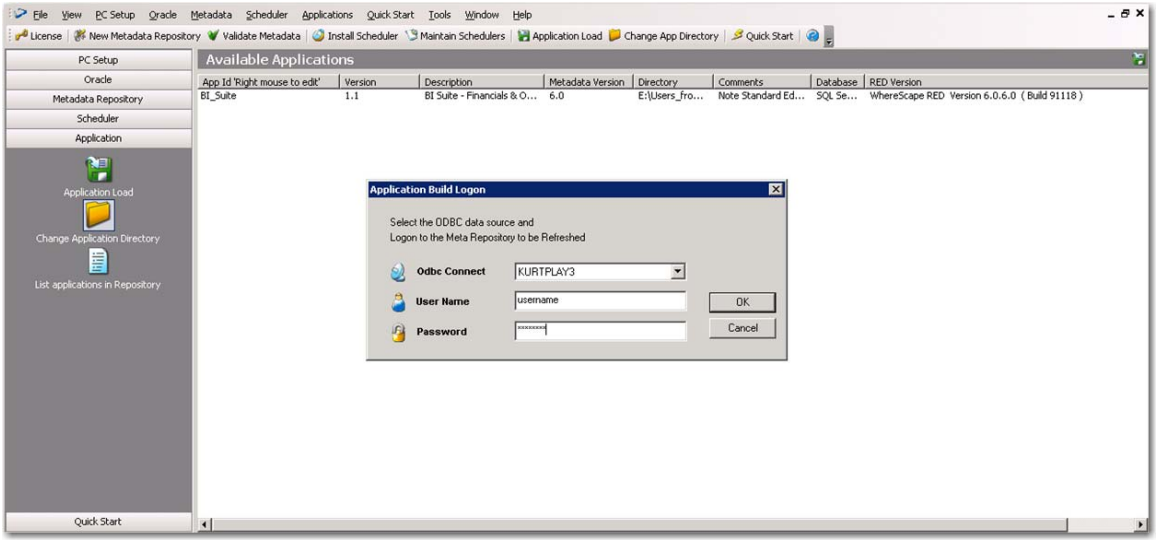
Fig. D.15
Start Logging



8 Choose a file name for your log and a directory for it to write to.

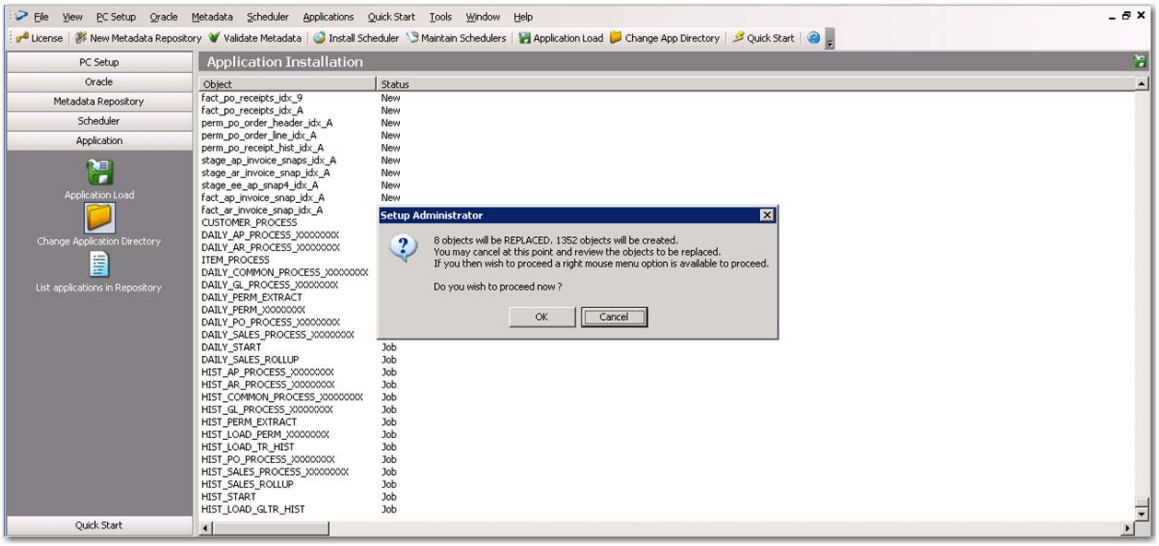
9 Install the application through Application Load. Right-click on the Common module and choose Install.

