



QAD Enterprise Applications

Training Guide
Demand Management 6.1
Technical Training

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

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

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

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

Copyright ©2012 by QAD Inc.

DemandManagementTechnicalTraining_TG_v6.1/m3w/mdf

QAD Inc.

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

Contents

Change Summary	vii
Course Description	2
Course Objectives	2
Audience	2
Prerequisites	2
Course Credit and Scheduling	2
System Requirements	2
QAD Resources	3
Product Help	3
QAD Web Resources	3
Chapter 1 Hardware and Software Requirements	5
DME Application Overview	6
Additional Applications	7
Typical System Architecture	8
User Workstations	10
DME Hardware Requirements	11
DME Software Requirements	12
DME Database Requirements	13
Chapter 2 Install SQL Server 2008	15
Preparing for the SQL Installation	16
Launching the Application	17
Setup Support Rules	18
SQL License Key	19
Setup Support Rules	20
Setup Role	21
Feature Selection	22
Instance Configuration	23
Server Configuration	24
Database Engine Configuration	25
Analysis Services Configuration	27
Reporting Services Configuration	28
Ready to Install	29

Successful Installation	30
Chapter 3 Installing DME & DME Batch Scheduler	31
Preparing for the DME Installation	32
Launching the Installation	33
Installation of Prerequisites	34
Beginning the Installation	35
License Key Installation	36
Setup Type	39
Installation Type	40
Supporting Disconnect Mode	41
Database Administration Tool	42
Internationalization Configuration Wizard	43
Installation Success	45
Preparing for the Batch Scheduler Installation	46
Launching the Installation	47
Destination Folder	48
Setup Type	49
Database Connection	50
Installation Success	51
Installation Validation	52
Chapter 4 Installing Planning Portal	53
Preparing for the Portal Installation	54
Setup Type	55
Installation of Prerequisites	56
Installation Address	57
Database Administration Tool	58
Review of Database Upgrade	59
Validation of Installation	60
Potential Error Message	61
Accessing the Portal Admin Tool	63
Successful Installation	64
Chapter 5 DME Database Optimization	65
SQL Server Memory Settings	66
DME Database Recovery Model	67
Tempdb Database	68
DME Ideal Disk Configuration	69
File Defragmentation	70
Re-indexing the DME Database	71

Chapter 6	DME Database Maintenance	73
	DME Database Maintenance	74
	DME Database Recovery Model	75
	Creating a Database Maintenance Plan	77
	Maintenance Plan Properties	78
	Job Schedule Properties	79
	Selecting Maintenance Tasks	80
	Selecting Maintenance Task Order	81
	Backup Database Task	82
	Check Integrity Task	83
	Shrink Database Task	84
	Update Statistics Task	85
	Maintenance Cleanup Task	86
	Report Options	87
	Maintenance Plan Review	88
	Creation of the Maintenance Plan	89
Chapter 7	DME Database Basics	91
	DME Database Overview	92
	DME Database Key Tables	93
	DME Database Important Notes	96
Chapter 8	Pivot Table View Maintenance	97
	Pivot Table Overview	98
	Default View Definition (vw_PVT_View_V40)	99
	Types of Data to Display in a Pivot Table	100
	Modifying a SQL View	101
	Pivot Table SQL View Notes	103
	Additional SQL View Notes	104
Chapter 9	Introduction to Integration with QAD	105
	DME-QAD Integration Overview	106
	DME-QAD Integration Details	110
	Housing Data Extracted from QAD	114
Chapter 10	QAD-to-DM Integration Preparation	115
	Integration Overview	116
	Preparation Overview	117
	Data to be Extracted	118
	Connecting the DM Server to QAD	119
	Creating a System DSN	120
	Creating a System DSN	121

Creating a Staging Database	122
Creating a Staging Database	123
Chapter 11 QAD-to-DM: Creating an SSIS Package	125
What is an SSIS Package?	126
Creating an SSIS Package	127
Creating an SSIS Package	128
Connection Managers	129
Connection to QAD	130
Connection to QADDM_Staging	132
Connection Manager	134
The Package Screen Layout	135
Activating the Toolbox	136
About the Toolbox	137
Adding a Task to the Control Flow	138
Next Steps	139
Chapter 12 QAD-to-DM: Creating a Data Flow Task	141
What is a Data Flow Task?	142
Adding a Task to the Control Flow	143
Building a Data Flow Task	144
Defining a Data Flow Source	149
Defining a Data Flow Source (ADO.NET)	150
Defining a Data Flow Destination	152
Adding a Table to QADDM_Staging	153
Defining a Data Flow Destination (OLE DB)	155
Completion of the Data Flow Task	157
Creating a Data Flow Task Review	159
Chapter 13 QAD-to-DM: Creating an Execute SQL Task.....	161
What is an Execute SQL Task?	162
Adding a Task to the Control Flow	163
Building an Execute SQL Task	164
Completion of the Execute SQL Task	167
Creating an Execute SQL Task Review	169
Chapter 14 QAD-to-DM: Building a Complete SSIS Package	171
Building a Complete SSIS Package	172
Connecting Multiple Tasks Together	173
Testing the SSIS Package	174
Examples of Package Execution	176
Installing the Package into the SQL Instance	178

Executing the SSIS Package from DM	184
Building the Stored Procedure	185
Building a Complete SSIS Package Review	186
Chapter 15 DM-to-QAD Integration Preparation	187
Integration Overview	188
Preparation Overview	189
Data to be Exported	190
Overview - Flow of Data From DM to QAD	191
Time Buckets	192
Connecting the DM Server to QAD	193
Chapter 16 DM-to-QAD: Creating the DM Export Tables	195
DM Export Tables	196
Creating the DM Export Tables	197
Populating QAD_ExportDates	199
Next Steps	201
Chapter 17 DM-to-QAD: Building the DM Export Procedure	203
DM Export Procedure	204
Defining the Export Parameters	205
Creating the Forecast Export Procedure	206
Additional Considerations	211
Next Steps	212
Chapter 18 Developing Custom Tasks	213
Custom Task Overview	214
Stored Procedure Requirements	215
Stored Procedure Parameters	216
Pre-Installed Stored Procedures	217
Creating a new Custom Task	218
Custom Task Example	220

About This Course

Course Description

QAD designed this course to cover the installation and basic technical configuration of the Demand Management (DM) program and its required application components. The course includes hands-on activities and exercises that allow students to practice installation and development activities in DM.

Course Objectives

By the end of this class, students will:

- Install SQL Server 2008
- Install all application components of the Demand Management Engine and Planning Portal
- Understand the DM database structure
- Configure the maintenance and optimization of the DM database
- Modify pivot table views
- Build integration between QAD Enterprise Applications and SQL
- Prepare a DM forecast file for output to QAD Enterprise Applications
- Understand the purpose as well as develop basic custom tasks

Audience

- DM technical support
- Implementation consultants and members of implementation teams responsible for technical activities

Prerequisites

- Demand Management Technical Learning Path
- Basics of SQL Server 2008

Course Credit and Scheduling

QAD designed this course to take 24 hours.

System Requirements

This course uses the QAD Enterprise Applications Enterprise Edition database. See the title page for the exact database title. Whether you are studying this material in a classroom or as a self-study exercise, this is launched in a virtual environment management system. QAD instructors will prepare classroom environments. If you encounter questions or problems on QAD software that are not addressed in this book, several resources are available.

QAD Resources

If you encounter questions or problems on QAD software that are not addressed in this book, several resources are available.

Product Help

All QAD products ship with integrated help systems. A properly installed QAD application will display help when you press the Help key (F1), or access it through the menu. The help covers the normal use of the product.

QAD Web Resources

The QAD website provides product and company overviews. The Print Solution option on the opening page provides a means of compiling desired content into a document specialized to your industry, business implementation, and needs.

<http://www.qad.com/>

From QAD's main site, you can access QAD's Learning or Support sites.

QAD Learning Portal for Training Opportunities

To view available training courses, locations, and materials, use the QAD Learning Portal. Choose Learning under the Global Services tab to access this resource.

QAD Support for Product Documentation and the QAD Knowledgebase

To access release notes, user guides, installation and conversion guides by product and release, visit the Support website. Support also offers an array of tools depending on your company's maintenance agreement with QAD. These include the Knowledgebase and direct links to QAD Support experts.

Choose Support under the Global Services tab.

Any QAD customer can register for a QAD web account by accessing the Support web site and clicking the Accounts link at the top of the screen. Your customer ID number is required. Access to certain areas is dependent on the type of agreement you have with QAD.

Chapter 1

Hardware and Software Requirements

DME Application Overview

DME Application Overview

- The Demand Management Engine (DME) consists of four applications plus a database
 - **DME Admin Tool** – Scenario configuration settings such as data loading and forecast methods
 - **DME Viewer** – Primary user-interface of the DME
 - **Batch Scheduler** – Allows for the automated execution of repetitive tasks as part of a batch process
 - **DME Connection** – Tells the DME application how to connect to the DME database
 - **DME Database** – the SQL Server database where all DME data is stored



DM-H&S-Req-020

Note The installation of the DME application allows you to install both the application and/or the database. The application that is installed on the “application server” is the same application that is installed on a “client machine”.

Additional Applications

Additional Applications

- In addition to the DME, the following add-on applications are also available
 - **Planning Portal** – A web browser-based tool to allow casual users to provide input to the forecasting process
 - **Inventory Optimization*** – A distribution requirements planning (DRP) and safety stock calculation tool
 - **Rough-Cut Capacity Planning*** – A tool to calculate the load against constrained resources based on the forecast or distribution plan

** Not covered within these training materials*

Typical System Architecture

Typical System Architecture

- Servers can be physical or virtual
- Deployment of the DME application (including the Planning Portal) involves three separate components
 - **Application** – the DME application
 - **Database** – the DME database
 - **Web Server** – the ability to provide users with access to the Planning Portal application

Typical System Architecture

- Deployment of the required components can be achieved through one of three scenarios
 - **One server** – the application, database and web server all reside on the same server (most common)
 - **Two servers** – the application and web server reside on one server and the database resides on another server
 - **Three servers** – the application, database and web server all reside on separate servers



DM-H&S-Req-050

Most companies deploy all three components on one server. In essence, this is a “dedicated DM server”. However, it is common that companies will have dedicated SQL Servers (servers where they house all of their SQL databases). In this instance, it is common for the DME database to reside on the SQL Server and the application/web components reside on another server.

Note If the database resides on a separate server from the application, there are some application components that must be installed on the database server (some customers do not like this). See installation guide for details.

User Workstations

User Workstations

- Users can access the DME application through one of several methods
 - **User workstation** – the application is installed on the user's workstation and the application connects to the database server (requires connection to the network)
 - **Remote desktop** – the user connects to the application server via a remote connectivity tool such as remote desktop (Terminal Services)
 - **Citrix or similar tool** – the application resides on a server that is accessible by multiple users



DM-H&S-Req-000

Smaller companies tend to install the software on a user's workstation as this is the lowest-cost and simplest approach. Larger companies (ones with a large number of DME users) tend to gravitate towards a remote accessibility model (Citrix or Terminal Services) as this minimizes the number of client installations required (typically one). The customer must consider any licensing costs associated with the remote accessibility options such as Citrix or Terminal Services.

DME Hardware Requirements

DME Hardware Requirements

- The DME is typically installed on an **application server**
 - The database can also reside on the same server
 - The application can also be installed on user workstations
 - The hardware required to support the application should meet these minimums
 - **CPU**–Pentium “Core” series – 2.0 GHz
 - **RAM**– 2 GB
 - **Hard Disk** – 200 GB (Application server)
 - **Operating System** – Windows Server 2003 or 2008 (Application server); Windows XP/Vista/7 (User workstations)



DME-H&S-Req-070

While it is not required, in customer implementations the software is always installed on the application server. The software can also be installed on one more user workstations, if desired.

DME Software Requirements

DME Software Requirements

- When installing the DME software on the application server, the following software is required
 - **Microsoft .NET Framework 3.5 SP1**
 - **MSXML 6.0 SP1**
 - **Microsoft SQL Server Native Client**
 - **Microsoft SQL Server Command Line Query**
 - **Microsoft SQL Server 2005 Backward Compatibility Components**
 - **Microsoft SQL Server 2005/2008 Management Objects Collection**
 - **Microsoft Exception Message Box**
 - **Microsoft SQL Server Integration Services**



DM-H&S-Req-080

The DME and/or Batch Scheduler installation program will check for the presence of these software requirements. If these components are not present, the installation program will install these components.

DME Database Requirements

DME Database Requirements

- The DME application utilizes a SQL Server database
 - The following software must be installed in order to support a typical QAD implementation
 - **Microsoft SQL Server 2005 or 2008**
 - Standard Edition or higher
 - **Progress Client (to support ODBC connectivity)**



DME-H&S-Req-090

The above refers to the software required on the database server (or other server depending on the architecture chosen). The requirement for the Progress Client is based on the assumption that the customer will be retrieving data from QAD Enterprise Applications for use by Demand Management.

Chapter 2

Install SQL Server 2008

Preparing for the SQL Installation

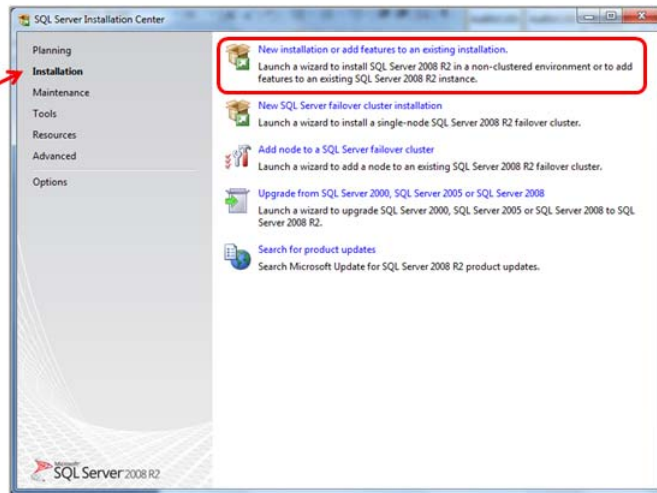
Preparing for the SQL Installation

- Download the SQL software media to an accessible drive on the server or place the media in a drive accessible by the server
- In conjunction with the customer IT team, determine the appropriate install locations for:
 - Application files
 - Database files

Launching the Application

Launching the Installation

- When the SQL Server Installation Center window appears click the **Installation** section and launch the “New installation” option

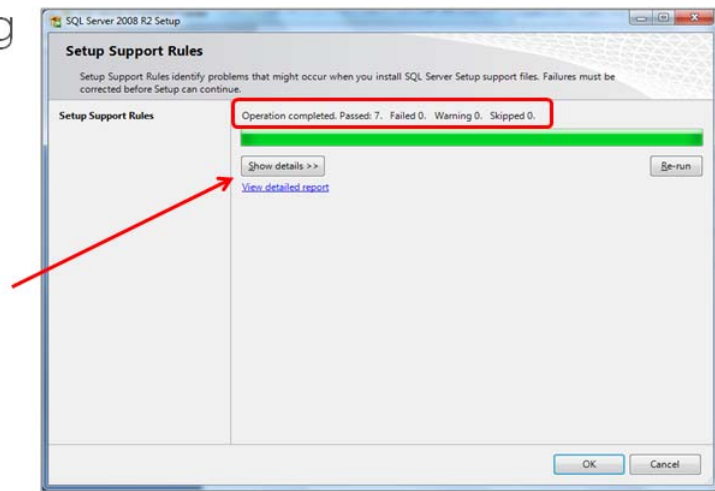


The “New installation or add features to an existing installation” will be the most commonly used option in the SQL Server Installation Center. This is also the best option when installing SQL Server on company equipment for demonstration purposes.

Setup Support Rules

Setup Support Rules

- The install program will determine if any problems might occur during the installation of support files, review any issues prior to proceeding



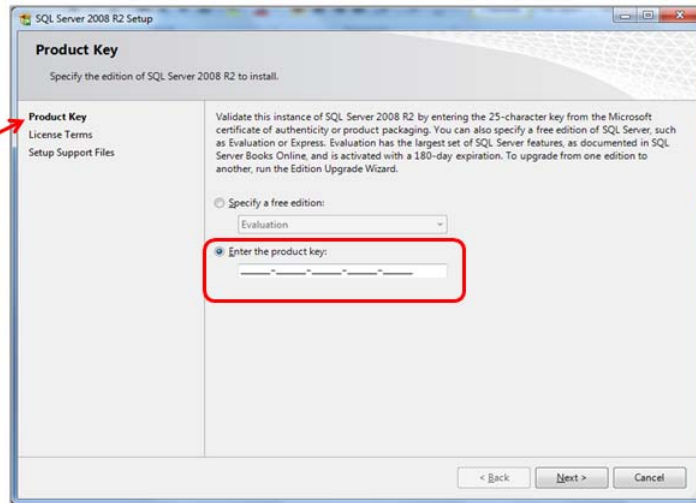
D:\N-Install-SQL-040

The “Show details” function will provide a detailed listing of the Setup Support Rules analysis. It is highly recommended (and in some cases required) that you resolve any issues prior to proceeding.

SQL License Key

SQL License Key

- Obtain the SQL license key from the customer and enter it in the **product key** section

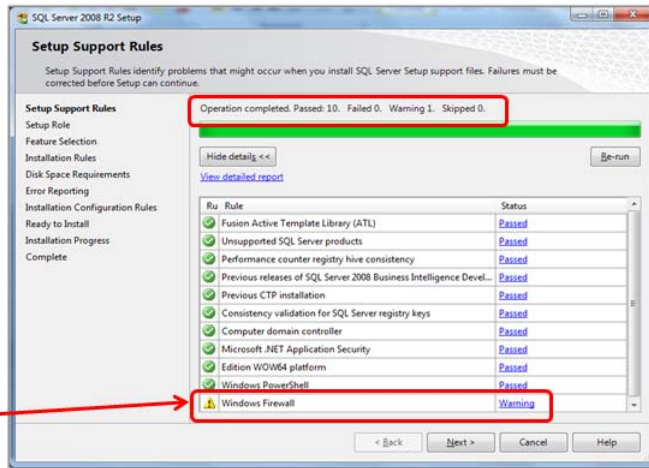


As noted in previous slides, you should be installing SQL Server Standard Edition or higher. It is possible to install an “evaluation” edition as long as a license is entered prior to the deployment of the database into a production environment.

Setup Support Rules

Setup Support Rules

- The install program will determine if any additional problems might occur during the installation, review any issues prior to proceeding

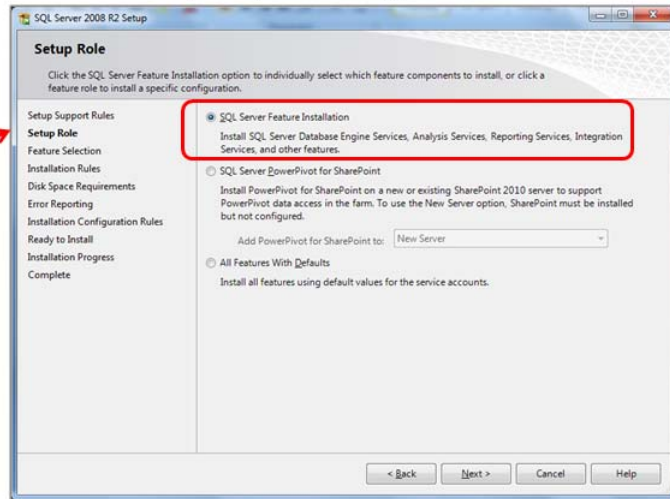


The “Show details” function will provide a detailed listing of the Setup Support Rules analysis. It is highly recommended (and in some cases required) that you resolve any issues prior to proceeding.

Setup Role

Setup Role

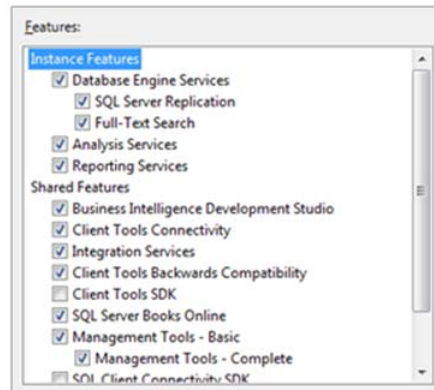
- We will choose the **SQL Server Feature Installation** option, this installs the database engine, analysis services, reporting services and integration services



Feature Selection

Feature Selection

- Within feature selection, the following features **must** be installed
 - Database Engine Services
 - Reporting Services
 - Business Intelligence Development Studio
 - Client Tools Connectivity
 - Integration Services
 - Management Tools – Basic/Complete



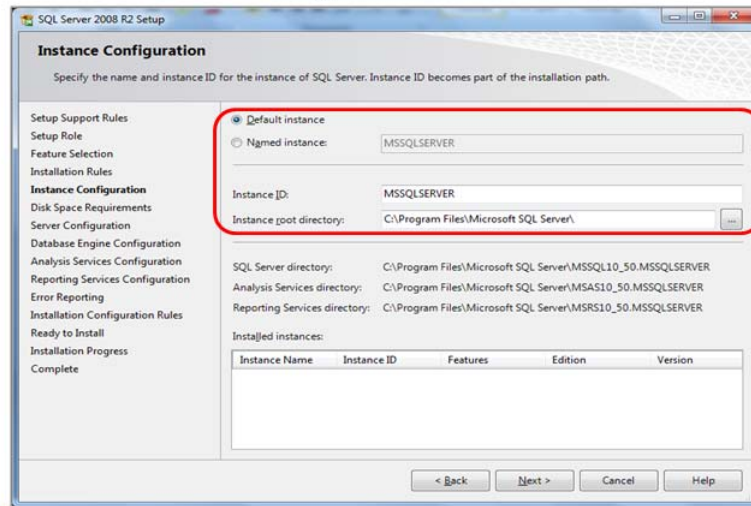
DMA-Install-SQL-080

Your customer can provide you guidance as to any additional features that should be included. Many customers prefer to include the SQL Server Books Online as it is an excellent SQL resource.

Instance Configuration

Instance Configuration

- In the Instance Configuration window, provide a name for the SQL instance (or use the default naming convention)

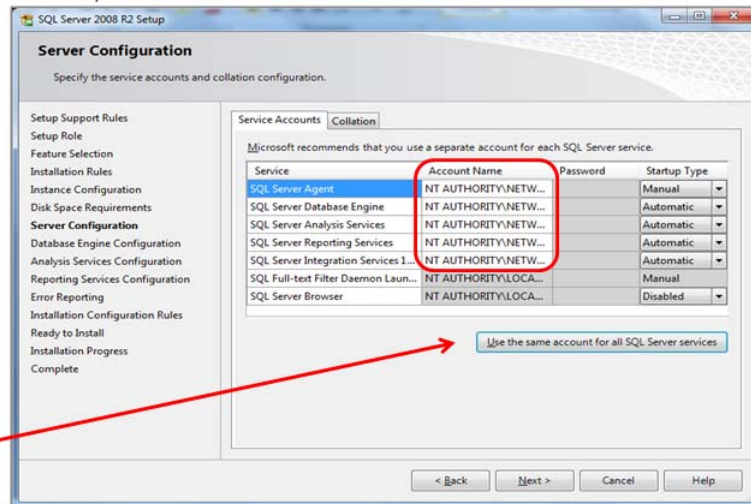


Your customer may provide guidance related to the naming of the SQL instance and its root directory. If you are unsure, the defaults are a good starting point.

Server Configuration

Server Configuration

- The various SQL services require an account name to execute
 - Specify the account name for each service



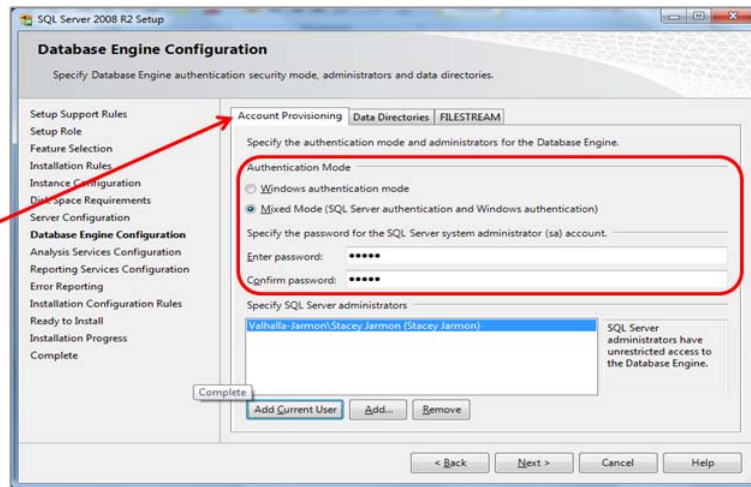
DM-Install-SQL-100

Your customer may provide guidance related to the account names to assign to the various SQL services. A good starting point is to assign either the “LOCAL SERVICE” or “NETWORK SERVICE” account name to the various services (as shown in this example). These settings can be changed later if necessary.

Database Engine Configuration

Database Engine Configuration

- It is critical to choose **Mixed Mode Authentication** in the Account Provisioning tab - also provide the password for the "sa" account

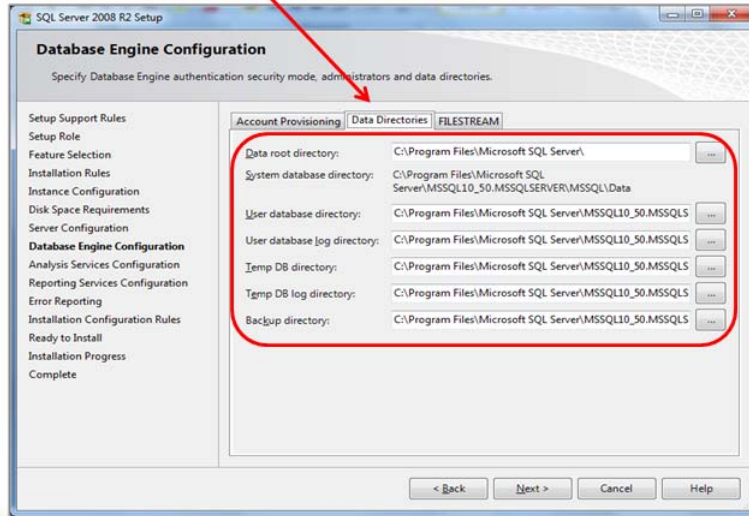


DVA-Install-SQL-110

Mixed Mode Authentication is a requirement of DME as several pieces of the application cannot utilize Windows authentication mode to connect to the SQL database. On this screen, SQL Server administrators can be defined.

Database Engine Configuration

- During this step, the locations of the databases can be defined on the **Data Directories** tab

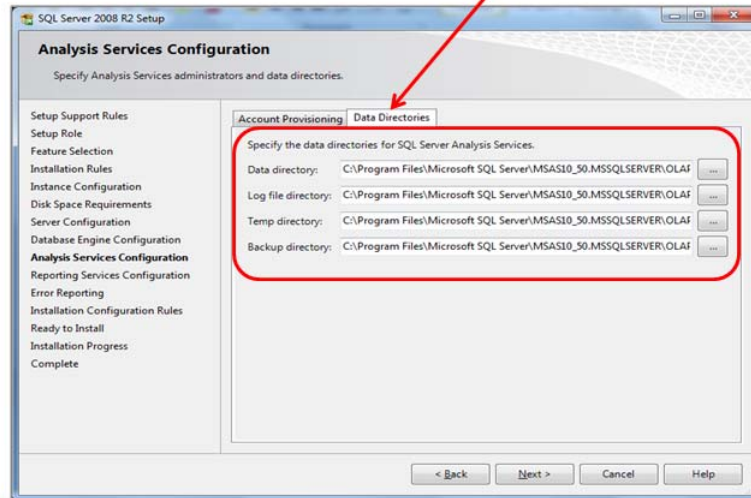


Consult with your customer to determine the appropriate data directory locations. The defaults provide good starting points. You will note during the DME installation, you have the ability to install the DME database in ANY directory of your choosing.

Analysis Services Configuration

Analysis Services Configuration

- During this step, the locations of the data directories for Analysis Services can be defined on the **Data Directories** tab.

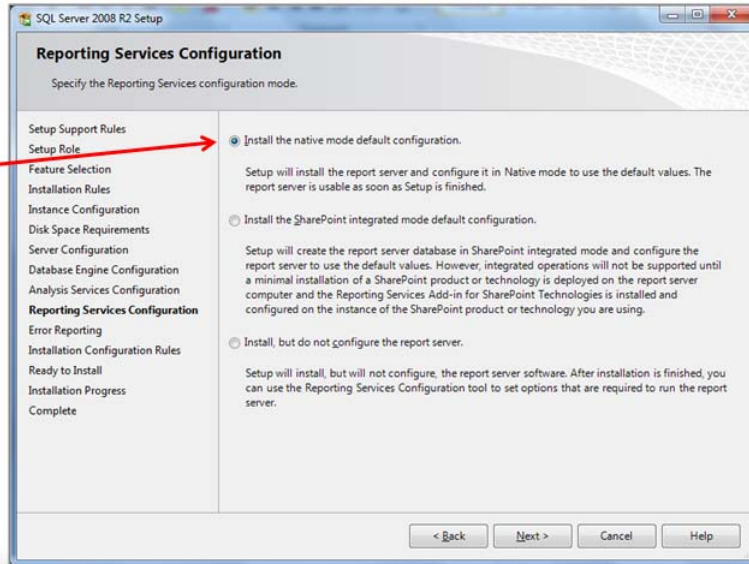


Consult with your customer to determine the appropriate data directory locations. The defaults provide good starting points.

Reporting Services Configuration

Reporting Services Configuration

- Select the "Install the native mode default configuration" option



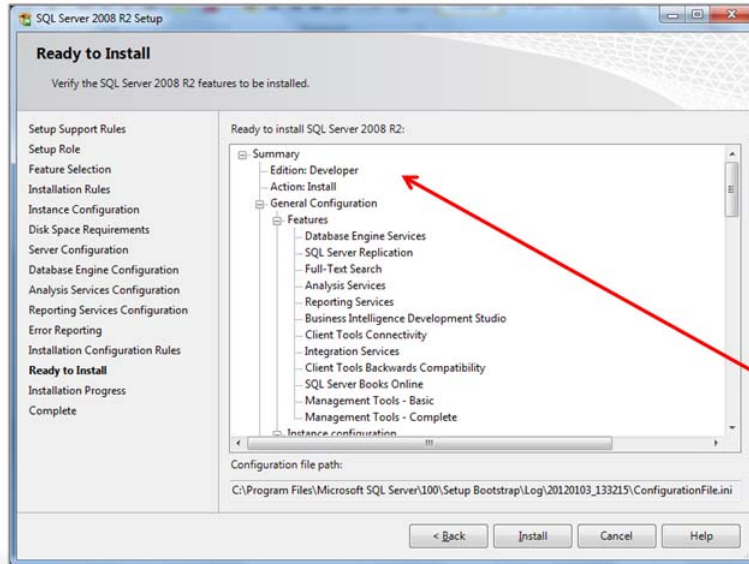
DM-Install-SQL-1-40

Consult with your customer to determine any of the other Reporting Services options are necessary.

Ready to Install

Ready to Install

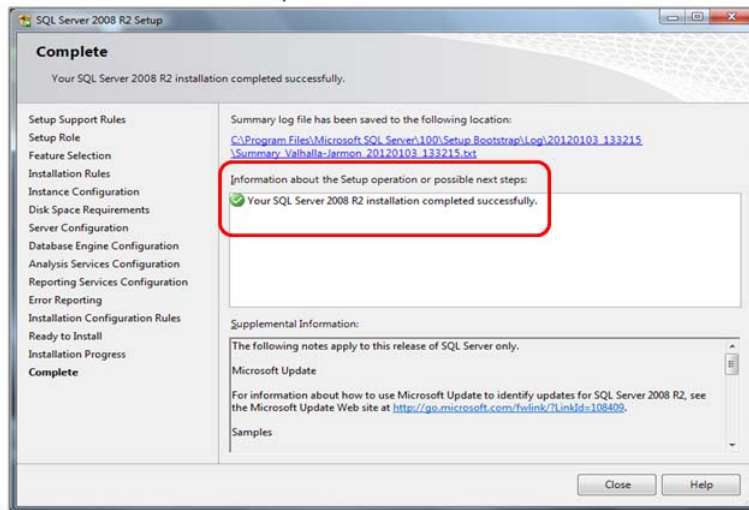
- Prior to beginning the installation, you can review the selections made to this point



Successful Installation

Successful Installation

- If the installation is successful, you will be notified regarding the completion or any required next steps



DM-Install-SQL-1.60

Rebooting of the server is not required after installing SQL Server unless you are prompted to do so.

Chapter 3

Installing DME & DME Batch Scheduler

Preparing for the DME Installation

Preparing for the DME Installation

- Download the DME software media to an accessible drive on the server
- In conjunction with the customer IT team, determine the appropriate install locations for
 - Application files
 - Database files
- Communicate your installation schedule to your QAD Support contact so a license key can be issued to you during the installation

Launching the Installation

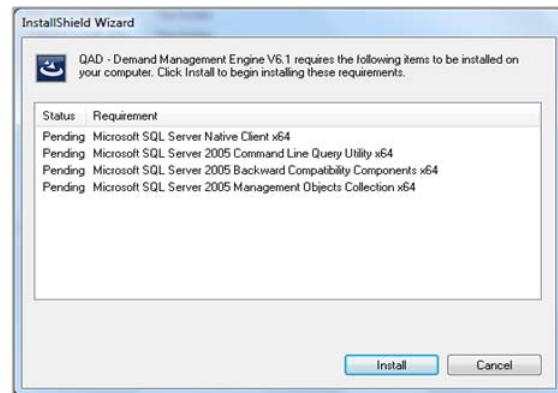
Launching the Installation

- Ensure that you are logged into the server as an administrator
- To begin the installation, double-click the **setup.exe** file contained within the Demand Management 6.1 installation folder

Installation of Prerequisites

Installation of Prerequisites

- Upon launching the DME installation, the installation program will determine if you have all software prerequisites installed
- If not, the software will install these prerequisites for you



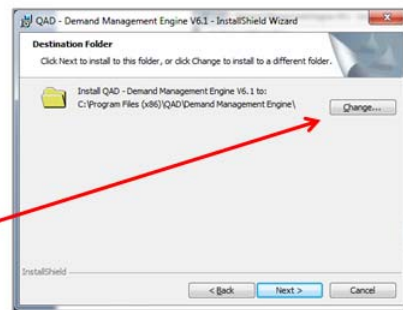
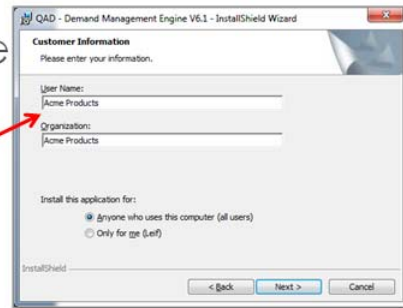
DM-Install-DME-040

Many of the prerequisites related to .NET versions and SQL Server components are already part of the DME installation package. The installation package may also attempt to download files required during this phase.

Beginning the Installation

Beginning the Installation

- Consult the client to define the appropriate name to use in the **Customer Information** section
- Within the **Destination Folder** window, select the appropriate target folder for the DME application



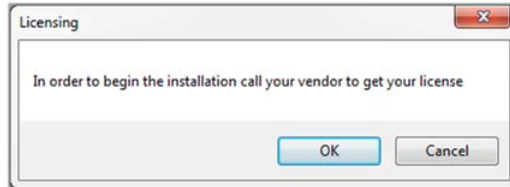
DME-Install-DME-050

Within the destination folder screen, you have the opportunity to define where the DME application will be installed. Please consult with the customer's IT team to determine the optimal location to install the application.

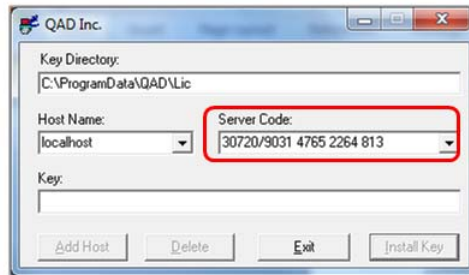
License Key Installation

License Key Installation

- The **Licensing** window alerts you that you need to provide a license key to continue



- Within the **QAD Inc.** window, make a note of the **Server Code**

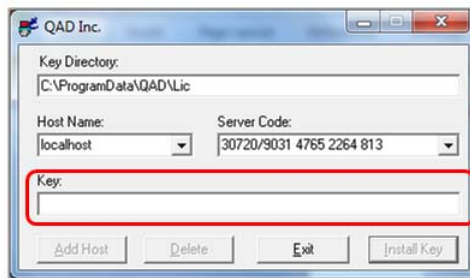


DME-Install-DME-0.60

While a license key is required to install DME, you cannot obtain the license key prior to launching the DME installation. The “Server Code” is a unique code that is generated on each server/workstation by the installation program. The license key that is provided is unique to this “Server Code” and cannot be used on other servers or workstations.

License Key Installation

- Call your **QAD Support Representative** to obtain a license key
 - Provide the "Server Code" displayed within the window
- Your Support Representative will provide you with a license key
 - Enter that number within the **Key** field and click **Install Key**

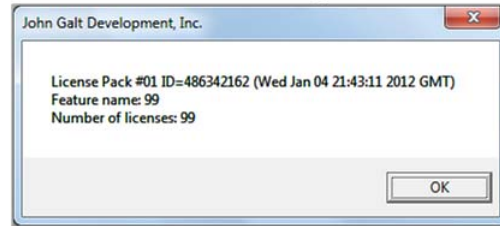


DME-Install-DME-070

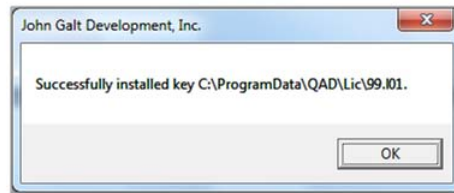
While a license key is required to install DME, you cannot obtain the license key prior to launching the DME installation. The "Server Code" is a unique code that is generated on each server/workstation by the installation program. The license key that is provided is unique to this "Server Code" and cannot be used on other servers or workstations.

License Key Installation

- If your license key installation was successful, you will see the following window



- After clicking **OK**, you will note the specific location of the license key install file



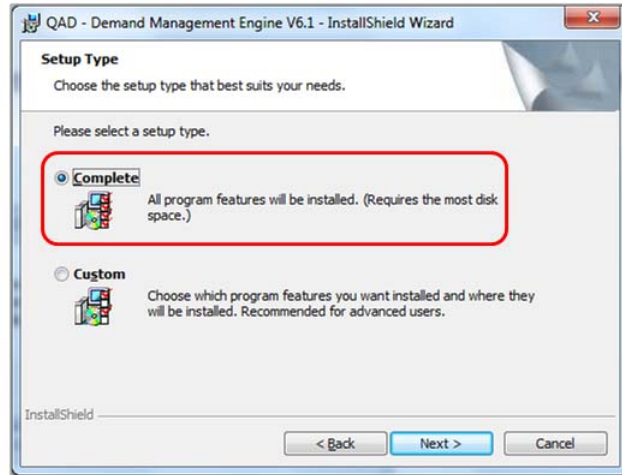
D:\Install-D\ME-080

When you see the “John Galt Development, Inc.” window appear, you might notice that it says “Number of licenses: 99”. This message does not have any relevance in terms of the number of users that a customer is entitled to. Please note that the software does check for the presence of the license key file, so please ensure that the license key file that is created (as denoted in the second picture above) is not deleted for any reason.

Setup Type

Setup Type

- For an initial installation of DME on a server, a **Complete** installation is recommended.
- For special situations, the **Custom** setting can be used to install specific software components



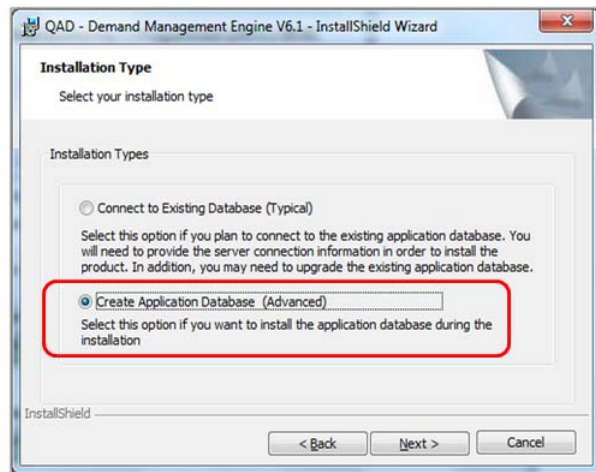
DME-Install-DME-090

When installing DME on the “application server”, a complete installation is recommended. If installing only on the database server, components such as the Viewer can be omitted.

Installation Type

Installation Type

- If you need to create a **new DME database** as part of this installation, choose **Create Application Database**
- If a DME database already exists, select the **Connect to Existing Database** option



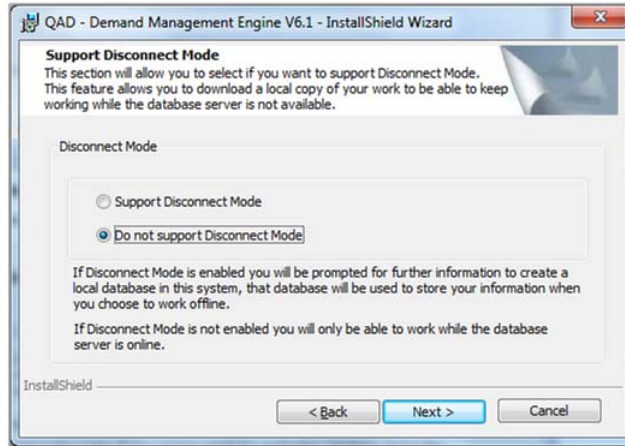
DME-Install-DME100

When installing DME on the “application server”, a complete installation is recommended. If installing only on the database server, components such as the Viewer can be omitted.

Supporting Disconnect Mode

Supporting Disconnect Mode

- Disconnect Mode allows a user to download a local copy of the database to their workstation

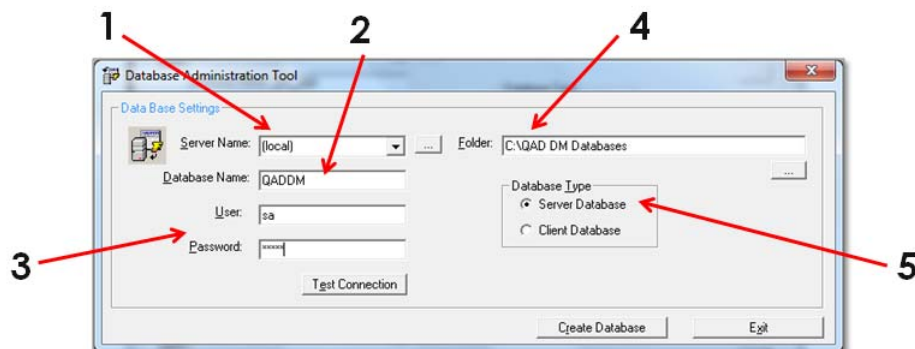


The Disconnect Mode functionality allows a user to download work to a database that resides on their workstation – allowing for “offline work”. This is only recommended when the customer has this defined requirement. Depending on the specific DME configuration in place, use of the Disconnect Mode needs to be documented very carefully to ensure data integrity.

Database Administration Tool

Database Administration Tool

- Within this window, provide the appropriate parameters to allow the installation to connect to the SQL environment and create a new DME database



To create a new DME database, define the following parameters prior to clicking Create Database:

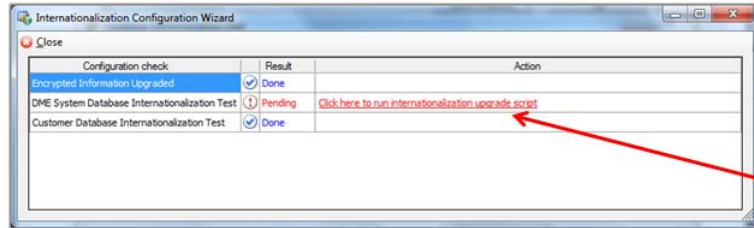
- 1** Enter the server name. If the database is being created on the same server as the DME installation, you can utilize “(local)” as the server name.
- 2** Enter the Database Name. The default value is “FX_JGD”. You may wish to change this to something more intuitive, such as “QADDM_Test”, “QADDM_Prod” or simply “QADDM”.
- 3** Enter the user name and password that should be used to connect to the SQL Server. It should have “sa”-level privileges.
- 4** Specify the folder into which the database should be created. The default is “C:\”. Please change that to a more appropriate destination.
- 5** On a server installation, choose “Server Database”. If you were creating a database for a client’s use of disconnected mode, you would use “Client Database”.

Note You can click “Test Connection” to check the validity of the SQL Server connection using the specified server/user/password.

Internationalization Configuration Wizard

Internationalization Configuration Wizard

- DME 6.1 has the ability to support double-byte data storage and usage
 - This wizard displays whether any installation tasks remain

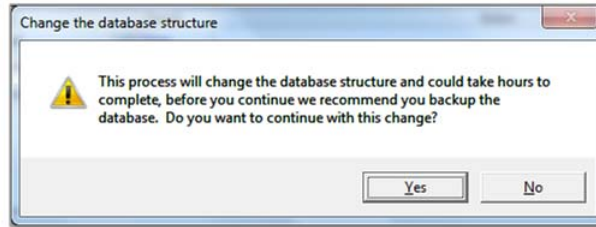


- If tasks remain, simply click the link in the **Action** field

If all of the Configuration Checks in this window show as “Done”, you may simply click “Close”.

Internationalization Configuration Wizard

- Upon executing the **Internationalization Upgrade Script** - you will see this message



- Click **Yes** - this action will only take a few minutes as our database is empty
- Once complete - the database activity will be completed

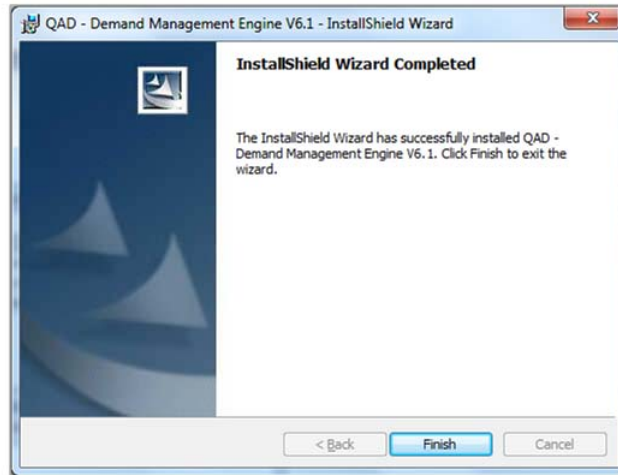


The “Change the database structure” warning message is meant to highlight the extra processing time that may occur if applied to a large database. However, as the database is empty (as part of a new installation), the conversion occurs very quickly.

Installation Success

Installation Success

- If the **InstallShield Wizard Completed** window is displayed you have successfully completed the DME installation



DME-Install-DME-1.50

In some rare cases, it is possible that this screen will appear after an “unsuccessful” installation. Pay attention to any error messages that appeared during the installation process as these may have affected the outcome of the installation process.

Preparing for the Batch Scheduler Installation

Preparing for the Batch Scheduler Installation

- Download the DME software media to an accessible drive on the server
- In conjunction with the customer IT team, determine the appropriate install locations for
 - Application files
 - Database files
- Communicate your installation schedule to your QAD Support contact so a license key can be issued to you during the installation



Dm-Install-DME-BS-020

Launching the Installation

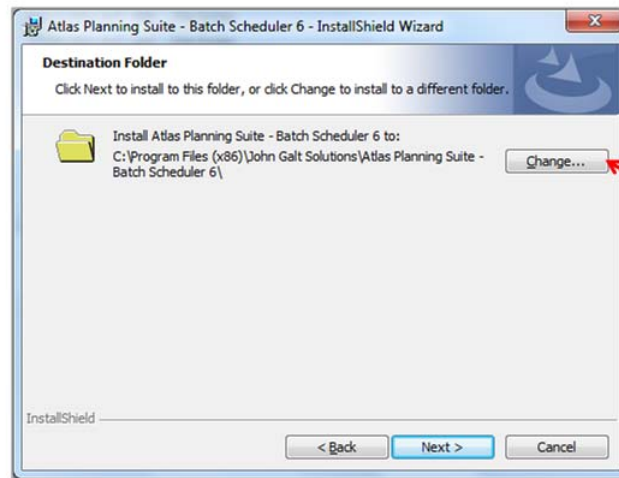
Launching the Installation

- Ensure that you are logged into the server as an administrator
- To begin the installation, double-click the **setup.exe** file contained within the Demand Management Batch Scheduler 6.1 installation folder

Destination Folder

Destination Folder

- If necessary, modify the **destination folder** for the Batch Scheduler application.



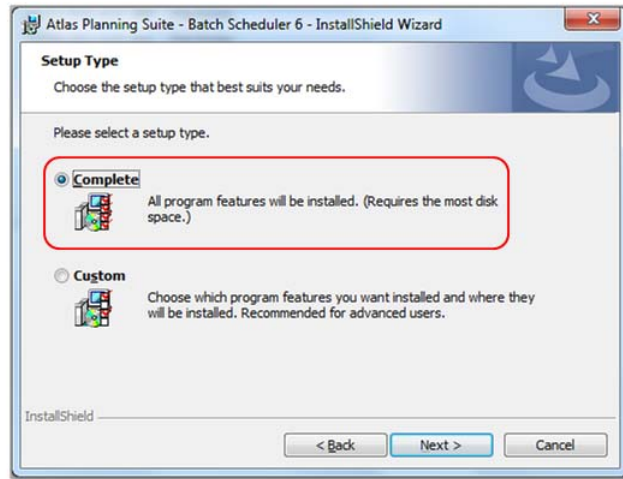
Dm-Install-DME-BS-040

Within the destination folder screen, you have the opportunity to define where the Batch Scheduler application will be installed. Please consult with the customer's IT team to determine the optimal location to install the application.

Setup Type

Setup Type

- For an initial installation of Batch Scheduler on a server, a **Complete** installation is recommended
- For special situations, the **Custom** setting can be used to install specific software components

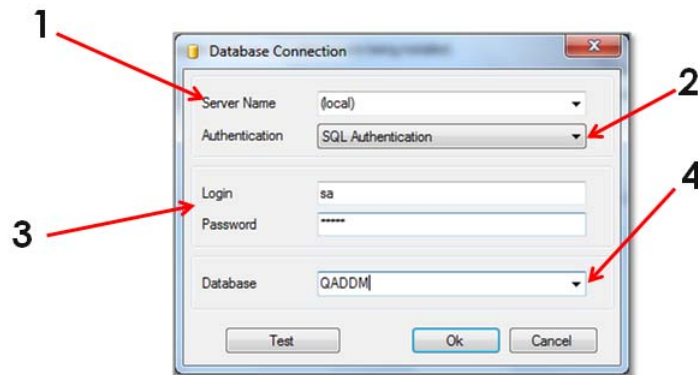


When installing Batch Scheduler on the “application server”, a complete installation is recommended.