Fig. D.16
Application Load



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

Fig. D.17
Confirm File Overwrite

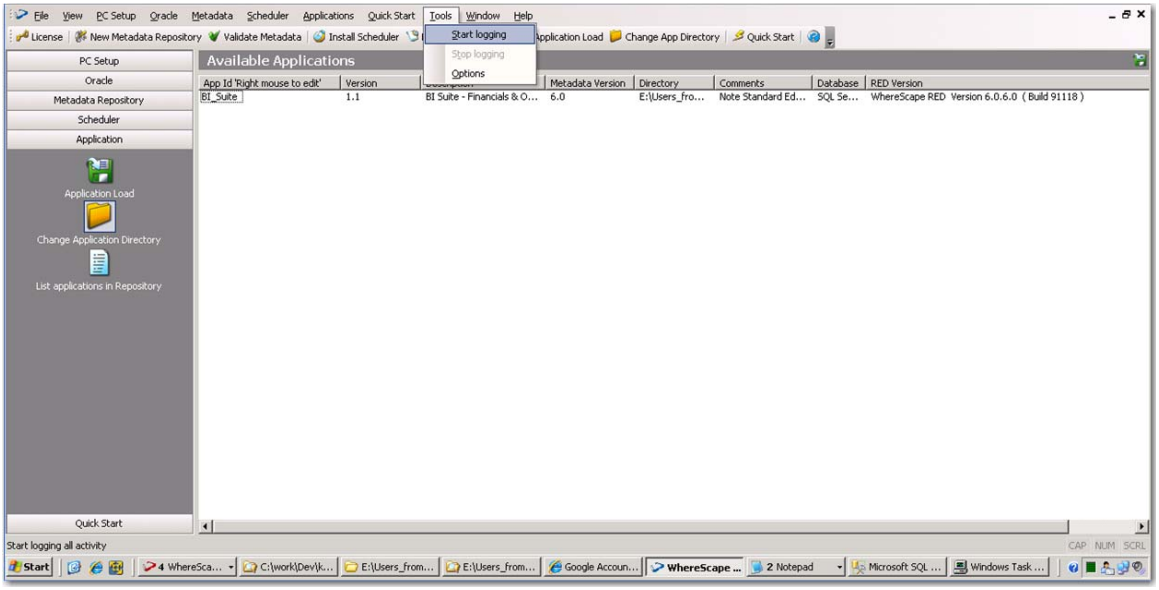


Install BI Analytical Module

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.