Database Connection

Database Connection

- Within this window, provide the appropriate parameters to allow the installation to connect to the SQL environment and update the DME database



Dm-Install-DME-6.1-0.60

To connect to the DME database, define the following parameters prior to clicking OK:

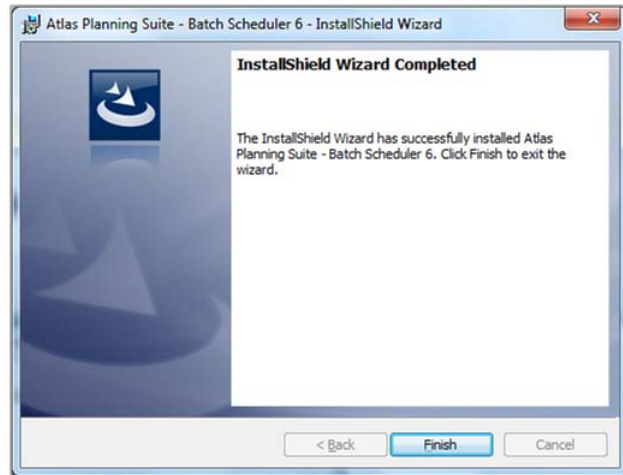
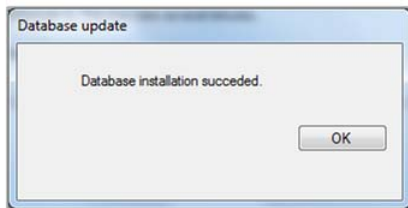
- 1 Enter the server name. If the database is being created on the same server as the DME installation, you can utilize “(local)” as the server name.
- 2 Enter either Windows Authentication or SQL Authentication as the method to connect to the database. At this time, SQL Authentication is easiest as you defined the “sa” role during the SQL installation but not necessarily the security related to use of Windows Authentication.
- 3 If using SQL Authentication, enter the user name and password that should be used to connect to the SQL Server. It should have “sa”-level privileges.
- 4 Enter the Database Name. The drop-down list will provide a list of available databases from the defined SQL Server.

Note You can click “Test” to check the validity of the SQL Server connection using the specified server/user/password.

Installation Success

Installation Success

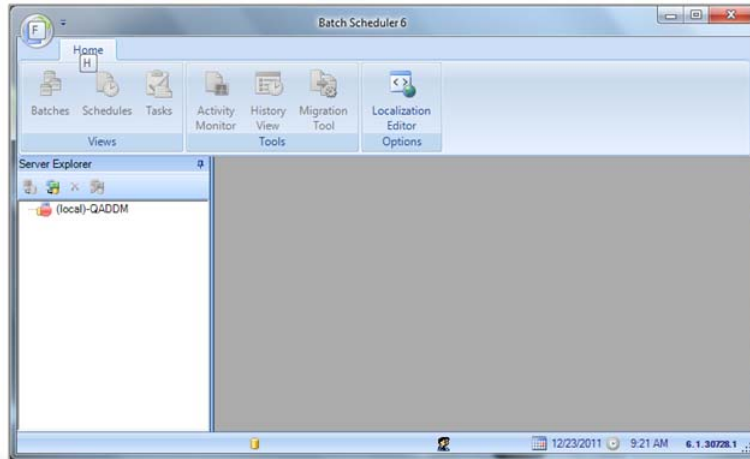
- If the database was updated successfully you will see a success message
 - This will be followed by an installation success window



Installation Validation

Installation Validation

- Launch the **Batch Scheduler Manager** from the Programs menu
 - If the application launches it has been installed correctly



Dm-Install-DME-BS-080

The location of the Batch Scheduler Manager may vary based upon settings in the installation program that are not visible to the user executing the install. In the examples shown here, the “John Galt/Atlas Planning Suite” version of the Batch Scheduler was shown, which may differ slightly from the “QAD Demand Management” version.

Chapter 4

Installing Planning Portal

Preparing for the Portal Installation

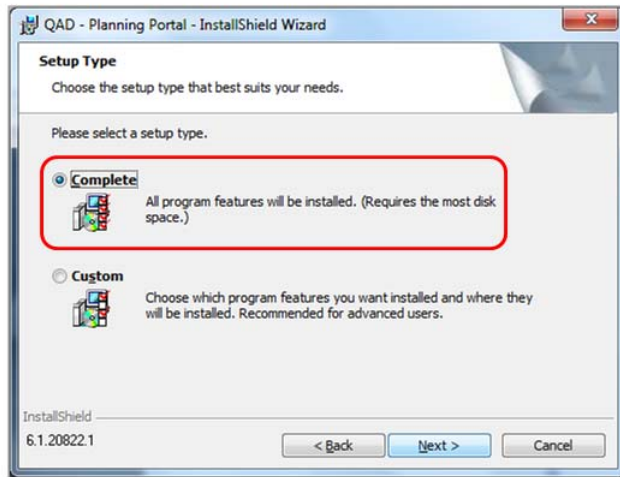
Preparing for the Portal Installation

- Download the SQL software media to an accessible drive on the server or place the media in a drive accessible by the server
- In conjunction with the customer IT team determine the appropriate install locations for
 - Application files
 - Web server components

Setup Type

Setup Type

- For an initial installation of the Planning Portal on a server a **Complete** installation is recommended
- For special situations the **Custom** option can be used to install specific software components



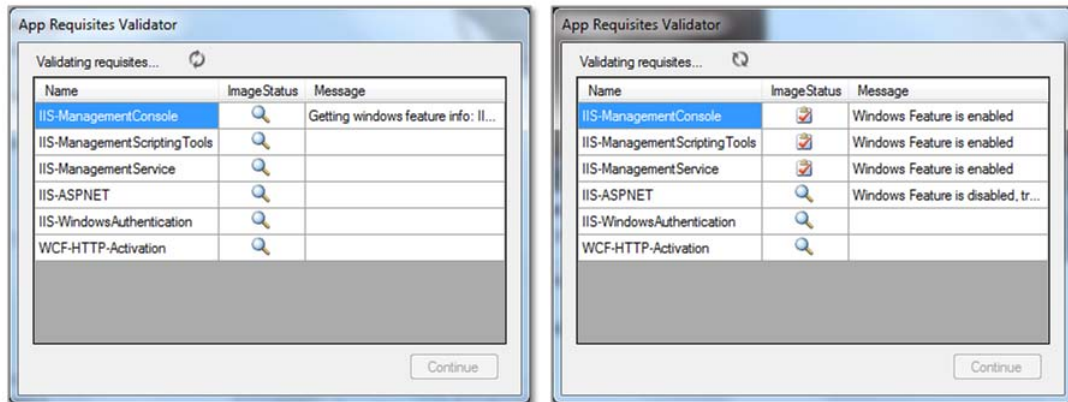
DVA-IPP-030

When installing Planning Portal on the “application server”, a complete installation is recommended.

Installation of Prerequisites

Installation of Prerequisites

- Upon launching the Portal installation, the installation program will determine if you have all software prerequisites installed

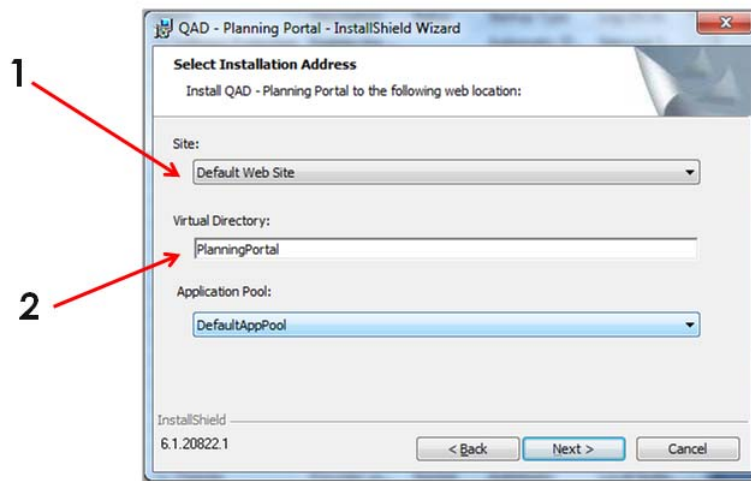


As the installation program validates the presence of prerequisites, it will also enable the feature during the installation as it is required to be enabled for the Portal software to operate.

Installation Address

Installation Address

- Specify the appropriate **Site** and **Virtual Directory** for the Portal installation



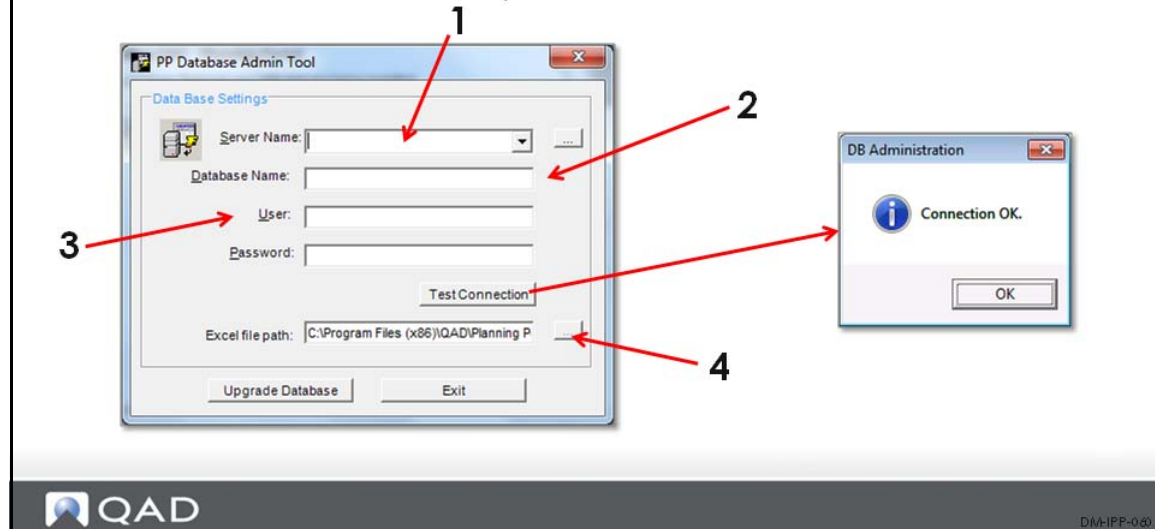
Specifying the appropriate parameters during the Select Installation Address process includes:

- 1 Identify the proper Site. Typically you will choose “Default Web Site” unless other web sites are already present on the server. Consult with the customer for any changes to the default setting.
- 2 Specify the Virtual Directory. This specifies the directory (both from a physical file perspective and a URL folder) that the Planning Portal will reside. The default value is “PlanningPortal”. It is recommended leaving the default name unless the customer has a alternate naming convention for the Portal (examples include: “DMPortal”, “ForecastingPortal”, etc.)

Database Administration Tool

Database Administration Tool

- Within this window, provide the appropriate parameters to allow the installation to connect to the SQL environment and add Portal functionality to the DME database



To connect to the DME database, define the following parameters prior to clicking Upgrade Database:

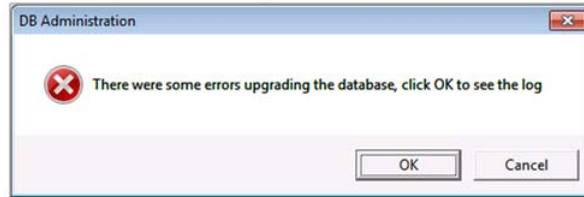
- 1 Enter the server name. If the database is being created on the same server as the DME installation, you can utilize “(local)” as the server name.
- 2 Enter the Database Name. Examples might be FX_JGD, QADDM, etc.
- 3 Enter the user name and password that should be used to connect to the SQL Server. It should have “sa”-level privileges.
- 4 Specify the folder into which the Portal application should save Excel files. The default is shown.

Note You can click “Test Connection” to check the validity of the SQL Server connection using the specified server/user/password.

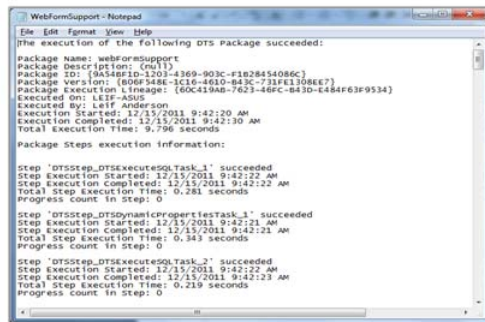
Review of Database Upgrade

Review of Database Upgrade

- Upon executing the **Database Upgrade**, you may see this message



- Click **OK**. This will open the log file
 - Review this file for any installation issues and restart if necessary



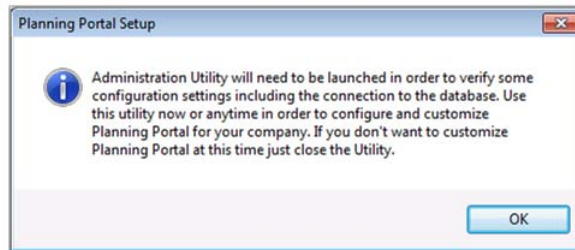
DM-IPP-070

The installation log may highlight specific issues that require your attention. It is possible that you will see messages such as “The execution of the following DTS package succeeded”, however, this may not indicate that a successful installation was achieved. In certain cases, you may have a “successful installation” only to find errors in later steps (such as when launching the Portal Admin Tool).

Validation of Installation

Validation of Installation

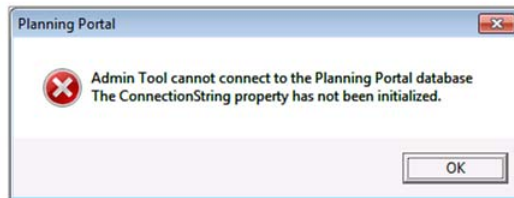
- Upon completion of the **Database Upgrade**, the installation program notifies you that it will launch the "Administration Utility" (known as the **Portal Admin Tool**)



Potential Error Message

Potential Error Message

- Upon launching the Portal Admin Tool you may encounter this error message

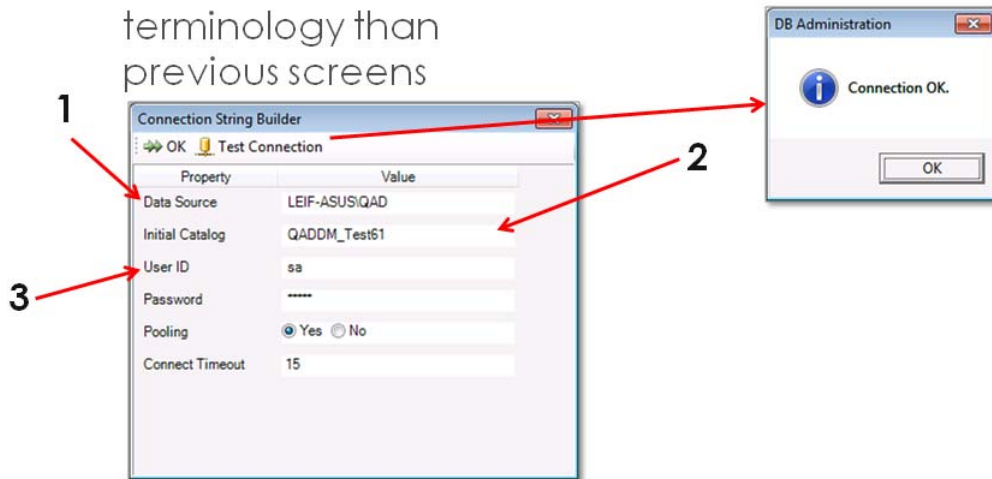


- If this error message appears click **OK** to proceed to the **Connection String Builder**

If you do not see this error message, please continue with the installation.

Potential Error Message

- Provide all of the values for **Connection String Builder**
 - Please note that this window uses different terminology than previous screens



To connect to the DME database, define the following parameters within the Connection String Builder:

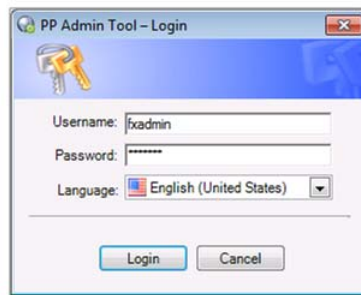
- 1 Enter the server name in the “Data Source” field.
- 2 Enter the Database Name in the “Initial Catalog” field. Examples might be FX_JGD, QADDM, etc.
- 3 Enter the user name and password that should be used to connect to the SQL Server. It should have “sa”-level privileges.

Note You can click “Test Connection” to check the validity of the SQL Server connection using the specified server/user/password.

Accessing the Portal Admin Tool

Accessing the Portal Admin Tool

- Access the Portal Admin Tool by providing your DME user name and password
 - You should utilize the **fxadmin** (administrator) user ID

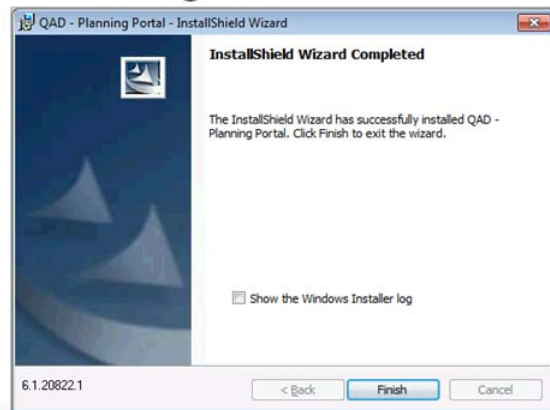


Don't forget that the administrator user ID in DME is always **fxadmin** (this is hard-coded into the software). The default password for this user ID is also "fxadmin".

Successful Installation

Successful Installation

- If you are able to access the Portal Admin Tool you have successfully installed the software
 - Upon exiting the Portal Admin Tool you should see the following window



Chapter 5

DME Database Optimization

SQL Server Memory Settings

SQL Server Memory Settings

- Recommend using the "Use a fixed memory size" option instead of the default setting "Dynamically configure SQL server memory"
- Memory settings should be based on server architecture
 - **Database is on same server as the application** – memory setting should be 25% to 50% of total physical memory
 - **Dedicated SQL Server** – memory setting should be 75% of total physical memory



DMA-DME-DB-Opt-020

The settings above can be found within SQL Server Management Studio:

- Right click on the Database Server instance name and choose Properties
- Click on the Memory page of the Server Properties dialog box

DME Database Recovery Model

DME Database Recovery Model

- The database recovery model option must be set to **Simple**
- Backups should be scheduled using a frequency that minimizes potential data loss
Best practices
 - Daily backups
 - Backup stored on a different physical disk than the database



DME-DME-DB-Opt-030

The settings for the recovery model can be found within SQL Server Management Studio:

- Right click on the database name and choose Properties
- Click on the Options page of the Database Properties dialog box
- Change the recovery model and click OK.

Tempdb Database

Tempdb Database

- The Tempdb holds all temporary tables and temporary stored procedures
 - It fills any other temporary storage for tables generated by SQL Server
 - Use the following settings to achieve the best performance
 - Allow the tempdb database to automatically expand as needed
 - Set the original size of the tempdb database file to a reasonable size to avoid frequent expanding – a good setting is 20% of the Engine database size
 - Set the file growth increment to either "by 20%" or "by 100Mb"
 - The tempdb database should be on a fast I/O subsystem and use a high-speed RAID0 or SAN drive



DM-DME-DB-Opt-040

The settings for the tempdb database can be found within SQL Server Management Studio:

- Right click on the "tempdb" database name and choose Properties
- Click on the Files page of the Database Properties dialog box
- Change the "Autogrowth" settings and click OK.

DME Ideal Disk Configuration

DME Ideal Disk Configuration

- On the SQL Server machine, the system files and the database files should be separated to at least two hard drives:

File/Database	Two Drives (C:, D:)	Three Drives (C:, D:, E:)	Four Drives (C:, D:, E:, F:)
System files	C:	C:	C:
OBV_DATA file group	D:	D:	D:
Other file groups	D:	D:	E:
Tempdb	D:	E:	F:



DME-DME-DB-Opt-050

The settings for the DME database can be found within SQL Server Management Studio:

- Right click on the DME database name and choose Properties
- Click on the Files option to see the current home for the various database files

File Defragmentation

File Defragmentation

- For best performance defragmentation of all hard drives should occur once per month
 - While the default settings for initial and increment size are optimal in most cases, ensure that the settings reflect each file groups particular share of the overall database size

File Groups	Percentage
OBV_DATA	70%
OBV_IDX	15%
SER_DATA	7%
SER_IDX	3%
Primary	5%



DM-DME-DB-Opt-0.00

The settings for the DME database can be found within SQL Server Management Studio:

- Right click on the DME database name and choose Properties
- Click on the Files option to see the current settings for the various database files

Re-indexing the DME Database

Re-indexing the DME Database

- Proper indexes are the key factor for the performance of any database application
 - To maximize the power of the indexes, re-indexing of the DME database is recommended
 - Should be performed on a daily basis
 - A stored procedure named USP_JGS_Reindex is available in the DME database - it reindexes:
 - Observations
 - Series
 - StatisticsValues
 - Dates
 - TBL_FLIFT_CalendarMapping
 - TBL_FLIFT_Calendar
 - Other tables can be added as needed



DME-DME-DB-Opt-070

Re-indexing the DME database is a must. Complaints of slow batch and application performance over time frequently point to this root cause. In most implementations, the stored procedure referenced above will be registered as a custom task within the DME application and scheduled using the Batch Scheduler.

Chapter 6

DME Database Maintenance

DME Database Maintenance

DME Database Maintenance

- Early in the implementation of Demand Management it is highly recommended that you define a **maintenance plan** that accomplishes two objectives
 - **Optimizing database performance** – checking database integrity and updating database statistics
 - **Creating a database backup** – establishing database backups at regularly defined intervals



DM-DME-DB-Maint-020

Companies that already have a SQL Server infrastructure typically already have a strategy in place for database maintenance. Seek the advice of your customers IT team regarding maintenance plans. During the implementation, make sure that a maintenance plan is in place as this may save countless hours of rework in the event a database backup is needed.

DME Database Recovery Model

DME Database Recovery Model

- Prior to defining the maintenance plan, it is important to define the DME **database recovery model**
 - **Three options exist** – simple, bulk-logged and full
 - **Simple** recovery models are the fastest to process
 - minimize log space and allow restoration of the database to the time of backup
 - **Bulk-logged and full** provide the greatest flexibility for point-in-time restores but comes at a higher performance cost
 - **Best practice is to utilize “simple”** – allows for best performance for non-transaction systems such as Demand Management



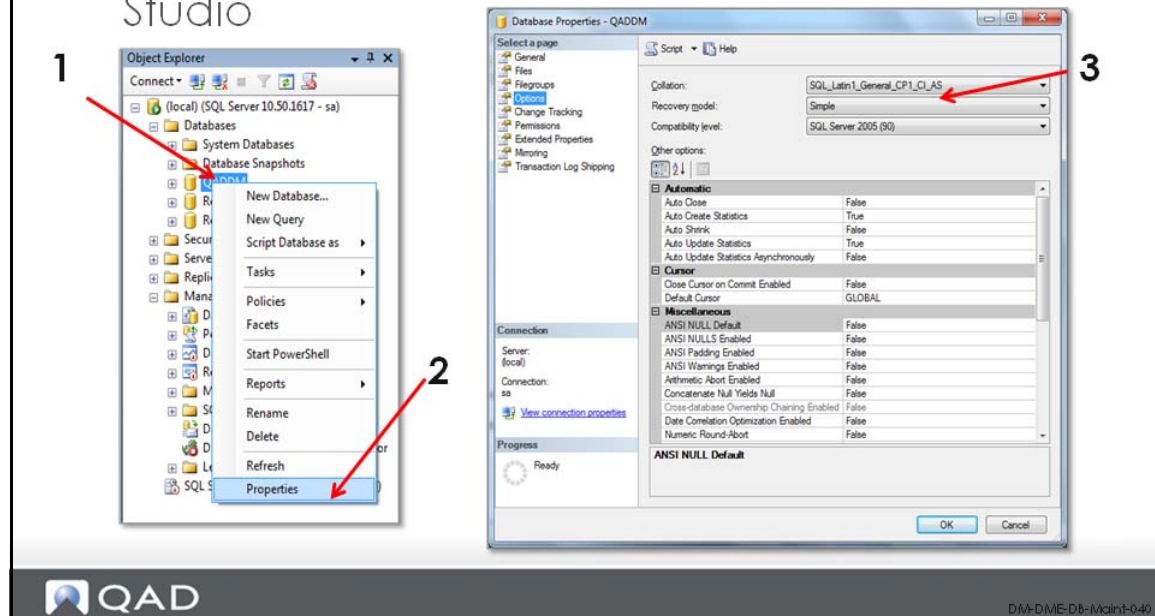
DME-DME-DB-Maint-030

Companies that already have a SQL Server infrastructure typically already have a strategy in place for database maintenance. Seek the advice of your customers IT team regarding maintenance plans.

The best practice for backups is to use a “simple” model. As DME is not a transaction-oriented system, the potential for data loss is minimized by establishing a backup plan that includes daily backups.

DME Database Recovery Model

- A database **recovery model** can be established within SQL Server Management Studio



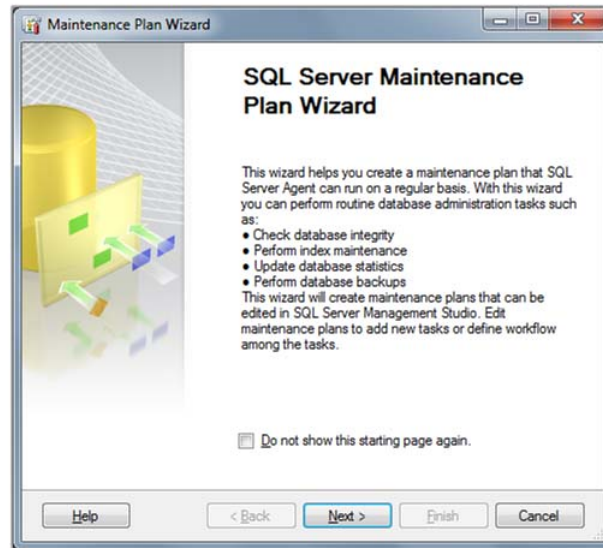
To create a recovery model, follow this procedure:

- 1 Right-click on the database name in SQL Server Management Studio.
- 2 Select Properties.
- 3 In the Recovery Model field, select “Simple” (or other desired model) and click OK.