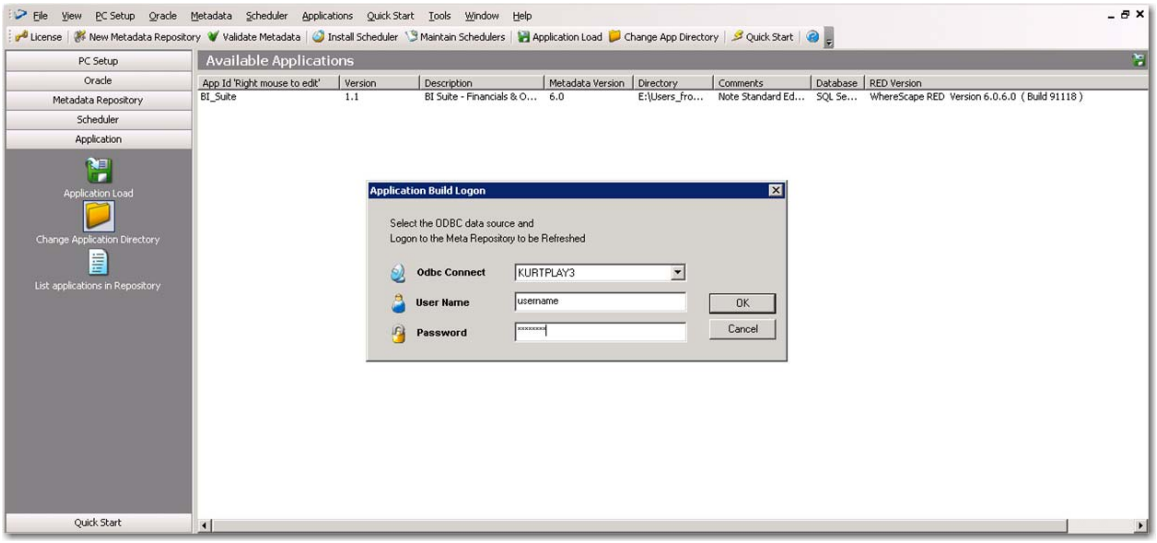
Fig. D.18
Start Logging



13 Choose a file name for your log and a directory for it to write to.

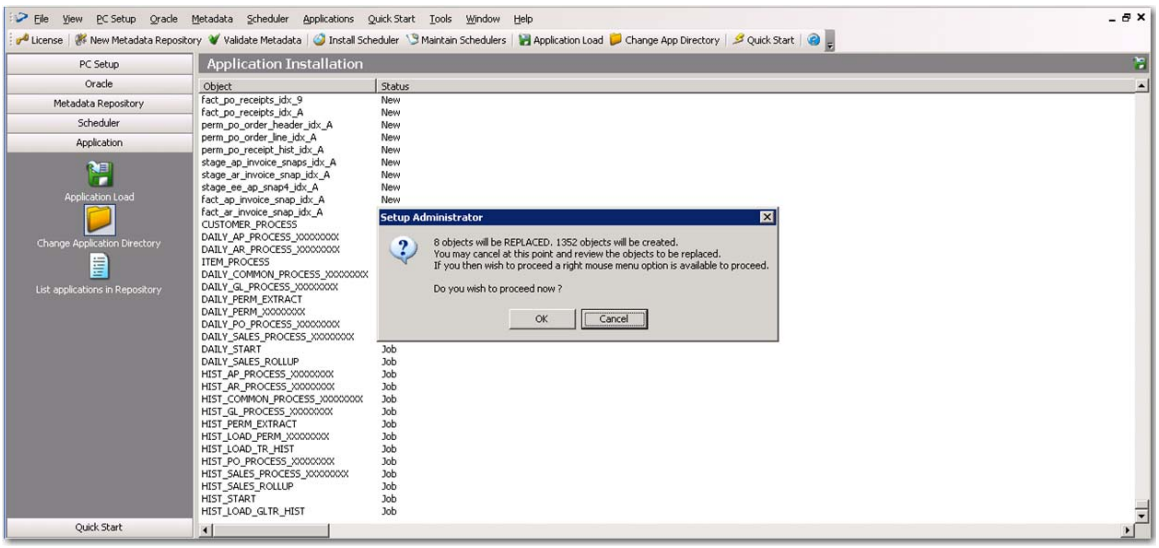
14 Install the application through Application Load. Right-click on the desired module and choose Install.

Fig. D.19
Application Load



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

Fig. D.20
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 can 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 longer, close the screen and start over.