Creating a Database Maintenance Plan

Creating a Database Maintenance Plan

- A database **maintenance plan** can be established within SQL Server Management Studio

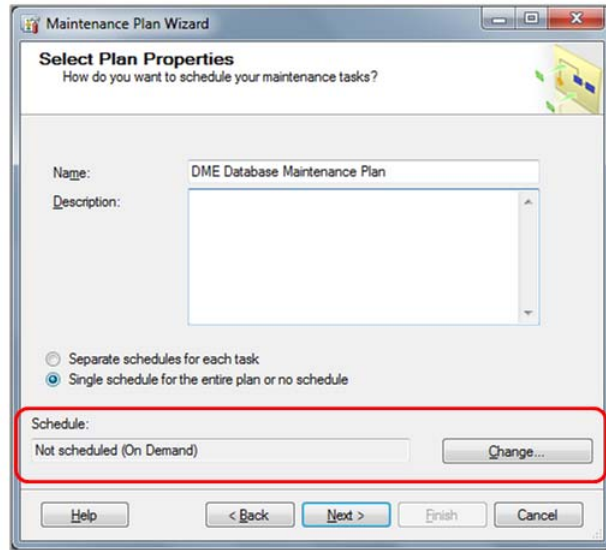


Within SQL Server Management Studio, open the Management Folder, right-click Maintenance Plans and select “Maintenance Plan Wizard”.

Maintenance Plan Properties

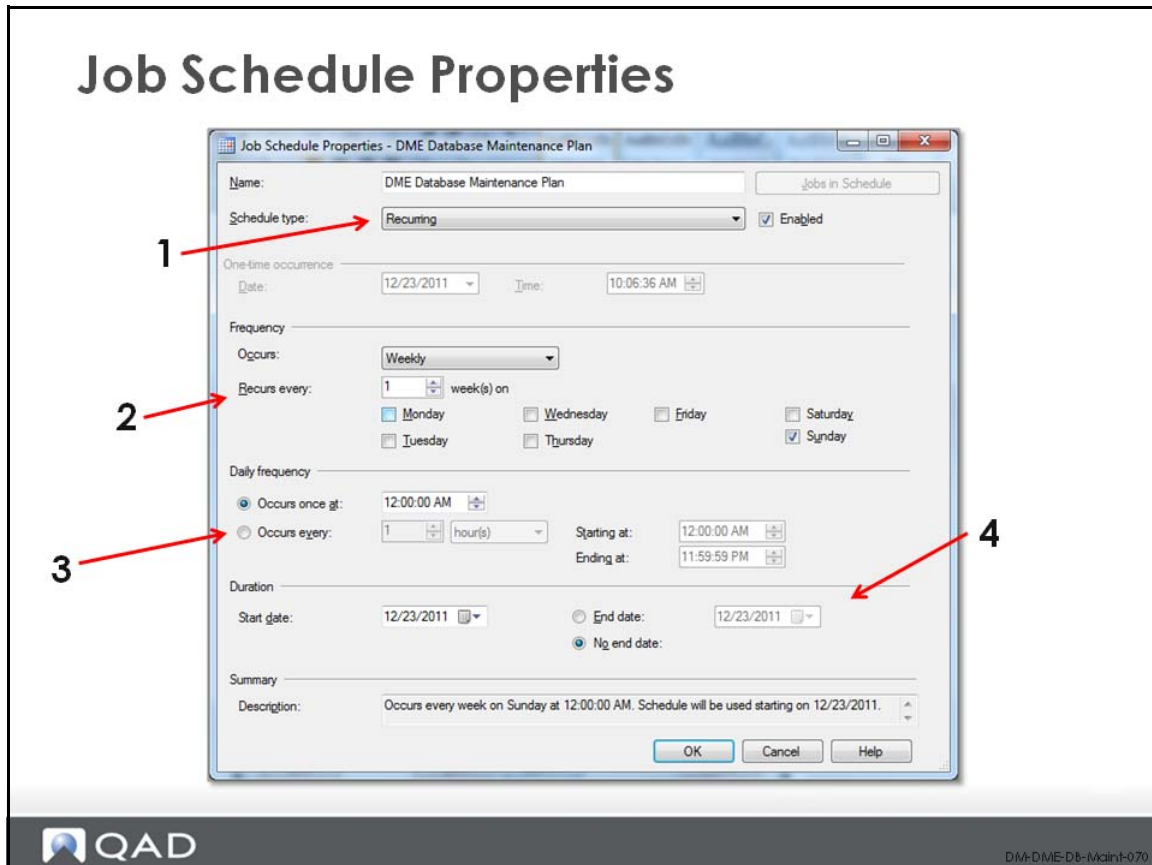
Maintenance Plan Properties

- The **maintenance plan** properties allow you to specify:
 - Plan Name
 - Plan Schedule



Within SQL Server Management Studio, open the Management Folder, right-click Maintenance Plans and select “Maintenance Plan Wizard”.

Job Schedule Properties



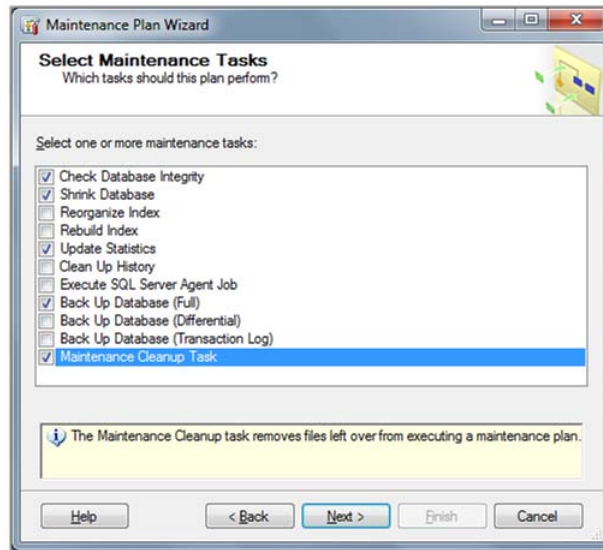
Various parameters can be defined:

- 1 The schedule type allows for recurring or one-time execution.
- 2 Frequency allows for the definition of frequency of execution (daily, weekly, etc.). Pick specific days of the week as needed.
- 3 Within the selected days, daily frequency allows for the definition of when during that day the plan should be executed.
- 4 The duration settings allow for a fixed time range if necessary.

Selecting Maintenance Tasks

Selecting Maintenance Tasks

- Various **maintenance tasks** can be selected within the wizard



DM-DME-DB-Maint-080

At a minimum, the selected maintenance tasks should include:

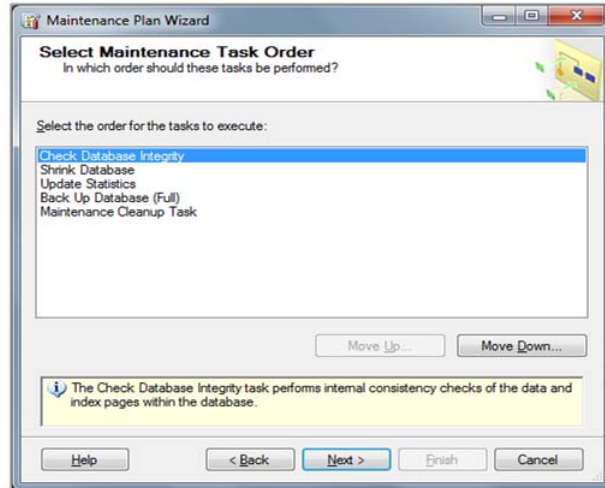
- 1 Check Database Integrity
- 2 Shrink Database
- 3 Update Statistics
- 4 Back Up Database
- 5 Maintenance Cleanup Task

Database reindexing is typically achieved through a DME custom task. You may choose to handle backups separately depending on your customer's needs.

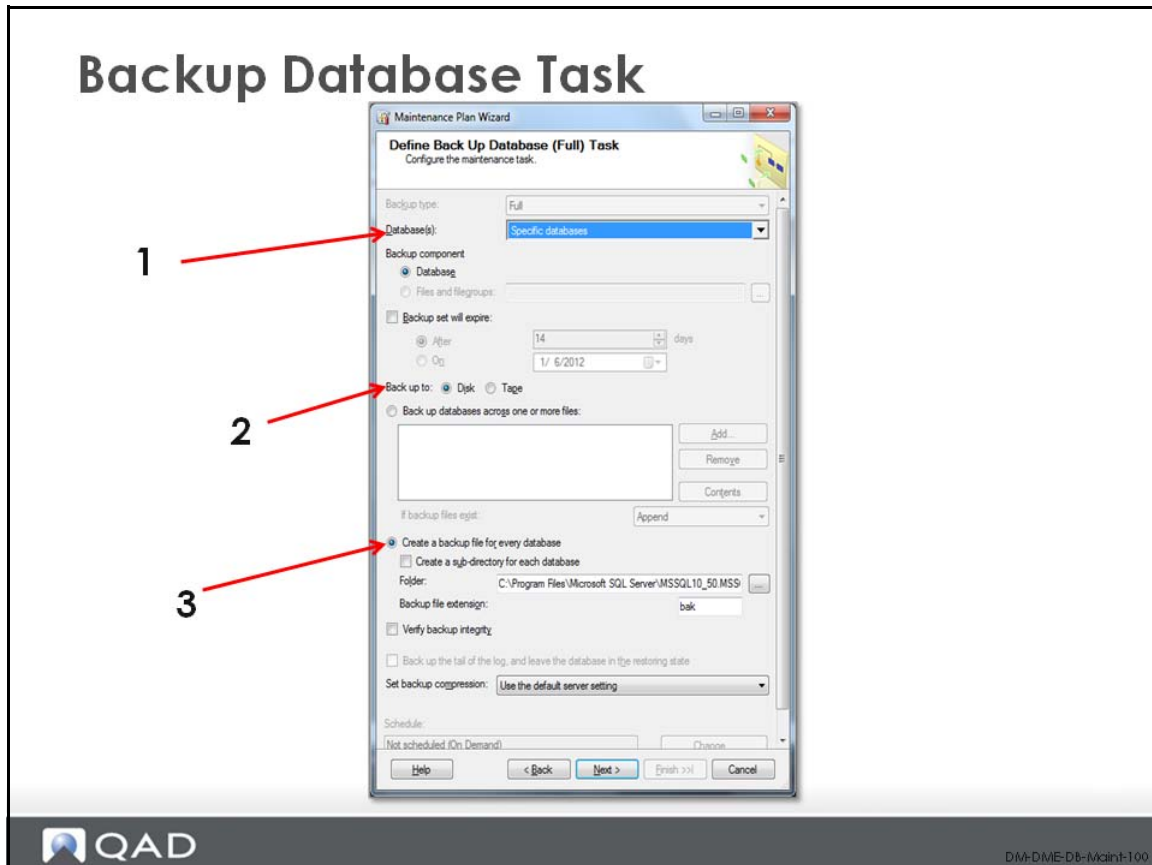
Selecting Maintenance Task Order

Selecting Maintenance Task Order

- The various maintenance tasks can be arranged in an order of your choosing
- It is recommended to execute the database backup first



Backup Database Task



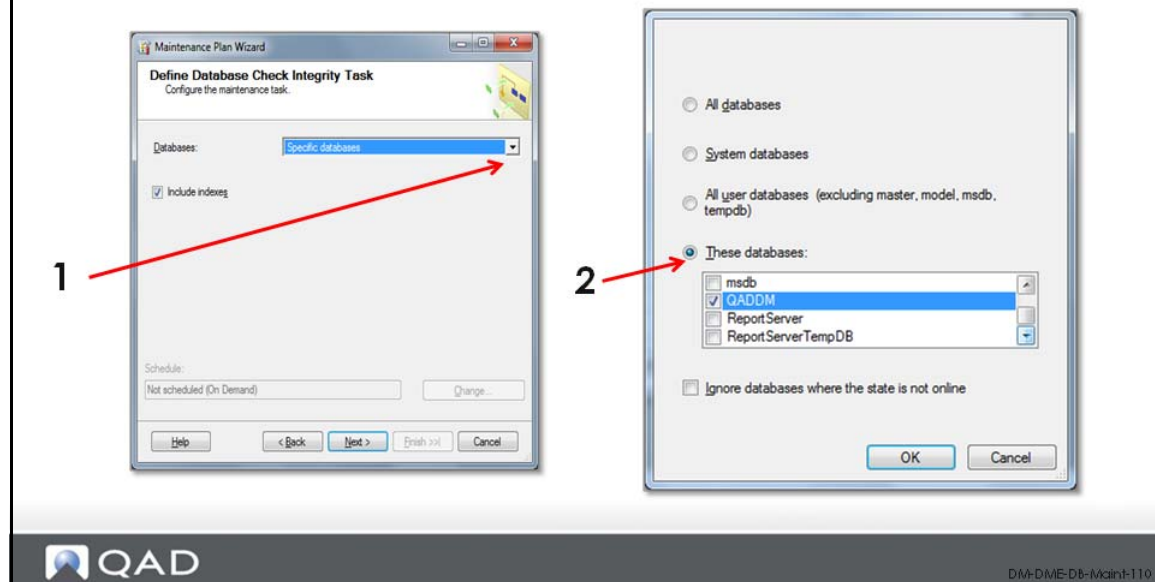
Various parameters can be defined:

- 1 Click the drop-down list under databases and choose Specific databases.
- 2 Specify the backup type (disk, tape)
- 3 Specify the location of the backup folder.

Check Integrity Task

Check Integrity Task

- As part of this task select the **database(s)** that you wish to be included as part of the task

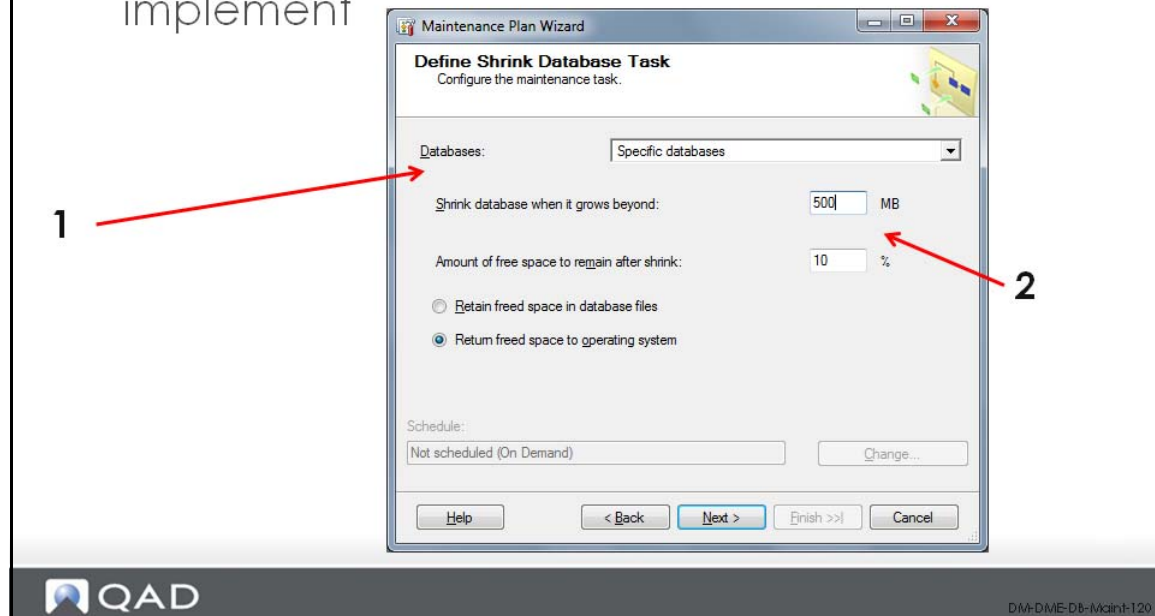


- 1 Click the drop-down list under databases and choose Specific databases
- 2 Select the exact databases you require (such as the QADDM database)

Shrink Database Task

Shrink Database Task

- As part of this task select the **database(s)** and shrink **parameters** that you wish to implement

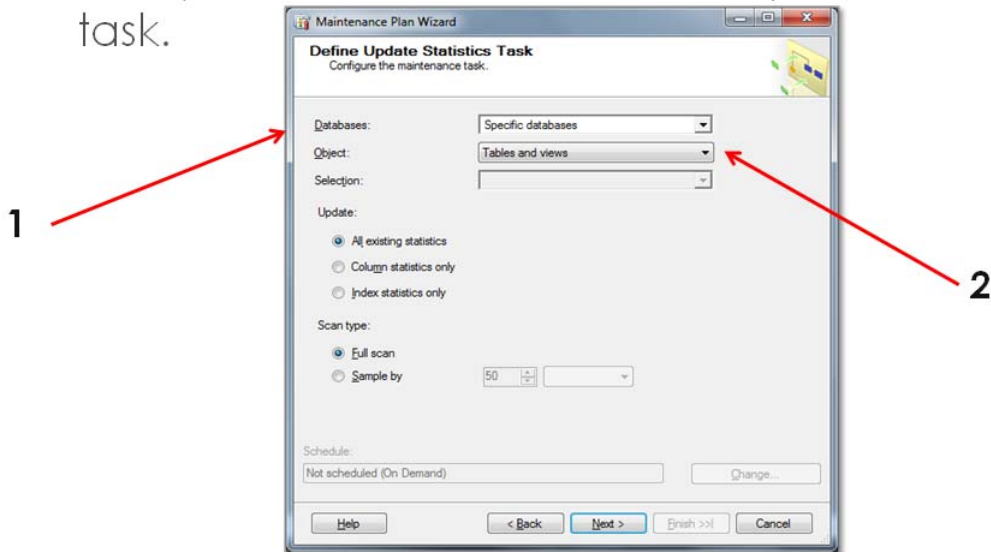


- 1 Click the drop-down list under databases and choose Specific databases.
- 2 Select the shrink parameters. Select a “shrink database when it grows beyond” parameter that makes sense relative to the overall size of the database. This should be a minimum of 500mb or at least equal to the current size of the DME database times 1.5.

Update Statistics Task

Update Statistics Task

- As part of this task, select the **database(s)** that you wish to be included as part of the task.

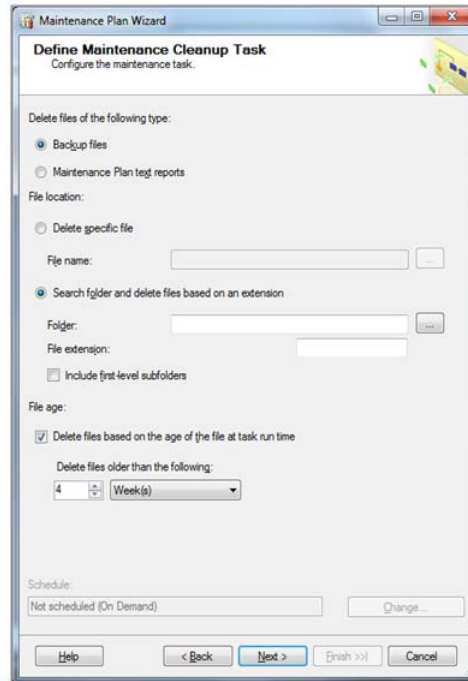


- 1 Click the drop-down list under databases and choose Specific databases.
- 2 Select tables and views under the Object drop-down list.
- 3 All other parameters can be left at their default settings unless otherwise directed by the customer.

Maintenance Cleanup Task

Maintenance Cleanup Task

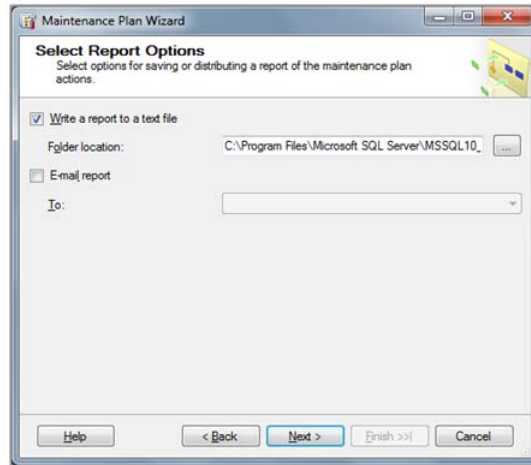
- This task can cleanup previously created backup files and report files
- Defaults can be used unless otherwise directed by the customer



Report Options

Report Options

- To create a report displaying the **maintenance plan actions** specify the destination or email recipient

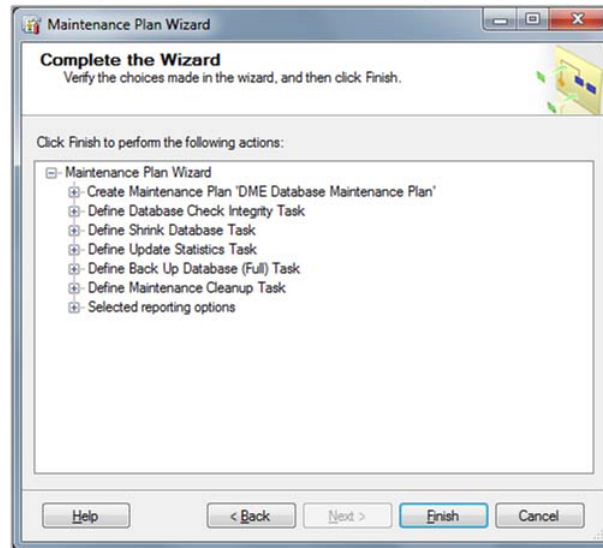


The E-Mail report functions require that SQL database mail functions have been activated.

Maintenance Plan Review

Maintenance Plan Review

- Prior to creation of the maintenance plan **review** the plan created by the wizard



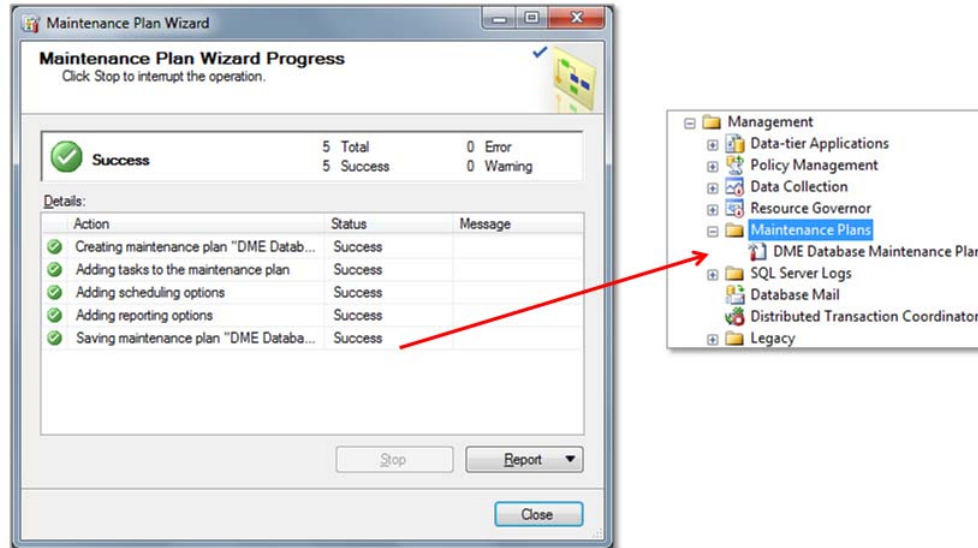
DW-DME-DB-Maint-1.60

If you need to change any of your options, simply click “Back” and navigate to the relevant option set.

Creation of the Maintenance Plan

Creation of the Maintenance Plan

- The maintenance plan will be created and visible within the **Maintenance Plans** folder



If any errors occur during the Plan build, you will be notified in the Maintenance Plan Wizard Progress window.

If successful, the plan will be visible in the Maintenance Plans folder under Management in your database instance.

Chapter 7

DME Database Basics

DME Database Overview

DME Database Overview

- The Demand Management Engine (DME) database is stored within SQL Server
- Some key facts related to the database
 - The default database name during installation is "FX_JGD"
 - During implementation this is typically changed to "QADDM" or "QADDM_Production"
 - With a few exceptions almost all DME configuration is stored within the DME database



DM-DME-DB-Basics-020

The default database name (FX_JGD) stands for "Forecast X – John Galt Development". During implementation, this name is usually changed to QADDM or some similar derivation (such as "QADDM_Test" for a test database and "QADDM_Production" for a production database).

As almost all DME configuration is stored within the database, it is extremely difficult to "copy over" changes from one database to another. Configuration changes typically have to be replicated by the QAD Consultant or Super-User and not through an automated means.

DME Database Key Tables

DME Database Key Tables

- Although there are over 100 tables in the DME database, these primary tables house a majority of data
 - **Series** – This table stores the individual series contained in each scenario along with their “series attributes”
 - **Observations** – This houses all of the observations data (aka “opinion lines”) associated with each series. There is one database record per series per time period
 - **Scenarios** – This table stores details associated with each scenario



DME-DME-DB-Basics-030

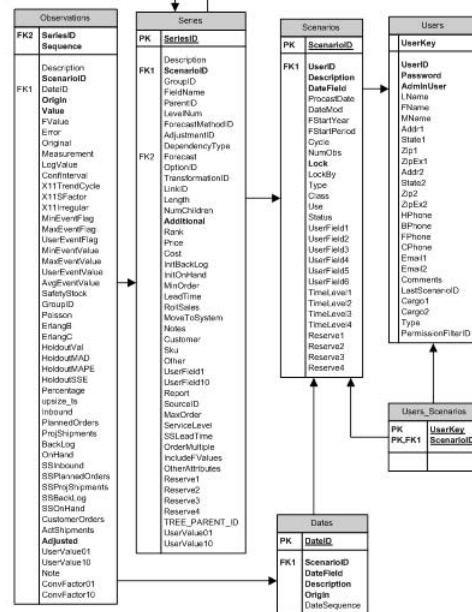
Understanding the contents of these three tables is key to a successful implementation. Many customers will utilize external reporting tools (such as Cognos, Crystal, Access, etc.) to tap into the DME database.

Your ability to relate the data contained in these three tables to what a user sees within the DME Viewer will make the “external reporting” phase of an implementation much easier.

DME Database Key Tables

- Many of the tables within the database link to each other using one or more of these fields

- **ScenarioID** – a numeric ID associated with a scenario
- **SeriesID** – a numeric ID associated with an individual series
- **UserID** – a numeric ID associated with an individual user



DM-DME-DB-Basics-0-40

Within SQL Server Management Studio, you can view the primary keys for each database table. This will assist you in understanding how tables link together.

DME Database Key Tables

- To maximize database performance, the DME database is divided into the following six file groups
 - The Series and Observations tables are in their own file groups because they are the largest tables and are accessed on a frequent basis

File Group	File	Description
PRIMARY	FX_JGD_DATA	Statistics and other meta data tables
SER_DATA	SER_DATA	Data in the Series table
SER_IDX	SER_IDX	Indexes for the Series table
OBV_DATA	OBV_DATA1, OBV_DATA2	Data in the Observation table
OBV_IDX	OBV_IDX	Indexes for the Observation table
(N/A)	FX_JGD_LOG	Log files for transaction processing



After you installed the DME database, you should see seven unique files in the folder you specified.

The filenames listed in this slide may vary from your actual implementation depending on the name given to the database (if you changed FX_JGD to QADDM, then the primary file group’s file would be “QADDM_DATA” rather than “FX_JGD_DATA”).

DME Database Important Notes

DME Database Important Notes

- It is not recommended that users or super-users directly access database tables
- Any manipulation of data contained within the DME database should be accomplished through custom tasks (stored procedures)

Chapter 8

Pivot Table View Maintenance

Pivot Table Overview

Pivot Table Overview

- The Demand Management Engine (DME) allows users to create Microsoft Excel pivot tables from within the Viewer
- The pivot tables are driven by SQL Views defined within the DME database
- Two default views are provided as part of the installation
 - VW_PVT_VIEW_V40
 - VW_PVT_DRP_V40
- During implementation, these views should be modified to meet customer requirements



DIM-PI-Tb-View-Install-020

The Pivot Table functionality (along with SQL Views) allows users to create “reports” that can be used outside of the Viewer.

As the data is exported to a Microsoft Excel spreadsheet, the spreadsheet can be shared with anyone the user wishes.

This functionality is “one-way”, that is, data only flows FROM the DME database to the spreadsheet and not the other direction.

Default View Definition (vw_PVT_View_V40)

Data Field	Definition
Utr	Engine UserID
SeriesID	Unique series ID number
Scn	Scenario
U01-10	Series.UserField1 through 10
SKU	SKU/Part Number
Customer	Customer Name (if used this way)
DT	Date (YYYY-P)
Origin	"Hist" if Observations.Origin = 1 or "Fcst" if Observations.Origin = 2
YR	Year
QTR	Quarter
MON	Month
MAPE	Mean Absolute Percentage Error
RSqr	R-Squared
HIST_DMD	Historical Demand (Observations.Value where Origin = 1)
FCST_DMD	Forecasted Demand (Observations.Fvalue where Origin = 2)
MDLD_DMD	Fitted Demand (Observations.Fvalue where Origin = 1)
SFTY_DMD	Safety Stock Values
HIST_REV	Price * Historical Demand
FCST_REV	Price * Forecasted Demand
MDLD_REV	Price * Fitted Demand
SFTY_REV	Price * Safety Stock Values
ConvFactor01-10	Observations.ConvFactor01 through 10
UserValue01-10	Observations.UserValue01 through 10

The view named vw_PVT_View_V40 contains a variety of fields from the series, observations and statistics values tables.

It is highly recommended that this default view be customized during a customer implementation in order to make the pivot table's contents relevant to their needs and their DME configuration.

Types of Data to Display in a Pivot Table

Types of Data to Display in a Pivot Table

- Series Fields
 - Examples: Part number - customer name - product family etc
- Observations Fields (Opinion Lines)
 - Examples: Sales history- statistical forecast - consensus forecast
- Custom Fields
 - **Formulas** - Example: Projected revenue (defined by consensus forecast times price)
 - **Logical** – Example: “History/Forecast” (if time period is in the past, use sales history, else use consensus forecast)
- External Data Sources
 - Data from any other database can be linked as long as there is a logical link between data



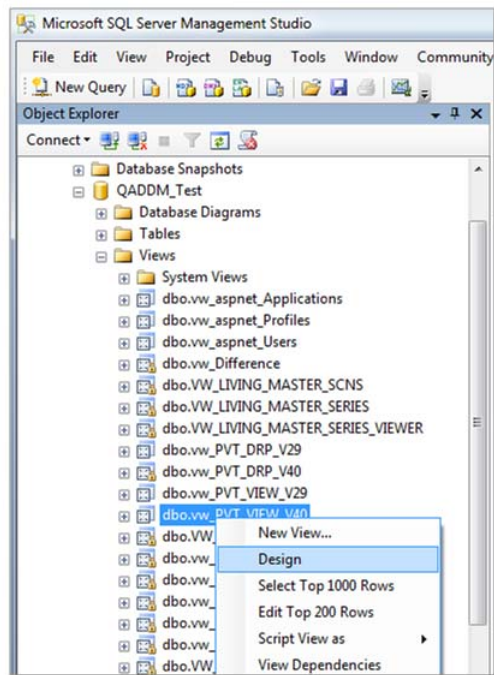
DM-Pt-Tb-View-Maint-040

The SQL View gives us the opportunity to pass along a “customized” dataset to the Excel pivot table. Determine your customer’s needs when creating or customizing a pivot table SQL view.

Modifying a SQL View

Modifying a SQL View

- Utilize SQL Server Management Studio for easy SQL view modification



DM-Pt-Tb-View-Maint-050

From within SQL Server Management Studio:

- 1 Expand the + next to the DME database name.
- 2 Expand the + next to the Views folder
- 3 Right-click on the desired view and click “Design”
- 4 The design window appears on the right side

Modifying a SQL View

- Use the view design window to add, modify and delete view fields

SQL query building through field selection

Direct modification of a SQL query

Column	Alias	Table	Output	Sort Type	Sort Order	Filter
UserID	Usr	US	<input checked="" type="checkbox"/>			
Other		SR	<input checked="" type="checkbox"/>			
LevelNum	LvNum	SR	<input checked="" type="checkbox"/>			
SeriesID		SR	<input checked="" type="checkbox"/>			

```

SELECT US.UserID AS Usr, SR.Other, SR.LevelNum AS LvNum, SR.SeriesID, SC.Description AS Scr, SR.Userf
SR.UserField4 AS uf04, SR.UserField5 AS uf05, SR.UserField6 AS uf06, SR.UserField7 AS uf07, SF
SR.UserField10 AS uf10, SR.Sku, SR.Customer, OB.Description AS DT, CASE OB.Origin WHEN 1 TH
LEFT(DT.DateField, 4) AS YR, CASE CAST(RIGHT(DT.DateField, 2) AS int)
WHEN 1 THEN 'Q01' WHEN 2 THEN 'Q01' WHEN 3 THEN 'Q01' WHEN 4 THEN 'Q02' WHEN 5 THEN 'C
WHEN 9 THEN 'Q03' WHEN 10 THEN 'Q04' WHEN 11 THEN 'Q04' WHEN 12 THEN 'Q04' ELSE 'N/A' E
SV.RSqr, CASE OB.Origin WHEN 1 THEN OB.Value ELSE 0 END AS HIST_DMD, CASE OB.Origin WH
  
```

From within SQL Server Management Studio:

- 1 Add, delete or modify the contents of the view as necessary.
- 2 Click “save” when completed.

Note The design window also shows the corresponding SQL. Changes to the SQL script can be made directly within this area as an alternative to using the visual selection tools.

Pivot Table SQL View Notes

Pivot Table SQL View Notes

- For the SQL view to be visible within the Viewer's pivot table functionality, the following requirements must be met:
 - The name of the SQL view must begin with "vw_PVT_"
 - The view must include "ScenarioID" as one of the data fields (from Scenario.ScenarioID)
 - The view must include "SeriesID" as one of the data fields (from Series.SeriesID)
- The parameters behind any pivot table created by the DME can be modified directly within the spreadsheet (use "Ctrl-Shift-H" to access the parameter worksheet)



DME-Pivot-View-Maint-070

If your SQL view does not appear in the Viewer's list of available views, it is most likely due to one of the reasons listed in this slide.

Additional SQL View Notes

Additional SQL View Notes

- The same SQL views can be used by other reporting tools such as Cognos or Crystal
- Additional filtering criteria may need to be utilized to limit the amount of data retrieved by the view
- Add fields to the SQL view judiciously
 - Excel pivot tables can easily grow well past 100mb



DM-Pi-Tb-View-Maint-080

Pivot tables can be very large if the customer's database has a large number of series and/or a large number of fields as part of the view.

It is very common for customer pivot tables to be in excess of 200mb.

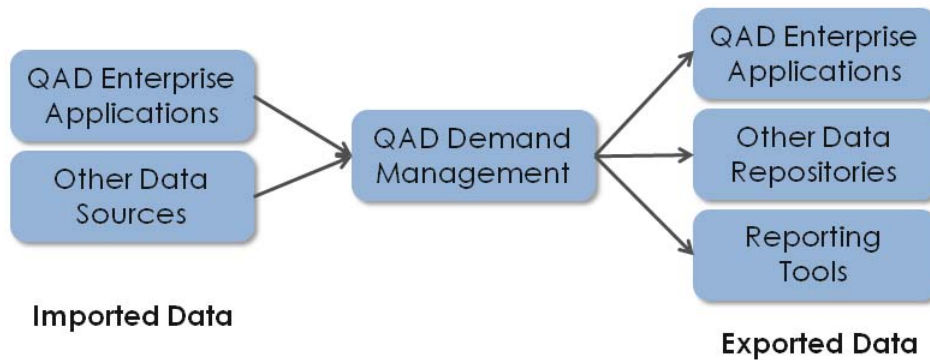
Chapter 9

Introduction to Integration with QAD

DME-QAD Integration Overview

DME-QAD Integration Overview

- The Demand Management application is essentially a stand-alone application
- It was not developed to integrate with any specific ERP application



DME-QAD Integration Overview

- Data Imported into Demand Management
 - QAD is a common data source for DM
 - Invoice History
 - Part Master Attributes
 - Customer Master Attributes
 - Data can be imported from a variety of sources
 - SQL
 - Excel/Access
 - ODBC Connection
 - Data must be imported in a specific format
 - Data must typically be transformed prior to loading into DM (in order to present in the correct format)



DM-INT-QAD-030

It is important to note that there is no standard integration between QAD Enterprise Applications and QAD Demand Management.

Not only are the two applications supported by different database engines (Progress vs. SQL), every customer requires a different set of data in order to support their DME implementation.

DME-QAD Integration Overview

- Data Exported from Demand Management
 - QAD is a common destination for DM data
 - Item/Site/Week Forecast
 - Safety Stock
 - Forecast data is exported using a QAD-developed custom task
 - Configurable SQL Stored Procedure to produce a .csv file in a QAD-ready format
 - Must specify what opinion line(s) are used
 - Must specify what DM fields represent item and site
 - Requires definition of a DM Calendar Scenario (for monthly forecasting scenarios)
 - File is FTPed to QAD server (typically by the above SQL procedure)
 - The Demand Planning menu functions of QAD (36.5.9) are used to import the file into Forecast Maintenance



DM-INT-QAD-040

Refer to the (36.5.9) “Work Aid” for specific instructions on the setup of these menu functions.

DME-QAD Integration Overview

- Data Exported from Demand Management
 - Other output files can easily be created
 - SQL's export functions (of views or tables)
 - Reporting tools can easily read SQL databases
 - A view can be created for easy access and formatting of data
 - Tables can be accessed directly and joined for proper extraction



DM-INT-QAD-050

Most reporting tools on the market today can read SQL databases. Thus, every customer has many choices when it comes to extracting data from DME.

DME-QAD Integration Details

DME-QAD Integration Details

- **Key task:** Developing an SSIS package for QAD data extraction
 - Key software requirements
 - Process utilizes SQL Server Business Intelligence Development Studio or Visual Studio
 - Requires Progress to SQL ODBC drivers
 - Gathering business requirements
 - Determine types of data required to extract from QAD
 - Opinion line data (time-series data)
 - Attribute data
 - Determine source tables/fields for data
 - Create QADDM_Staging database
 - This database stores all data extracted from QAD
 - Create one table for each unique extract



DME-QAD Integration Details

- Developing an SSIS package (con't)
 - Write SQL queries to pull relevant fields
 - Recommend raw data extracts with little transformation
 - Build SSIS package
 - First step: truncate tables in QADDM_Staging to remove data from previous package execution
 - Subsequent steps: one step for each query
 - Final step(s): perform any transformation required to convert raw data into required DM format (may alternately use SQL views for low complexity transformation)
 - Publish SSIS package into SQL instance

DME-QAD Integration Details

- Formatting Data for DM Loading
 - Sales (Invoice) History or any other time series data
 - Specific fields:
 - Series Key Fields (example: Item, Customer)
 - Time buckets 1 to xx representing the sales history for time periods (weeks or months) 1 to xx (representing the total number of past periods that the customer wishes to load into DM)
 - NOTE: There should be one record per unique series (example: one record per Item/Customer combination)
 - Attributes Data (example: Part Attributes)
 - Specific fields:
 - Series Key Fields (example: Item)
 - Series Attributes Field(s) (example: Item Desc, Product Line, Item Status, Std. Cost, etc.)
 - One numeric field containing a zero

DME-QAD Integration Details

- Formatting Data for DM Loading (con't)
 - Transformation required to put data into the required format can be accomplished through:
 - A transformation task established as part of the SSIS package
 - Stored procedures that manipulate data within the QADDM_Staging database
 - Views that organize data from the QADDM_Staging database into the proper format
 - General recommendations:
 - Views should not be used for very large datasets (more than 5,000 summarized records as this may slow data loading in DM)

Housing Data Extracted from QAD

Housing Data Extracted from QAD

- Secondary Database: QADDM_Staging
 - Purpose: serves as a staging area for data extracted out of QAD for use by DM
 - Populated by a SQL Server Integration Services (SSIS) package
 - Tables containing data extracted from QAD
 - QAD_ih_hist: contains invoice history extracted from QAD
 - QAD_pt_mstr: contains part master attributes from QAD
 - Other tables as determined by the implementation requirements
 - User-Defined Tables
 - Typically start with "UserDefined_" prefix
 - Contains data used for a specific purpose within DM
 - Contains data defined by users, not extracted from another system



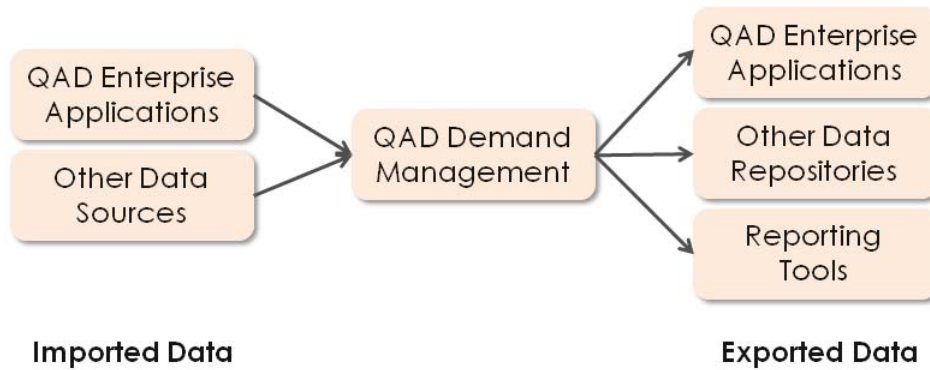
Chapter 10

QAD-to-DM Integration Preparation

Integration Overview

Integration Overview

- The Demand Management application is essentially a stand-alone application
- It was not developed to integrate with any specific ERP application



Preparation Overview

Preparation Overview

- What data is required to extract from QAD?
- How does a Windows Server connect to a QAD Enterprise Applications database?
- Where will the extracted data reside?

Data to be Extracted

Data to be Extracted

- Most DM implementations require the following types of data to be extracted from QAD
 - **Invoice History** – this data represents what was sold in the past
 - **Part Master Attributes** – this data describes the products that a company sells
 - **Customer Master Attributes** – this data describes the customers that a company sells to
- Data can be extracted from any table within QAD



DM-QAD-DMInt-Prep-040

An important point to remember is that data can be extracted from any QAD table – even custom tables. Your customer’s business requirements will drive the appropriate choice. There is no limit to the number of data extracts that can be utilized.

Connecting the DM Server to QAD

Connecting the DM Server to QAD

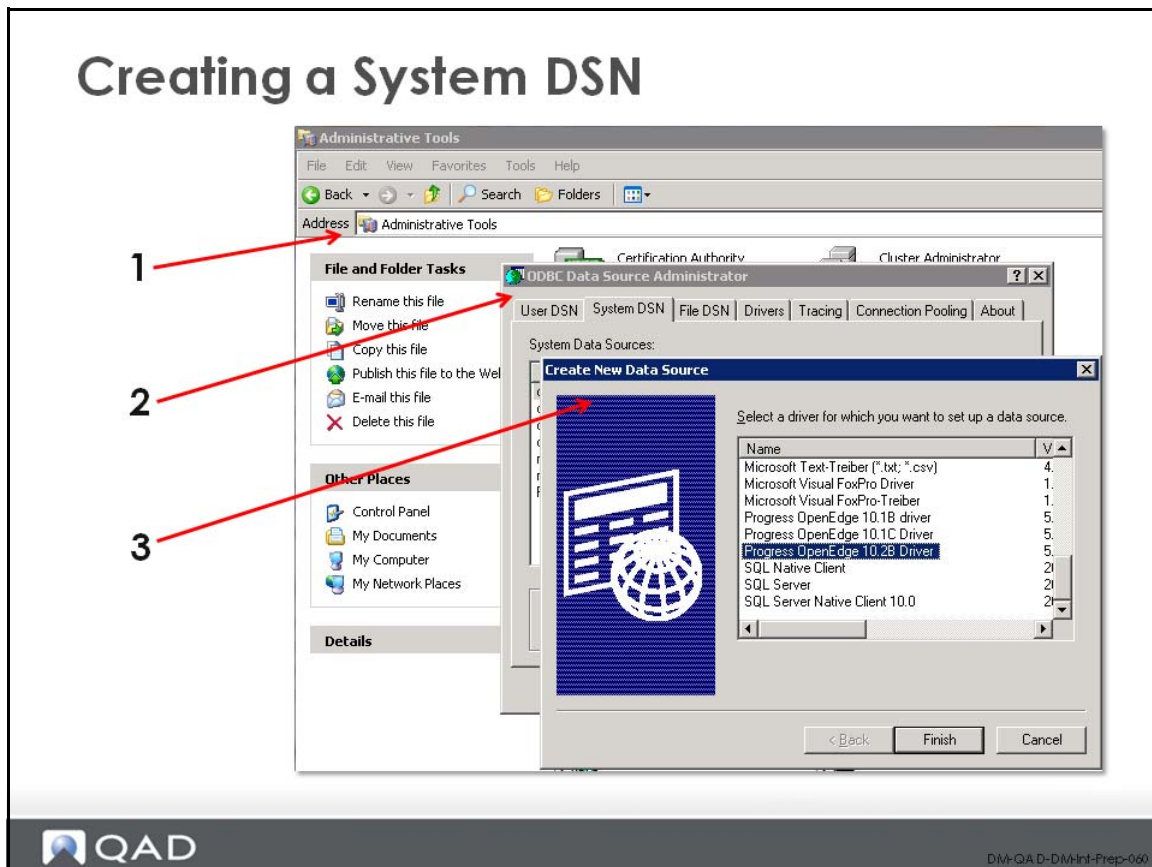
- The Windows Server on which the DM SQL Server database is installed can connect to QAD via ODBC
- The Progress ODBC drivers requires the installation of the Progress client on the DM server
- A "System DSN" is created to establish the connection between the server and QAD



DM-QAD-DMInt-Prep-050

This presentation does not address the installation process for the Progress client. Refer to its installation guide or a subject matter expert for more details.

Creating a System DSN



To create a system DSN, follow this procedure:

- 1 Open Control Panel -> Administrative Tools.
- 2 Select Data Sources (ODBC) and click on the System DSN tab and click “Add.”
- 3 Select the appropriate Progress driver.

Creating a System DSN

Creating a System DSN

- Determine the following entries:
 - DSN Name
 - Host Name (IP/Server)
 - Port Number
 - Database Name
 - User ID
- Use the "Test Connect" function to test the connection

The screenshot shows the 'ODBC Progress OpenEdge Wire Protocol Driver Setup' dialog box. It has four tabs: 'General', 'Advanced', 'Security', and 'About'. The 'General' tab is selected. The fields are as follows:

Field	Value
Data Source Name:	QADTest
Description:	
Host Name:	10.160.0.66
Port Number:	10101
Database Name:	mfgtest
User ID:	qaddm

At the bottom of the dialog, there are four buttons: 'Test Connect', 'OK', 'Cancel', and 'Apply'. A 'Help' button is also present next to the Data Source Name field.



DM-QAD-DMInt-Prep-070

The important task here is to verify that the connection actually works. Use the Test Connect button to test the connection. If the connection is not working, you won't be able to design the integration package that extracts data from QAD.

Creating a Staging Database

Creating a Staging Database

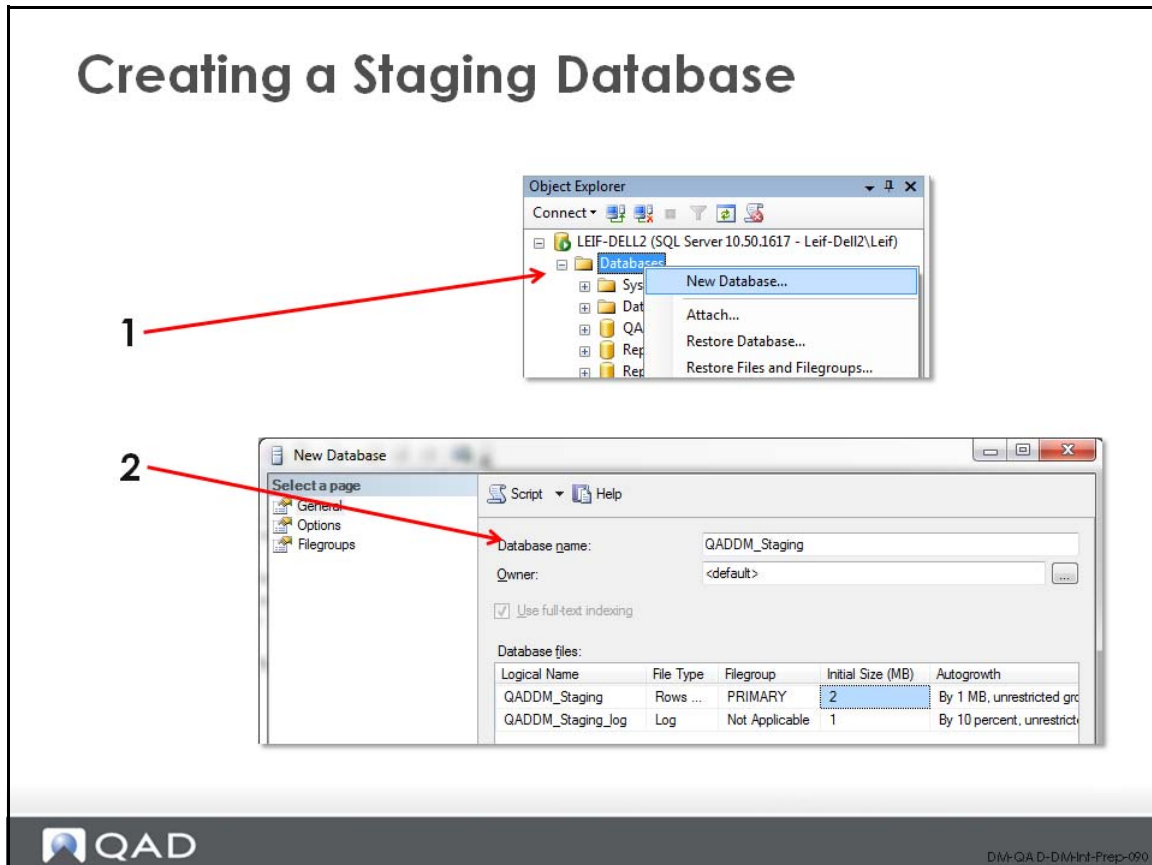
- Data extracted from QAD cannot be imported directly into Demand Management
 - DM has very specific data formats
 - Extracted data requires summarization and transformation prior to loading into DM
- A "staging database" must be created on the SQL Server to house the data extracted from QAD
 - Create a new database named QADDM_Staging (or similar name) in the same SQL instance
 - Create one table per data extract



DM-QAD-DMInt-Prep-060

The staging database is essential to integrating QAD and DM. Very little data transformation is recommended as part of the data extraction process, primarily because of limited functionality of the ODBC drivers as well as the desire to minimize data transfer times. It is recommended that all data transformation and summarization occur on the SQL Server.

Creating a Staging Database



To create a new staging database, open SQL Server Management Studio and follow this procedure:

- 1 Right-click Databases and select New Database.
- 2 In the “New Database” window, enter the Database Name. In the “Database files” portion of the window, you can specify a specific location for the database files to be stored.
- 3 Click OK when completed.

Chapter 11

QAD-to-DM: Creating an SSIS Package

What is an SSIS Package?

What is an SSIS Package?