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

Manually Installing the QAD BI Web Application

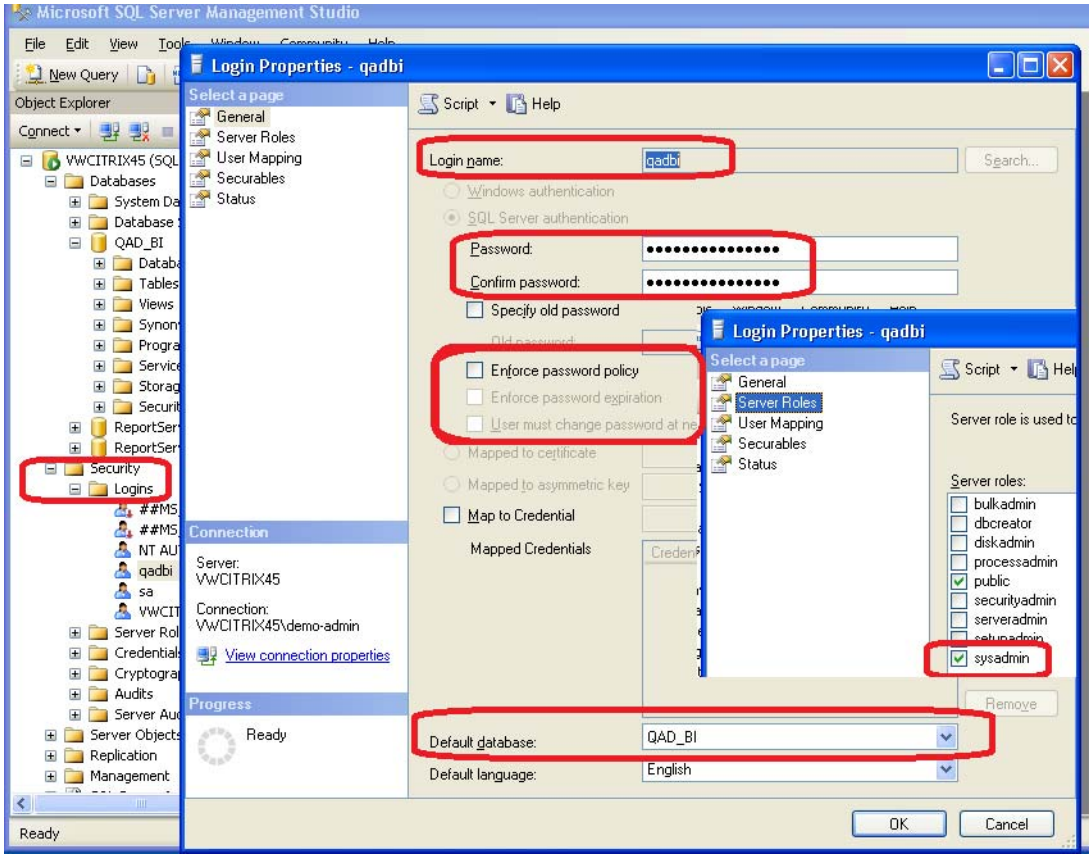
The QAD BI Portal cannot be installed in the same Tomcat instance as QXtend. QXtend requires the use of a specific DOM parser library that is incompatible with the BI application usage. Install the two applications in separate Tomcat instances.

Installation Steps

Follow these steps to manually 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 side of the window) and make sure that the Server roles public and sysadmin are both selected.

Fig. D.21
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_portal.zip` into the Tomcat webapps subdirectory. 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 the Tomcat `server.xml` file, located in Tomcat's `/conf` directory. 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 configuration parameter values for your environment.

The contents are as follows:

- repository Element:
  - **dbType**: Database platform. The value must be SQL for SQL Server.
  - **driver:** JDBC driver to load at runtime for business intelligence server database. For SQL Server, the value is com.microsoft.sqlserver.jdbc.SQLServerDriver.
  - **jdbcUrl**: The JDBC URL to connect to Business Intelligence Portal database.
  - **user**: User name for the database. The BI Portal requires a SQL Server login; it cannot use a Windows login to authenticate itself.
  - **pwd**: Password for the database.
  - **logCheckInterval**: How often, in minutes, the portal performs its maintenance tasks; for instance, log the system state, clean up logs, and so on. The default is 15 minutes.
  - **logDaysToKeepEvents**: Number of days to store system events in the database; aw_evt_system and aw_evt_request tables. The 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. 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 Use your SQL tool of choice to connect to the SQL Server database where the BI Portal Repository will reside.

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 the required repository tables.

Note This script drops the tables first if they exist and re-creates them. Make sure that you are running this script 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 necessary, 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 entry with your specific mail server parameters.
- **USERSERVICE:** Customize the text that is used in the Subject of the Welcome and Password Change e-mails 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. 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.

Getting Started in the BI Portal Web Application

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

19 You must build your model before you can do any analysis. Select Administration|Model Administration and click Rebuild Master.

Product Information Resources

QAD offers a number of online resources to help you get more information about using QAD products.

QAD Forums (community.qad.com)
Ask questions and share information with other members of the user community, including QAD experts.

QAD Knowledgebase (knowledgebase.qad.com)*
Search for answers, tips, or solutions related to any QAD product or topic.

QAD Document Library (www.qad.com/documentlibrary)
Get browser-based access to user guides, release notes, training guides, and so on; use powerful search features to find the document you want, then read online, or download and print PDF.

QAD Learning Center (learning.qad.com)*
Visit QAD's one-stop destination for all courses and training materials.

*Log-in required