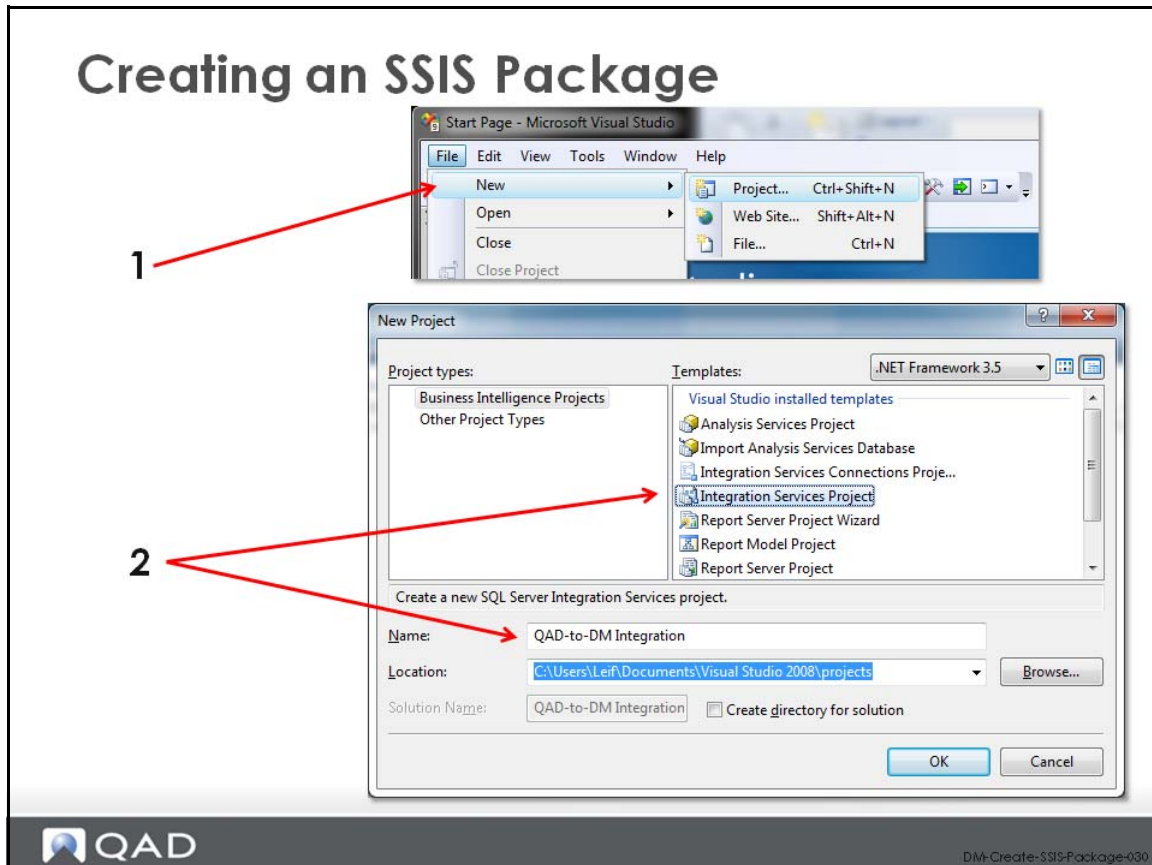
- To extract data from QAD, we must build a SQL Server Integration Services (SSIS) package
- An SSIS package is a collection of data-related actions
 - In this context, it is a collection of queries and actions designed to extract data from QAD and send it to the QADDM_Staging database
- SQL Server Business Intelligence Development Studio (BIDS) is an excellent tool to create SSIS packages
 - BIDS is part of the Microsoft Visual Studio product



DM-Create-SSIS-Package-020

The SQL Server Business Intelligence Development Studio, along with the ability to execute SSIS packages, requires SQL Server Standard Edition or higher.

Creating an SSIS Package



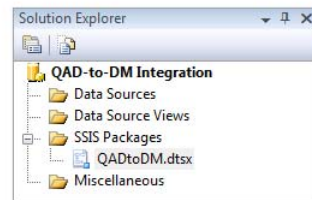
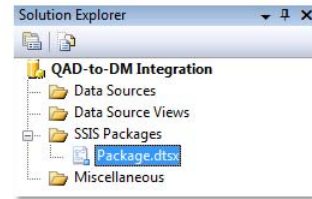
Open SQL Server Business Intelligence Development Studio. To begin a new SSIS package, follow this procedure:

- 1 Select File -> New -> Project.
- 2 In the “New Project” window, select the Integration Services Project template. Give the project a name, define the location where the project should be saved and click OK.

Creating an SSIS Package

Creating an SSIS Package

- In the **Solution Explorer** window, you will see "Package.dtsx" under the SSIS Packages folder
- Change the name of the package to a more meaningful name such as:
QADtoDM.dtsx



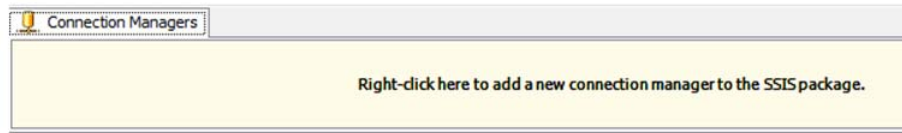
DM-Create-SSIS-Package-040

The name assigned to the package is important. You'll need to know it at a later step in the development process. It's always recommended to give the package a name that everyone understands (especially the customer!)

Connection Managers

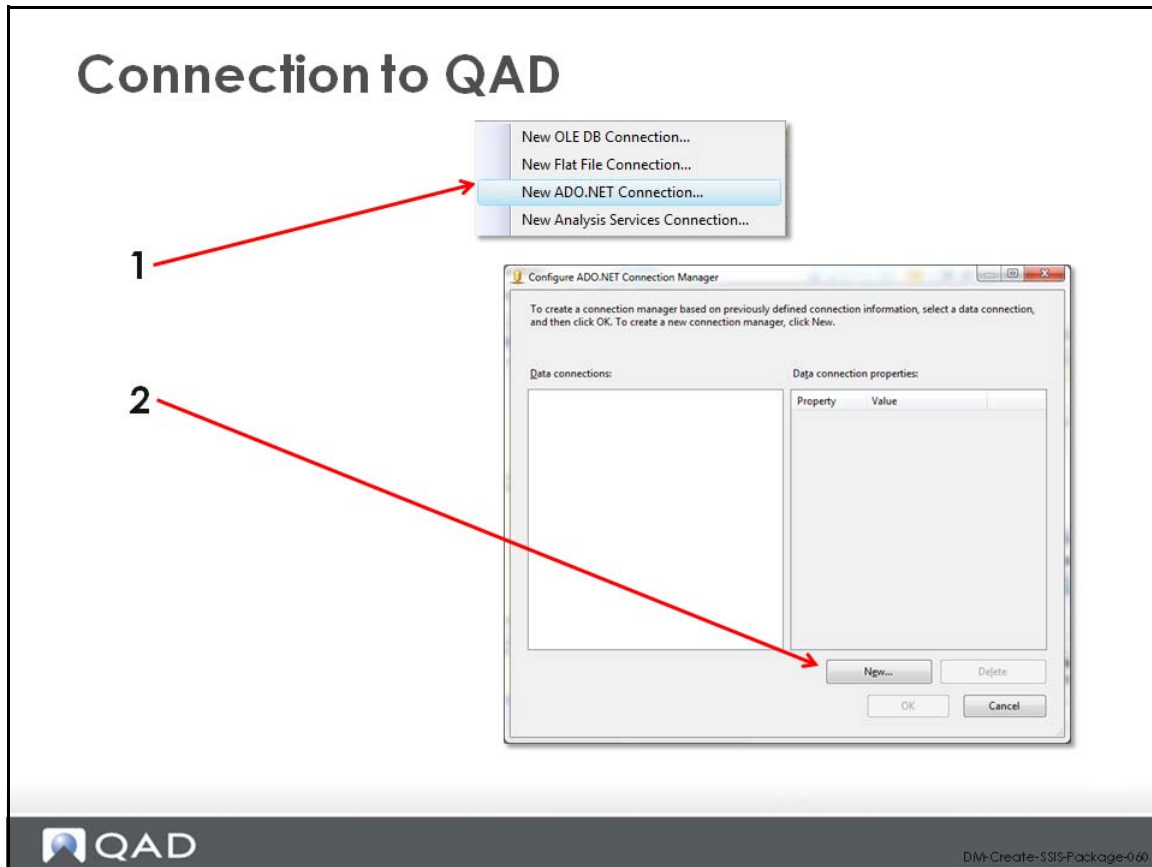
Connection Managers

- The **Connection Managers** window allows us to define the connections to various data sources and destinations
- We have to define a minimum of two connection managers
 - QAD Enterprise Apps (a data source)
 - QADDM_Staging database (a data destination)



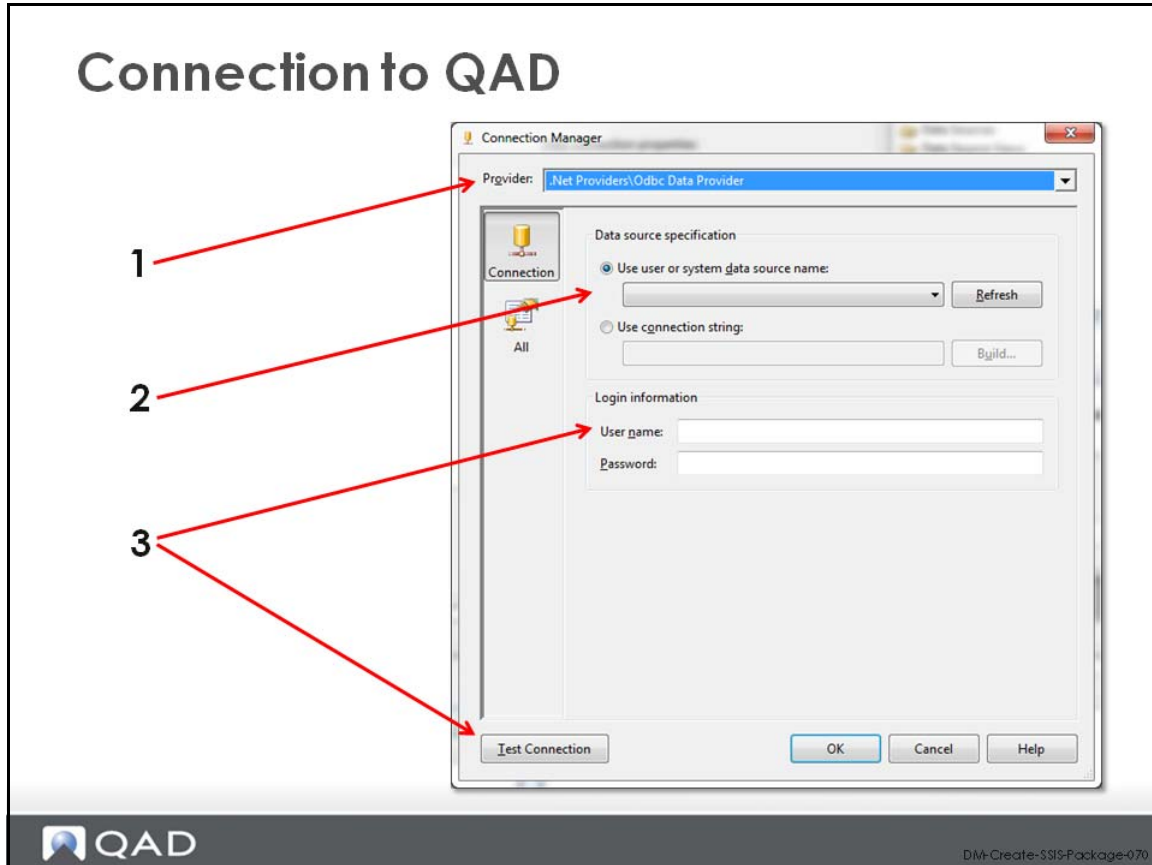
The name assigned to the package is important. You'll need to know it at a later step in the development process. It's always recommended to give the package a name that everyone understands (especially the customer!)

Connection to QAD



To create your connection to QAD, follow this procedure:

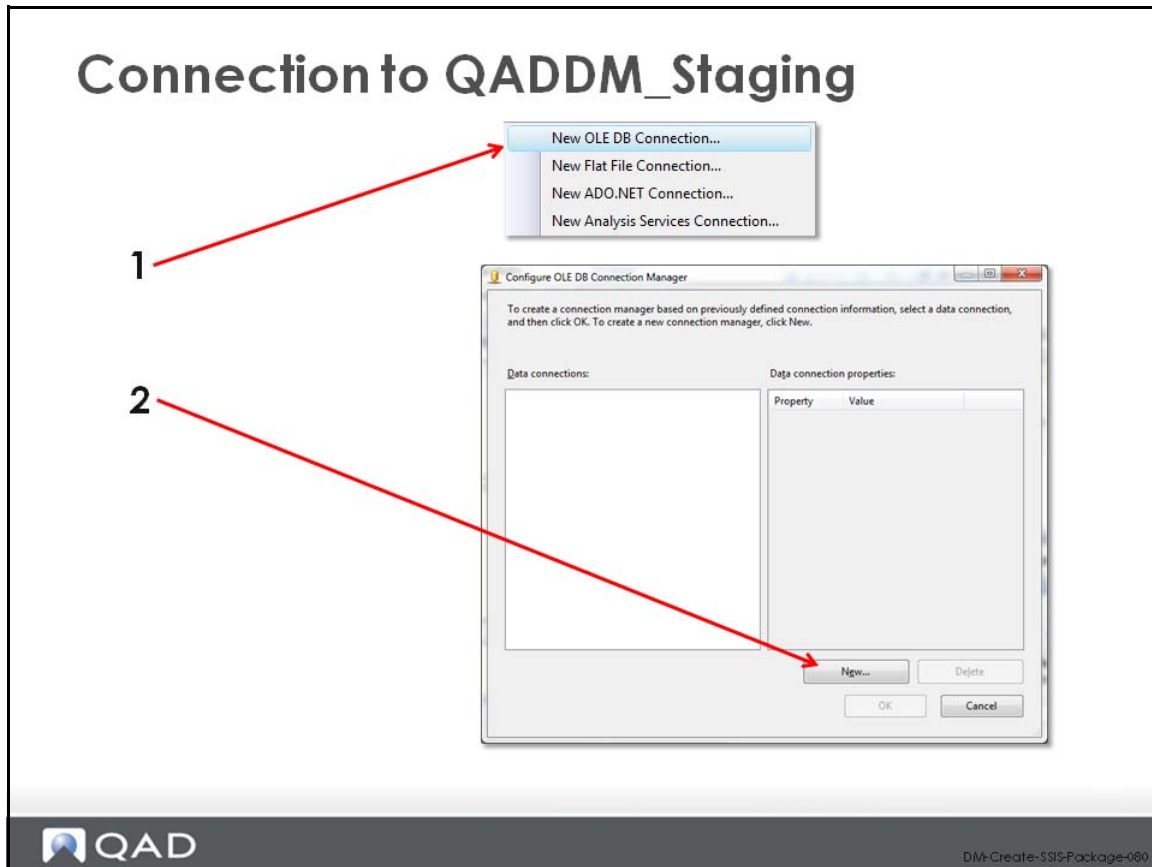
- 1 In the “Connection Managers” window, right-click and select “New ADO.NET Connection.”
- 2 In the “Configure ADO.NET Connection Manager” window, select “New...”



To create your connection to QAD, follow this procedure:

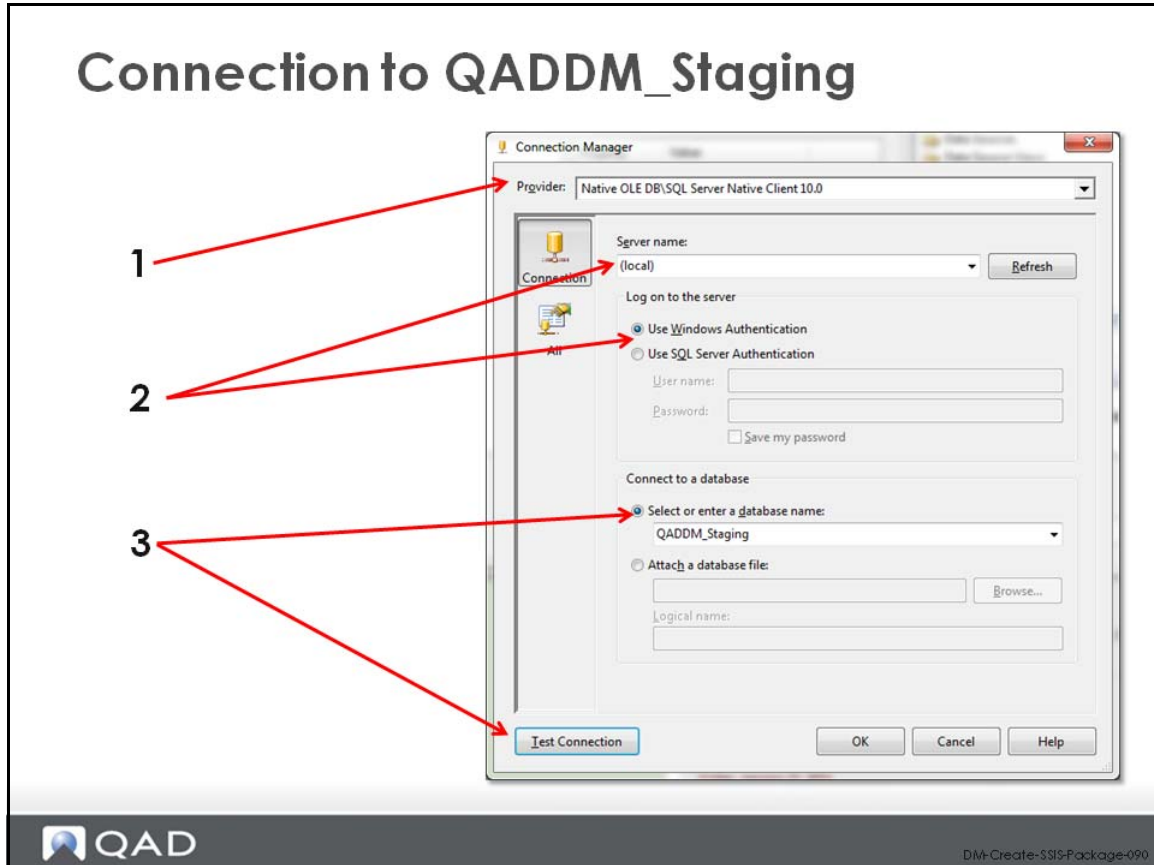
- 1 In the “Provider” drop-down, select “.Net Providers\Odbc Data Provider.”
- 2 In the “Use user or system data source name” drop-down, select the name of the DSN created for QAD connections.
- 3 Enter the appropriate user name and password in the “Login information” section. Click “Test Connection” to test the validity of the connection.

Connection to QADDM_Staging



To create your connection to QAD, follow this procedure:

- 1 In the “Connection Managers” window, right-click and select “New OLE DB Connection”.
- 2 In the “Configure OLE DB Connection Manager” window, select “New...”



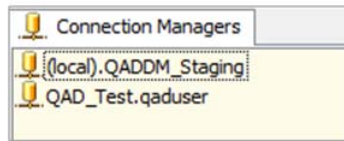
To create your connection to QAD, follow this procedure:

- 1 In the “Provider” drop-down, select “SQL Server Native Client 10.0” (depending on your version of SQL Server, you may have different choices – the key is to choose a “SQL Server” provider).
- 2 Enter the name of the SQL Server. If you are on the server itself, you may choose to use “(local)” to designate the use of the local server. In addition, enter the credentials required to log on to the SQL server (you may choose either Windows or SQL Server Authentication – make sure you are using “administrator” or “database owner” level permissions).
- 3 Select the “QADDM_Staging” database. Click “Test Connection” to test the validity of the connection.

Connection Manager

Connection Manager

- When you have completed the setup for the two connections, your **Connection Managers** window should have two entries



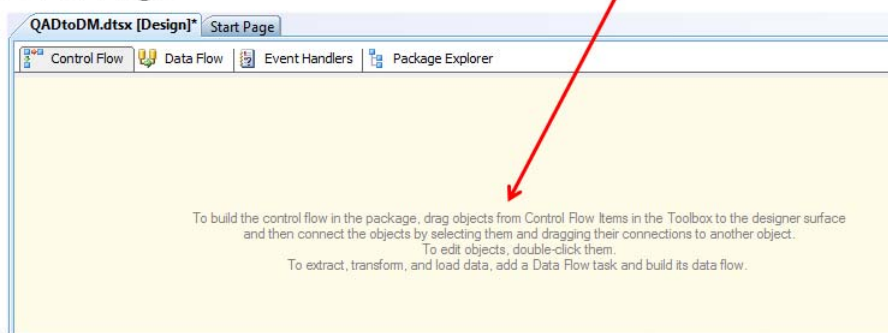
DM-Create-SSIS-Package-100

There is no limit to the number of connection managers that can be defined. It is common for multiple connections to be present – such as one to the Production QAD database, one for Test, etc.

The Package Screen Layout

The Package Screen Layout

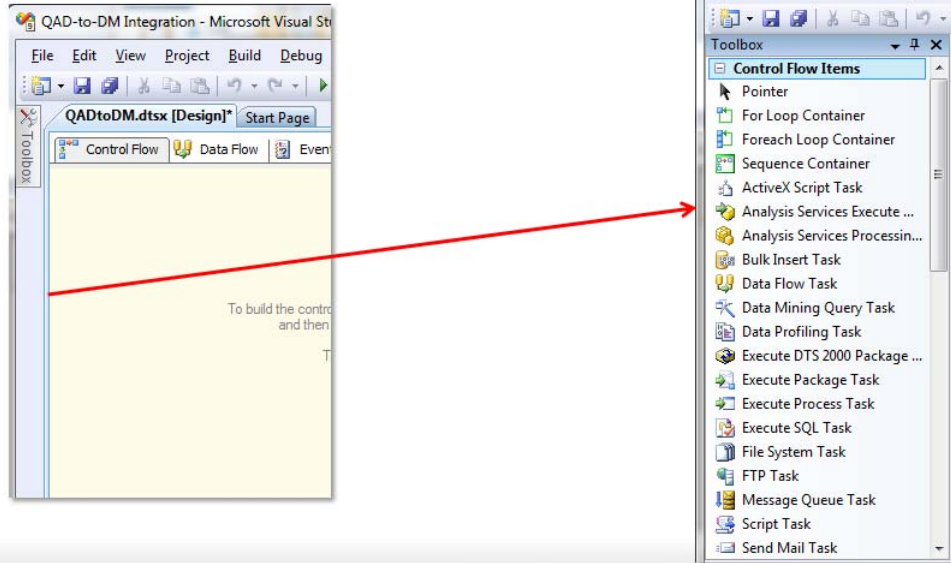
- The middle of the BIDS screen layout shows the package's "Control Flow"
 - In a new package this screen is blank
- Please read the information presented in this screen as it helps you start building the package



Activating the Toolbox

Activating the Toolbox

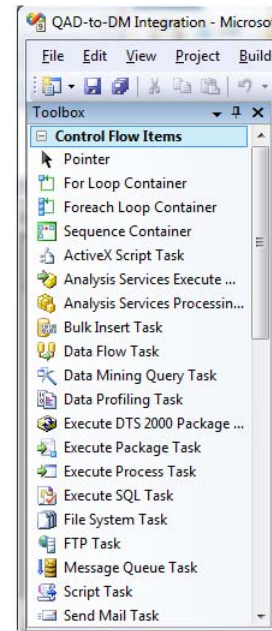
- Click and “pin” the “Toolbox” tab to view the available toolbox options



About the Toolbox

About the Toolbox

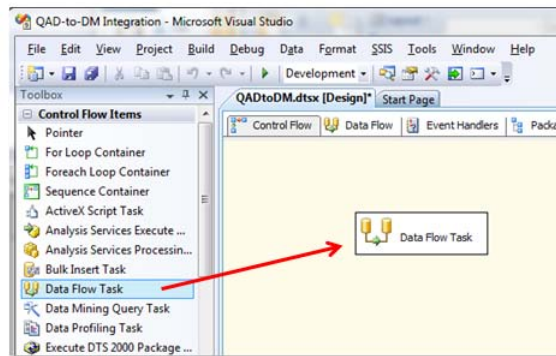
- Two “control flow items” are commonly used in the development of a QAD-to-DM integration package:
 - **Data Flow Task** – this allows for the “flow” of data from a source to a destination
 - **Execute SQL Task** – this allows for the execution of a SQL script to perform a data transformation



Adding a Task to the Control Flow

Adding a Task to the Control Flow

- To add a task to the Control Flow, simply highlight, drag and drop the task onto the Control Flow page
- In this example, a "Data Flow Task" has been added to the Control Flow



Next Steps

Next Steps

- At this point, you have created an SSIS package with the following characteristics
 - Unique package name
 - Defined connection managers
 - Visible toolbox items
- In the next section, you will learn how to:
 - Create a Data Flow Task that extracts data from QAD
 - Create an Execute SQL Task that performs data transformations in the QADDM_Staging database

Chapter 12

QAD-to-DM: Creating a Data Flow Task

What is a Data Flow Task?

What is a Data Flow Task?

- A Data Flow Task allows for the movement of data from a data source to a data destination
- In the context of QAD-to-DM integration, this task allows for the extraction of data from a QAD database to a SQL database
- A typical QAD-to-DM SSIS package will have at least two Data Flow Tasks
 - Invoice History
 - Part Master Attributes

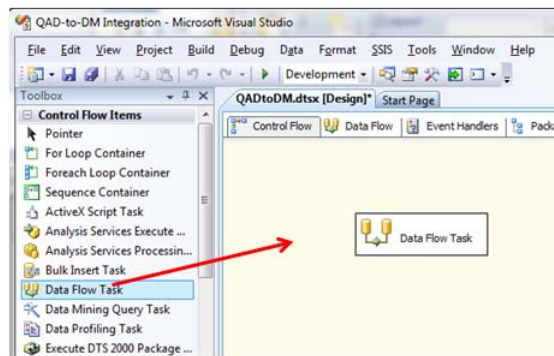


DM-Data-Flow-Task-020

Adding a Task to the Control Flow

Adding a Task to the Control Flow

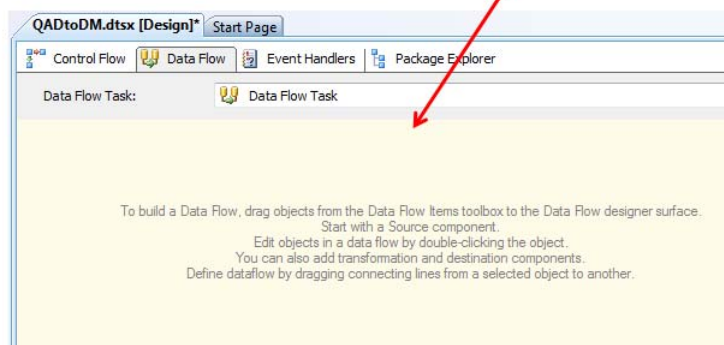
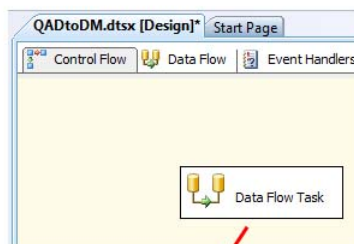
- To add a task to the Control Flow, simply highlight, drag and drop the task onto the Control Flow page
- In this example, a "Data Flow Task" has been added to the Control Flow



Building a Data Flow Task

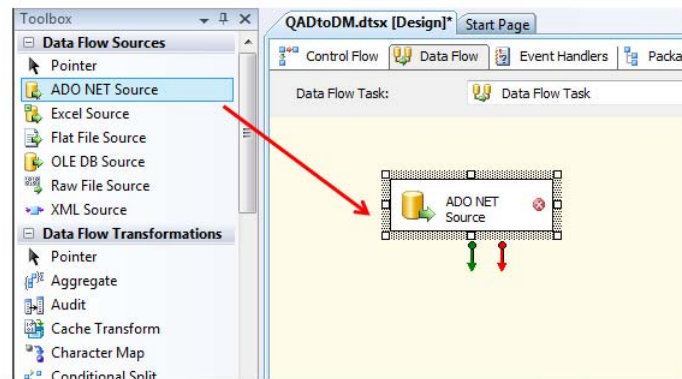
Building a Data Flow Task

- Double-click the Data Flow Task in the Control Flow window
 - A blank Data Flow Task Window appears



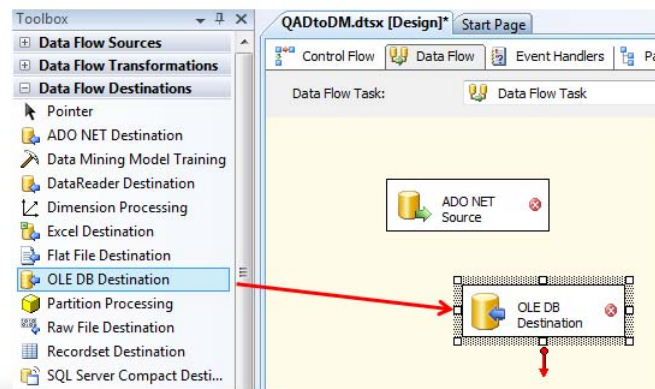
Building a Data Flow Task

- The first piece of a Data Flow Task is a Data Flow Source
- To add a Data Flow Source to the Data Flow, highlight, drag and drop the desired Data Flow Source onto the Data Flow page



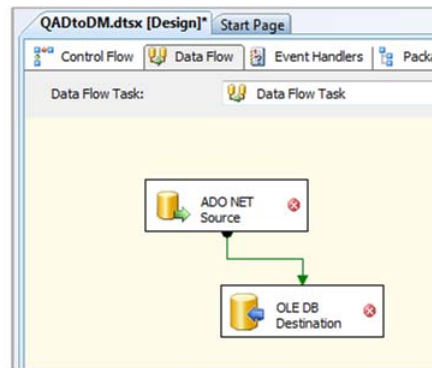
Building a Data Flow Task

- The second piece of a Data Flow Task is a Data Flow Destination
- To add a Data Flow Destination to the Data Flow, highlight, drag and drop the desired Data Flow Destination onto the Data Flow page



Building a Data Flow Task

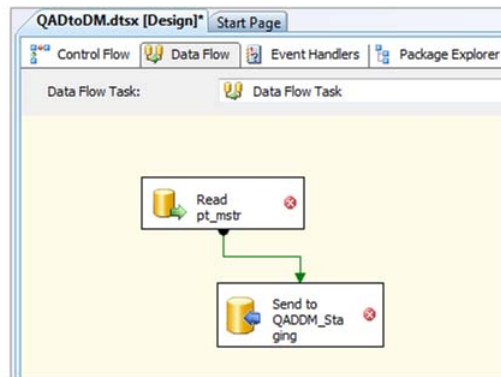
- The various pieces of the Data Flow Task must be connected together
- Drag the green arrow of the Source until it connects with the Destination piece



If you see a red circular “X” on any of the Data Flow components, this indicates that an error exists that will prevent the execution of this task.

Building a Data Flow Task

- You may wish to rename the components of the Data Flow Task

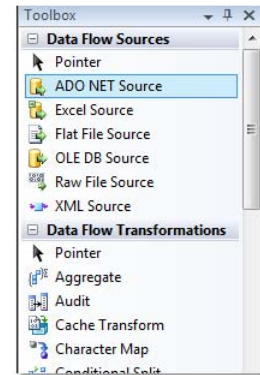


If you see a red circular “X” on any of the Data Flow components, this indicates that an error exists that will prevent the execution of this task.

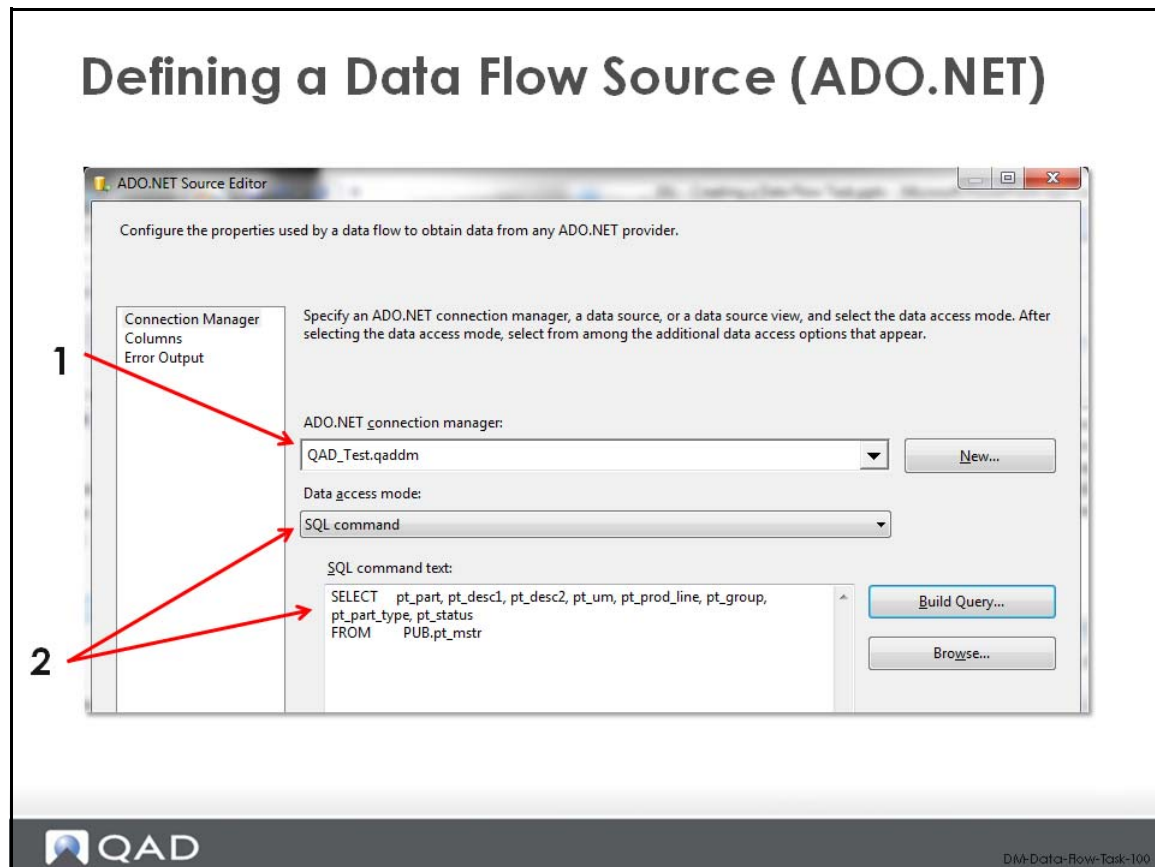
Defining a Data Flow Source

Defining a Data Flow Source

- A Data Flow Source defines the source data involved in the Data Flow
- Data Flow Sources can be of a variety of types
 - ADO.NET
 - Excel
 - Flat File
 - OLE DB
- When extracting data from a QAD database, it is recommended that you utilize an ADO.NET source



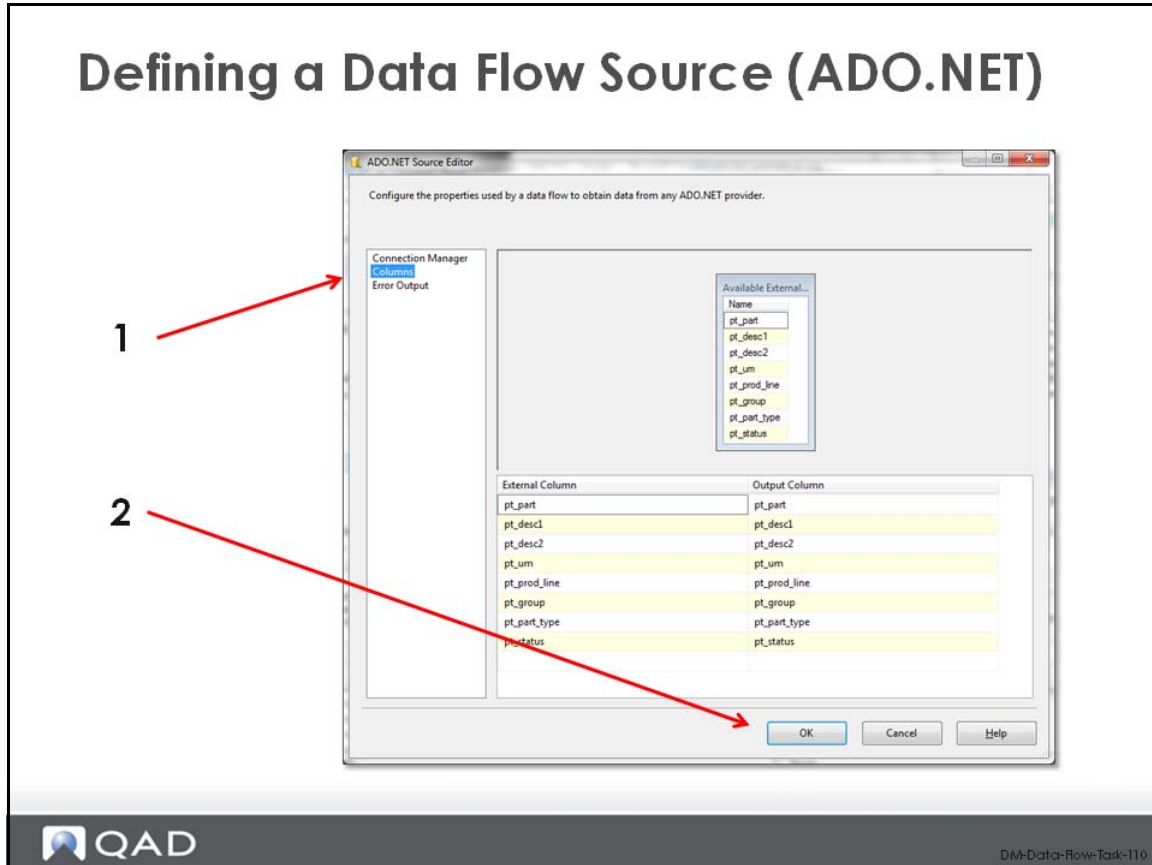
Defining a Data Flow Source (ADO.NET)



When double-clicking on the Data Source component, follow this procedure:

- 1 Select the appropriate connection manager that represents your connection to the data source.
- 2 Select SQL command in the “Data access mode” drop-down. Enter the appropriate SQL command in the “SQL command text” box.

Defining a Data Flow Source (ADO.NET)



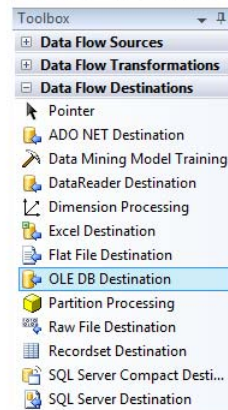
Check the validity of the connection by following this procedure:

- 1 Click the Columns option in the left window. You should see the fields from your query represented in the “External Column”. These fields should be duplicated in the “Output Column”, which represents the “Output” of the data query.
- 2 Click OK to complete the Data Source definition.

Defining a Data Flow Destination

Defining a Data Flow Destination

- A Data Flow Destination defines the destination of the data involved in the Data Flow
- Data Flow Destinations can be of a variety of types
 - ADO.NET
 - Excel
 - Flat File
 - OLE DB
- When sending data to a SQL database, it is recommended that you use an OLE DB destination rather than a SQL Server Destination due to performance



DM-Data-Flow-Task-120

The reasons for choosing an OLE DB destination rather than a SQL Server destination when sending data to a SQL database can be found in many publications on the internet. The OLE DB destinations tends to perform equally in terms of performance but has much better error reporting, thus becoming a preference for developers.

Adding a Table to QADDM_Staging

Adding a Table to QADDM_Staging

- In order to define a Data Destination, you must have a table defined in your QADDM_Staging database to house the data
- Use SQL Server Management Studio to create a destination table
- Create one table per Data Flow Task
- If possible, name the fields in the destination table with the same field names as used in the source table

Adding a Table to QADDM_Staging

- The example below shows the creation of a table to house a **part master-related** data extract

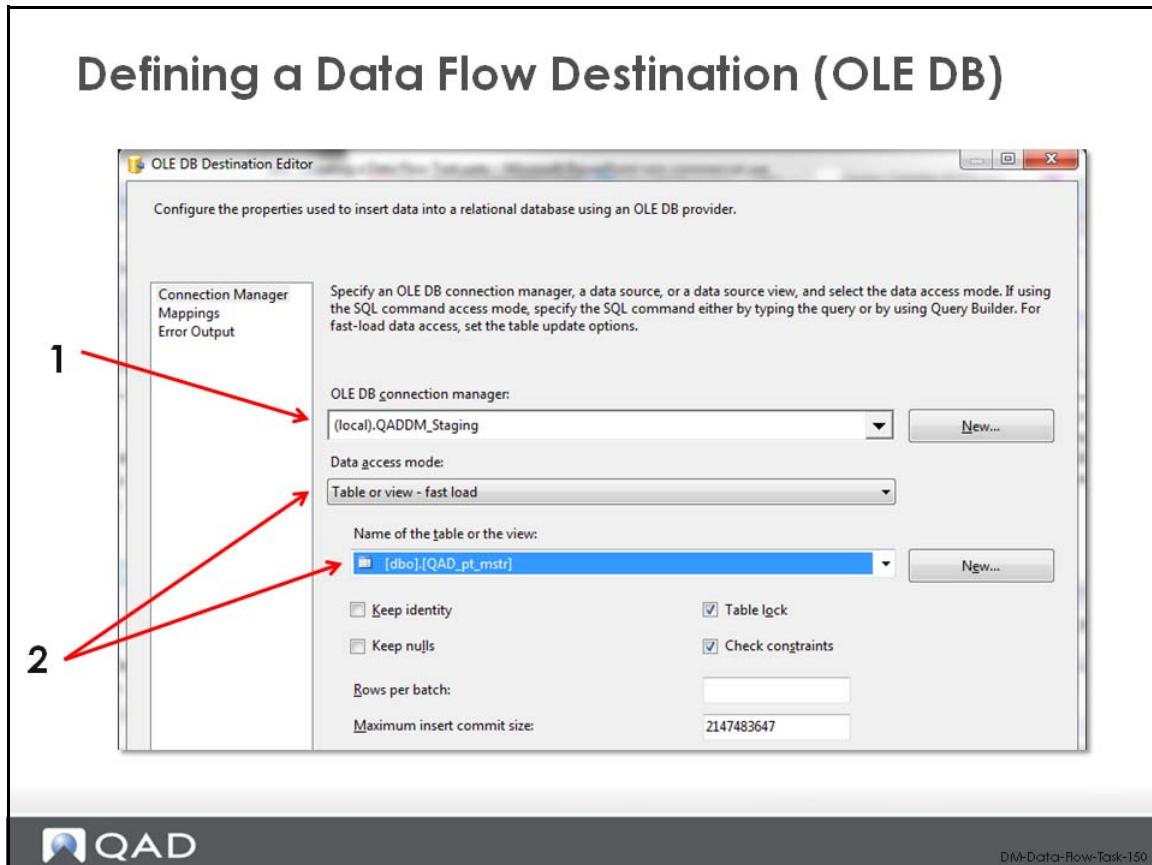
The screenshot illustrates the process of creating a new table in SQL Server. On the left, the Object Explorer shows the 'LEIF-DELL2' server with the 'QADDM_Staging' database selected. A red arrow labeled '1' points to the 'Tables' folder. On the right, the 'Table Designer' window for 'dbo.Table_1*' is open, showing a list of columns with their data types and 'Allow Nulls' status. A red arrow labeled '2' points to the column list. Below the table designer, a 'Choose Name' dialog box is open, with a red arrow labeled '3' pointing to the text input field containing 'QAD_pt_mstr'.

Column Name	Data Type	Allow Nulls
pt_part	nvarchar(50)	<input checked="" type="checkbox"/>
pt_desc1	nvarchar(50)	<input checked="" type="checkbox"/>
pt_desc2	nvarchar(50)	<input checked="" type="checkbox"/>
pt_um	nvarchar(50)	<input checked="" type="checkbox"/>
pt_prod_line	nvarchar(50)	<input checked="" type="checkbox"/>
pt_group	nvarchar(50)	<input checked="" type="checkbox"/>
pt_part_type	nvarchar(50)	<input checked="" type="checkbox"/>
pt_status	nvarchar(50)	<input checked="" type="checkbox"/>

To create a new table, follow this procedure:

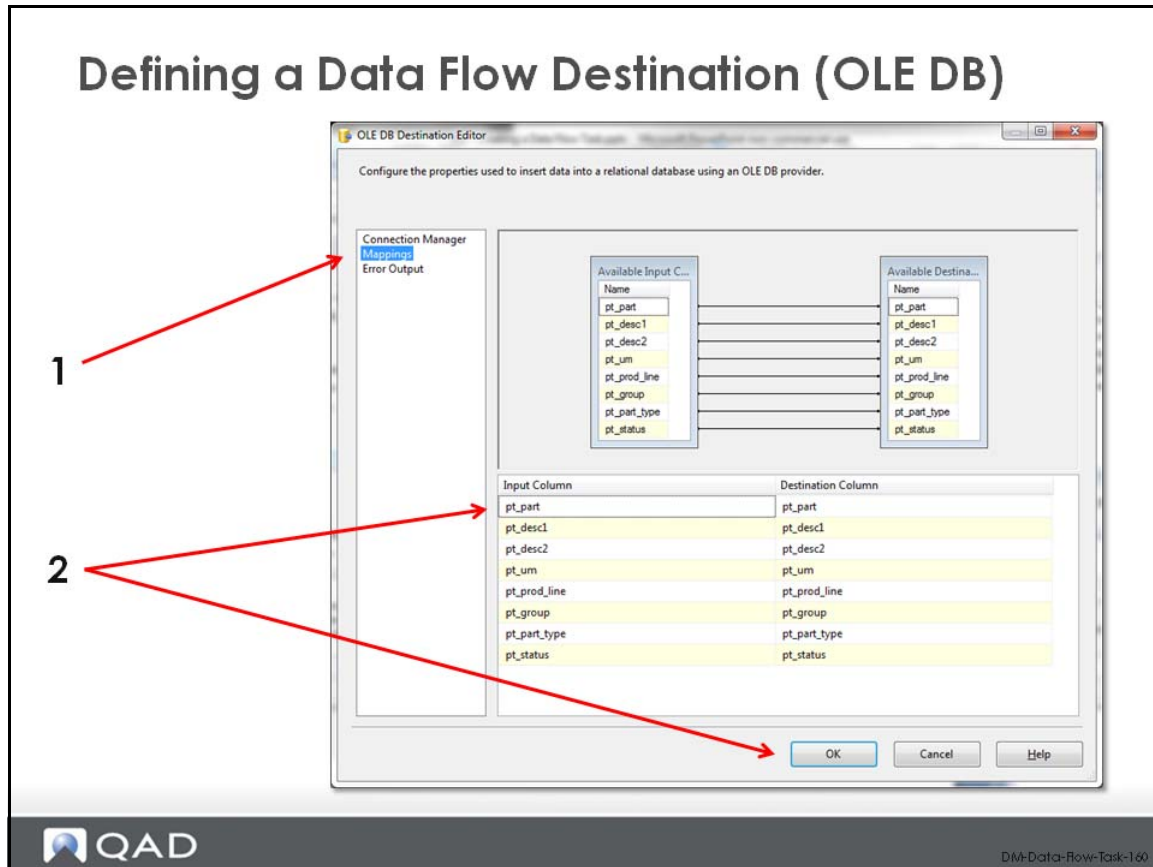
- 1 Open SQL Server Management Studio. Open the QADDM_Staging database. Right-click on the Tables folder and select New Table.
- 2 In the window that appears to the right of the Object Explorer, enter the column names and data types for the new table.
- 3 Select Save and give the table a name that reflects its contents.

Defining a Data Flow Destination (OLE DB)



When double-clicking on the Data Destination component, follow this procedure:

- 1 Select the appropriate connection manager that represents your connection to the data source.
- 2 Select Table or view – fast load in the “Data access mode” drop-down. Enter the name of the destination table in the “Name of the table” drop-down.



To continue defining the Data Destination component, follow this procedure:

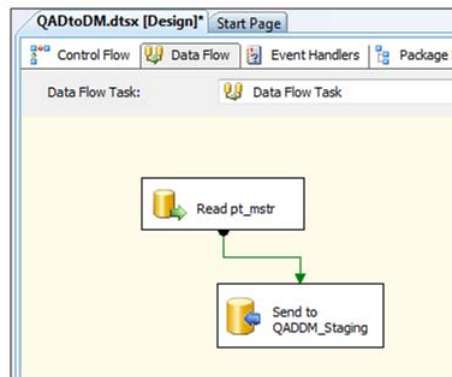
- 1 Click the Mappings option in the left window.
- 2 Ensure that the “Input Column”, which represents the data from your Data Source, map to the appropriate fields in the “Destination Column”, which represents the field names in your Data Destination. Click OK to complete this task.

As noted earlier, having field names in your SQL destination table that match the source data makes this mapping exercise much simpler.

Completion of the Data Flow Task

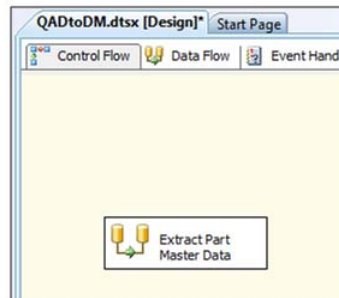
Completion of the Data Flow Task

- Look in the **Data Flow** window
 - If there are no circular red X's within either the Data Source or Data Destination components, your Data Flow has been successfully created and validated



Completion of the Data Flow Task

- Now switch to the **Control Flow** window
 - You should rename the "Data Flow Task" with a name that reflects the purpose of this task, such as "Extract Part Master Data"



Creating a Data Flow Task Review

Review

- At this point, you have created a single Data Flow Task
- During a typical DM implementation, you will create multiple Data Flow Tasks
 - Simply repeat the procedure outlined in this document
- Within the Control Flow window, multiple Data Flow Tasks can be connected together by dragging the green arrow from one task to another

Chapter 13

QAD-to-DM: Creating an Execute SQL Task

What is an Execute SQL Task?

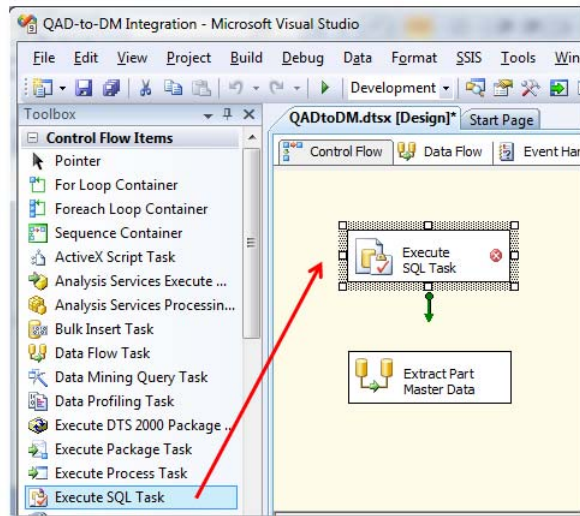
What is an Execute SQL Task?

- An Execute SQL Task allows for the execution of a SQL script to perform some data manipulation
- In the context of QAD-to-DM integration, this task allows for the manipulation of data transferred from a QAD database to the QADDM_Staging database
- At a minimum, a typical QAD-to-DM SSIS package will have at least one Execute SQL Task

Adding a Task to the Control Flow

Adding a Task to the Control Flow

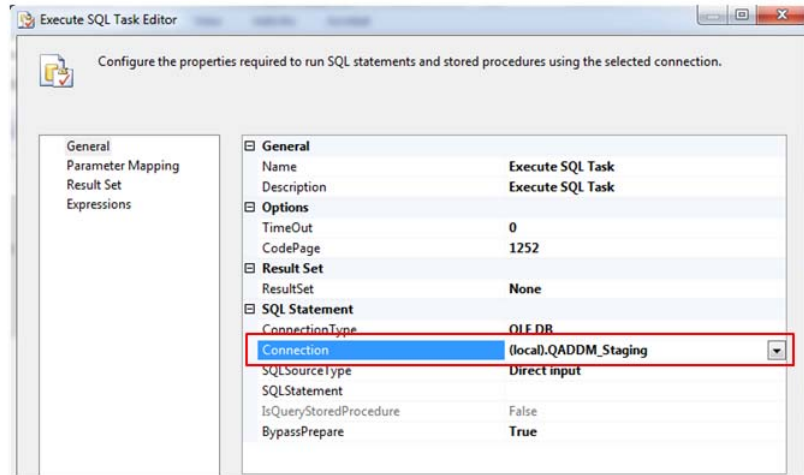
- To add a task to the Control Flow, simply highlight, drag and drop the task onto the Control Flow page
- In this example, an "Execute SQL Task" has been added to the Control Flow.



Building an Execute SQL Task

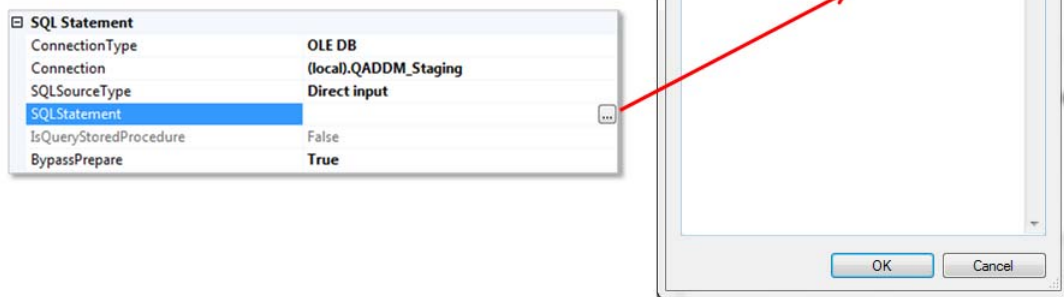
Building an Execute SQL Task

- Double-click the Execute SQL Task
 - Begin by specifying the **Connection** to the QADDM_Staging database



Building an Execute SQL Task

- The second step is to define the **SQL Statement** to be executed
- The example displayed here is a SQL script that clears the contents of our QADDM_Staging tables

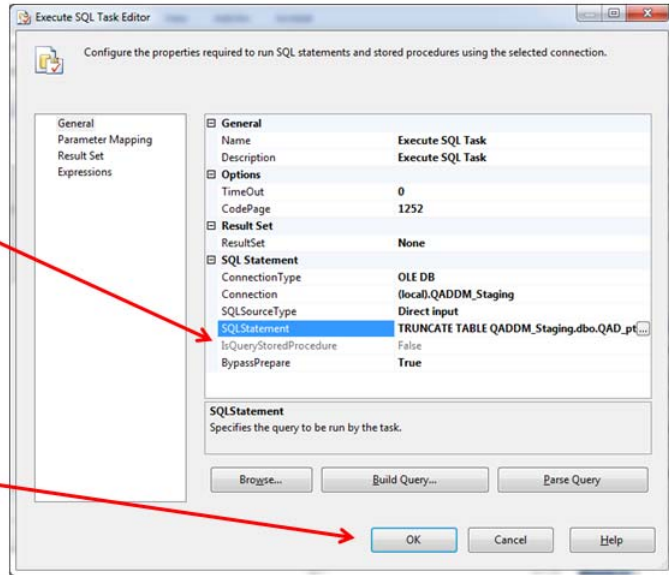


Building an Execute SQL Task

- Ensure that the **SQL Statement** is visible in the Editor window

1

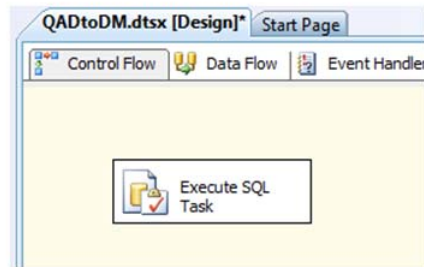
2



Completion of the Execute SQL Task

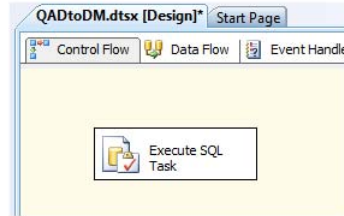
Completion of the Execute SQL Task

- Look in the **Control Flow** window
 - If there is not a circular red X within the task box your Execute SQL Task has been successfully created and validated

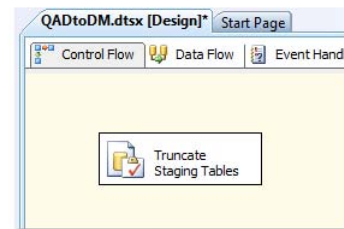


Completion of the Execute SQL Task

- Look in the **Control Flow** window
 - If there is not a circular red X within the task box, your Execute SQL Task has been successfully created and validated



- Rename your task with a name that reflects its purpose



Creating an Execute SQL Task Review

Review

- At this point, you have created a single Execute SQL Task
- During a typical DM implementation, you may create multiple Data Flow Tasks
 - Simply repeat the procedure outlined in this document
- Within the Control Flow window multiple Data Flow Tasks can be connected together by dragging the green arrow from one task to another

Chapter 14

QAD-to-DM: Building a Complete SSIS Package

Building a Complete SSIS Package

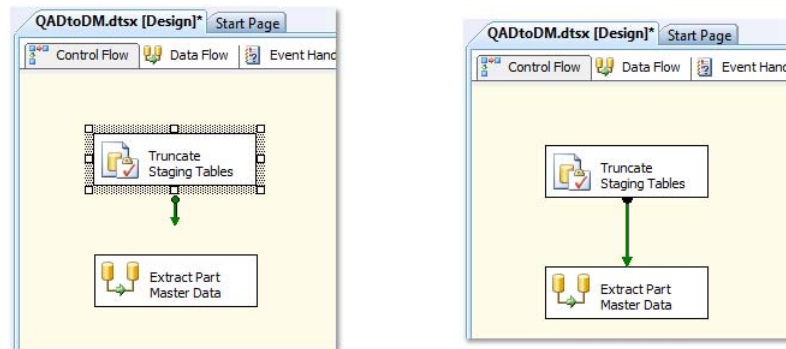
Building a Complete SSIS Package

- A typical SSIS package that supports QAD-to-DM integration will have multiple tasks
- The various tasks must be connected together in an appropriate sequence
- Once a package is completed various steps must be taken to make it operational
 - Testing of the package
 - Saving of the package
 - Installing the package into the SQL instance
 - Executing the package from a stored procedure/DM Custom Task

Connecting Multiple Tasks Together

Connecting Multiple Tasks Together

- A typical SSIS package will have multiple tasks
 - These tasks should be connected together in the sequence in which they should be processed



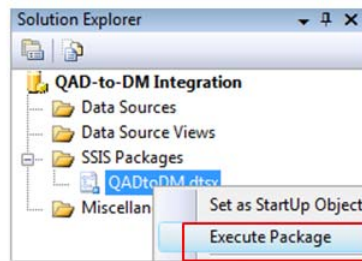
To connect tasks together, simply click-and-drag the green arrow from one task to another.

Testing the SSIS Package

Testing the SSIS Package

- As part of the development process, you will test the package before its deployment
- It is recommended that you save your work prior to testing
- The first task is to "execute" the SSIS package. Right-click on the package name in the Solution Explorer and select

Execute Package



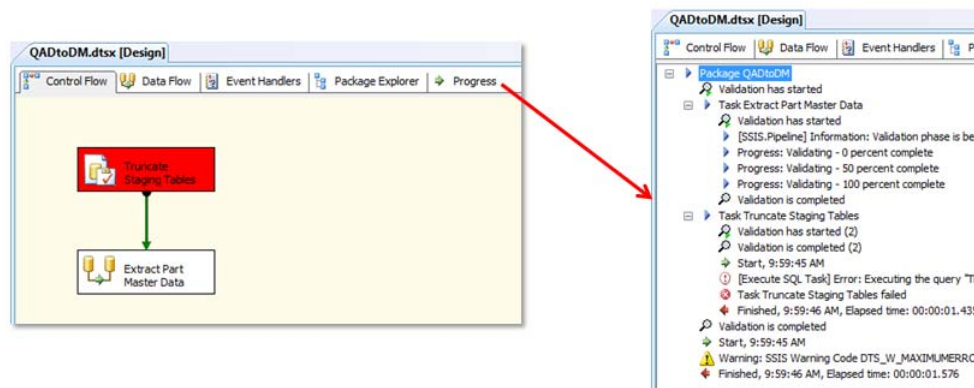
Testing the SSIS Package

- During the execution of the SSIS package you will see a color-coded display that reflects the status of a specific task
 - **Green** – this color represents the successful execution of a task
 - **Yellow** – this color represents that a task is currently being executed
 - **Red** – this color represents that an error occurred during this task

Examples of Package Execution

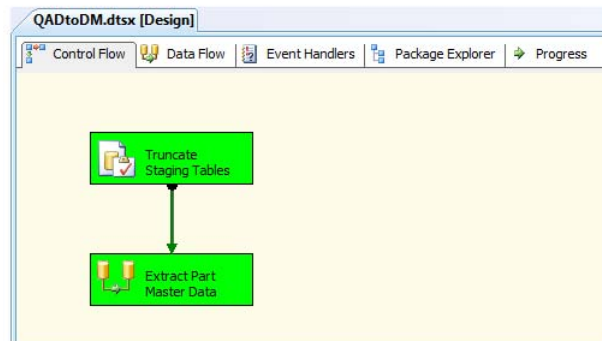
Examples of Package Execution

- In this example, an error occurred during the Truncate Staging Tables task
- The **Progress** tab can provide a detailed log related to the package execution



Examples of Package Execution

- In this example, all tasks are **green** representing successful execution
- The **Progress** tab can provide a detailed log related to the package execution



Installing the Package into the SQL Instance

Installing the Package into the SQL Instance

- Once the development and testing of the package is completed, the package must be installed into the SQL instance so that its execution can be automated
- Determine the full file path of the package (the file ending in .dtsx) as this will be required to install the package
- Open SQL Server Management Studio

Installing the Package into the SQL Instance

- Connect to **Integration Services**

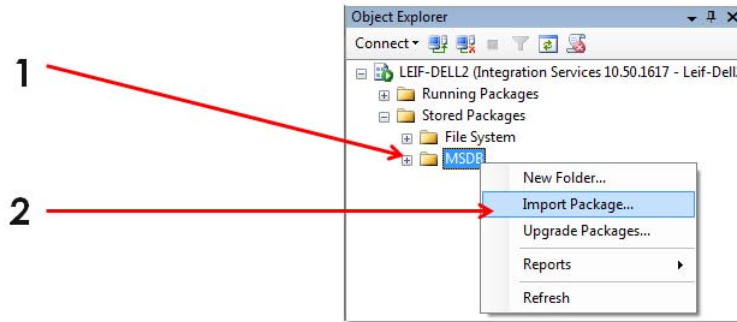


To connect to Integration Services, follow this procedure:

- 1 Select Integration Services from the “Server type” drop-down list.
- 2 Enter the appropriate server name and click “Connect”.

Installing the Package into the SQL Instance

- The first step will be to **Import Package**

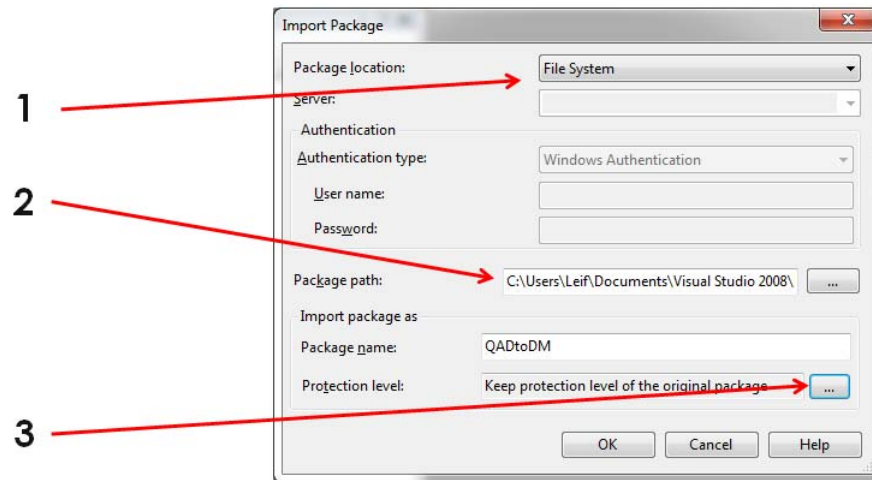


To import the package, follow this procedure:

- 1 Expand the Stored Packages folder.
- 2 Right-click on MSDB and select Import Package.

Installing the Package into the SQL Instance

- Enter the package information as shown in this example

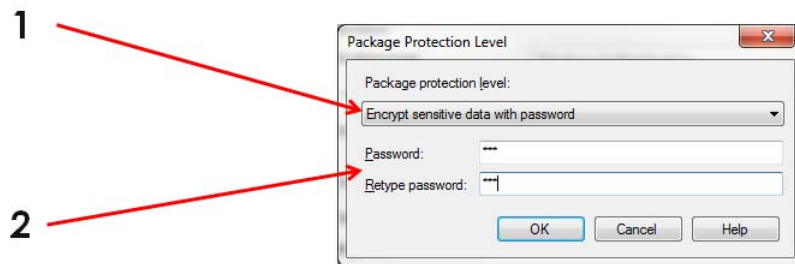


To import the package, follow this procedure:

- 1 Select File System in the “Package location” drop-down.
- 2 Specify the file path of the package. The “...” button can be used to locate the .dtsx file.
- 3 Click the “...” button to change the “Protection level”.

Installing the Package into the SQL Instance

- It is recommended that you protect the package with a password
 - Document this password as you will need it in a later step



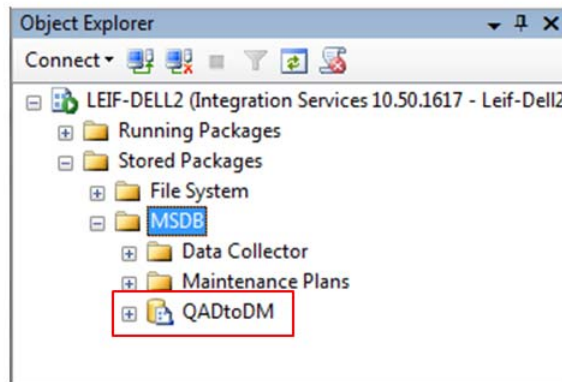
DM-Comp-S3IS-Package-120

To protect the package, follow this procedure:

- 1 Select Encrypt sensitive data with password in the “Package protection level” drop-down.
- 2 Enter a password in the “Password” and “Retype password” fields. Click OK when completed.

Installing the Package into the SQL Instance

- The package should now be visible within the **MSDB** sub-folder within the Stored Packages folder



Executing the SSIS Package from DM

Executing the SSIS Package from DM

- A goal of the technical implementation is to achieve the execution of the SSIS package from within Demand Management
- This is achieved by developing a **stored procedure** which executes the SSIS package
- This stored procedure will be registered as a **custom task** inside of DM
 - It can then be executed using the Batch Scheduler (see the Custom Tasks section for more details on this step)



DM-Comp-SSIS-Package-140

Refer to the section on Custom Tasks for more information about the creation of custom tasks.

Building the Stored Procedure

Building the Stored Procedure

- A recommended stored procedure for executing the SSIS package is outlined below
- This procedure can be modified depending on the needs of the specific implementation

```
CREATE PROCEDURE [dbo].[USP_QAD_RunImportPackage]
@ScnID Int,
@UserID Int
AS

-- This custom tasks executes the SSIS package that extracts and transforms
data from QAD Enterprise Apps

DECLARE @return_value int
EXEC @return_value = xp_cmdshell 'C:\DTExec.exe /sql QADtoDM /de "123"'
```



DM-Comp-SSIS-Package-150

The procedure outlined in this example assumes that the package was named QAD-to-DM and a password of 123 was used when the package was installed into the SQL instance.

Building a Complete SSIS Package Review

Review

- At this point you have created an SSIS package that transfers data from QAD into the QADDM_Staging database
- This package has been installed into the SQL instance
- A stored procedure has been developed to allow for the execution of the package
- A custom task has been created to allow for the execution of the package from the Batch Scheduler



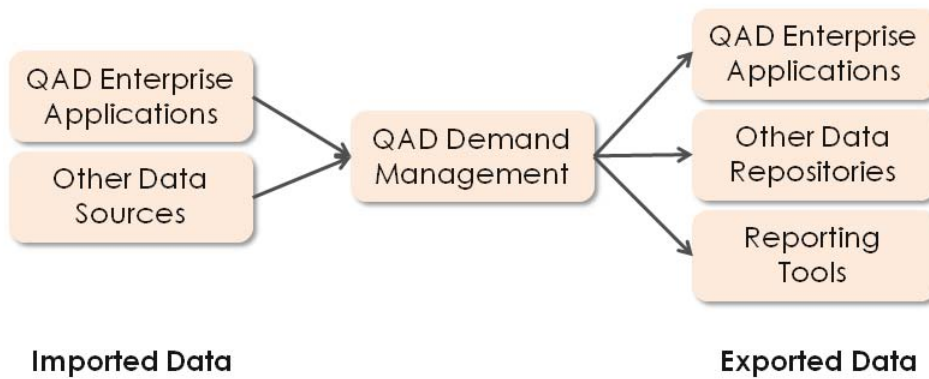
Chapter 15

DM-to-QAD Integration Preparation

Integration Overview

Integration Overview

- The Demand Management application is essentially a stand-alone application
- It was not developed to integrate with any specific ERP application



Preparation Overview

Preparation Overview

- What data is required to send from DM to QAD?
- How does data flow from DM to QAD?
- Does the DM Living Master scenario utilize something other than weekly time buckets?
- What mechanism should be used to transfer data from the DM (SQL) server to the QAD server?

Data to be Exported

Data to be Exported

- Most DM implementations require that a **forecast** be transferred from DM to QAD in order to support MRP
- The ultimate destination of the DM forecast inside of QAD is **Forecast Maintenance**
- The Forecast Maintenance functionality requires that a forecast is defined by
 - Item (Part Number)
 - Site
 - Week (beginning on a Monday)



DM-to-QAD-Integ-Prep-040

It is possible to export data other than the forecast to QAD. For example, some users of DM export safety stock numbers for use in QAD. With any data extract involving data going to a destination other than Forecast Maintenance, the extract of data will be a custom effort (potentially on both the DM and QAD sides).

Overview - Flow of Data From DM to QAD

Overview - Flow of Data From DM to QAD

- There is not a direct connection from the DM database to the QAD database
- A stored procedure (custom task) within the DM database generates a flat file contained the summarized forecast data
- The flat file is transferred from the DM server to the QAD server
- On the QAD server, the flat file is imported using the Demand Planner Import function (36.5.9.2)
- The data is then visible within Forecast Maintenance



DM-to-QAD-Integ-Prep-050

It is important to realize that the exporting of the forecast from DM to QAD involves more than one step. There is NOT a direct connection between the two databases. In the next Powerpoint, these topics are outlined in more detail.

Time Buckets

Time Buckets

- As mentioned earlier, the forecast export file will contain data in **weekly** time buckets
- If the DM Living Master scenario contains something other than weekly time buckets a **calendar scenario** must be created to define the splitting of data into weekly buckets
- If the Living Master is in monthly buckets, for example, the calendar scenario will outline the percentage of a month allocated to each week
 - A week may receive a forecast from two months (for weeks that straddle two months)



DM-to-QAD-Integ-Prep-060

It is important to realize that the exporting of the forecast from DM to QAD involves more than one step. There is NOT a direct connection between the two databases. In the next Powerpoint, these topics are outlined in more detail.

Connecting the DM Server to QAD

Connecting the DM Server to QAD

- The passing of a flat file from the DM server to the QAD server requires a mechanism to enable the file transfer
- Commonly used methods
 - **FTP** – this file transfer method is relatively simple and is used by many customers
 - It can be incorporated into the export custom task or can be scheduled via other mechanisms
 - Secure FTP (SFTP) is a common variation
 - **Linked Directories** – this file transfer method involves having a QAD server file path visible within the Windows Server
 - The flat file is then placed into this linked directory



DM-to-QAD-Integ-Prep-070

Consult with your customer to determine which file transfer method will work best in their environment.

Chapter 16

DM-to-QAD: Creating the DM Export Tables

DM Export Tables

DM Export Tables

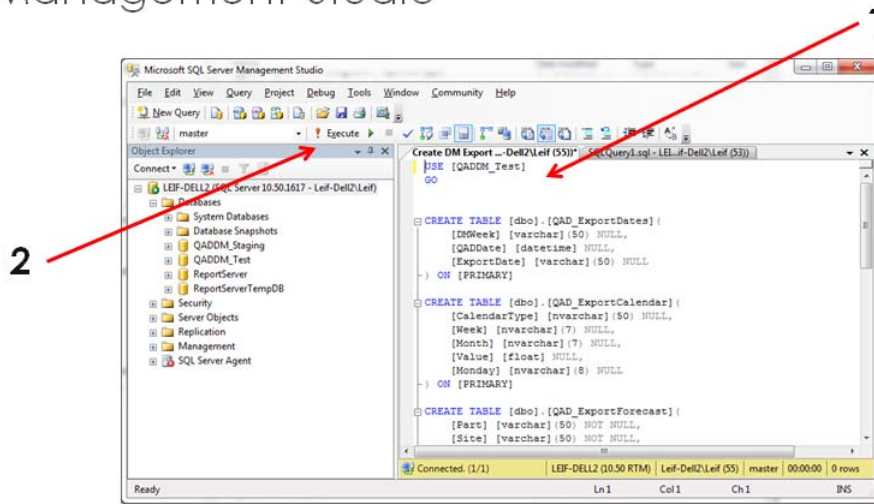
- The Demand Management export tables are custom tables added to the DM database to enable the forecast export process:
 - **QAD_ExportDates** – defines the specific starting date of a week
 - **QAD_ExportCalendar** – this table houses a “friendly version” of the calendar scenario
 - **QAD_ExportForecast** - houses the output of the forecast export custom task
 - **QAD_ExportParameters** – allows for the setting of forecast export parameters



Creating the DM Export Tables

Creating the DM Export Tables

- Obtain the “Create DM Export Tables.sql” script and open using SQL Server Management Studio



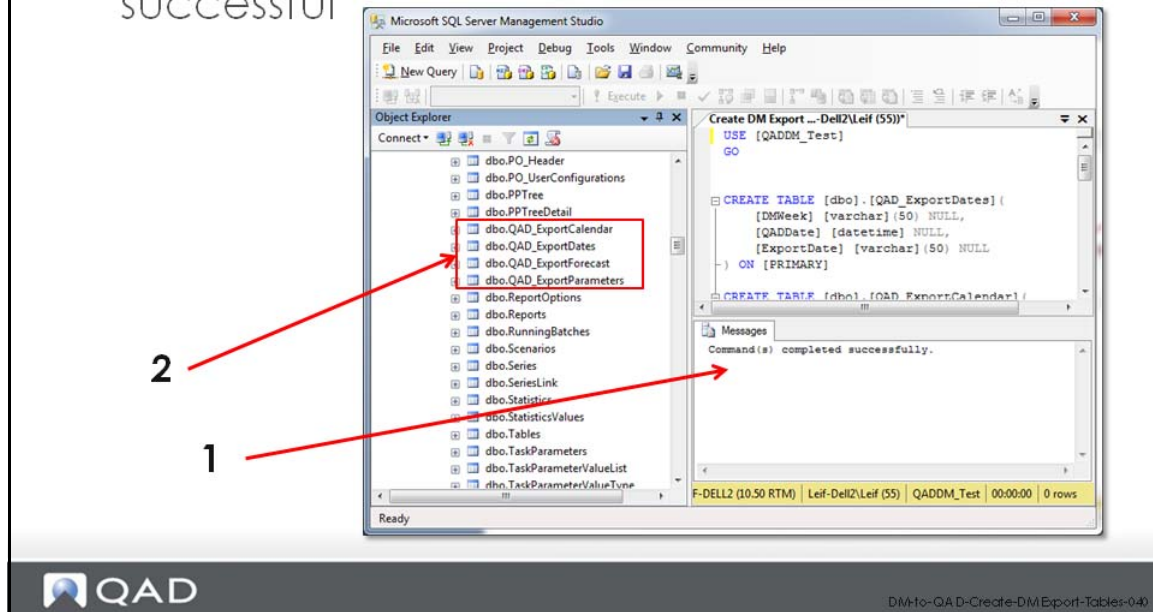
DM-to-QAD-D-Create-DM-Export-Tables-030

To create the DM export tables, follow this procedure:

- 1 Open the “Create DM Export Tables.sql” script using SQL Server Management Studio. In the “USE” statement of the script, ensure the correct DM database name is listed.
- 2 Click the ! Execute button to run the script.

Creating the DM Export Tables

- Review the **Messages** section and the **Tables** listing to see if the table creation was successful



The “Command(s) completed successfully.” message indicates that the script ran without errors. By expanding the Tables folder under the QADDM database, you should see the four QAD_Export tables.

Populating QAD_ExportDates

Populating QAD_ExportDates

- The QAD_ExportDates table is meant to define the specific relationship between a DM week (which is expressed as YYYY-WW) and a specific date
- The primary purpose of this table is to help manage the occasional 53 week year
 - In DM a year can only contain 52 weeks
 - This table allows for the proper mapping of the extra week
- The data in this table can be maintained and customized by the customer according to their requirements



DM-to-QAD-Create-DM-Export-Tables-050

It is possible to export data other than the forecast to QAD. For example, some users of DM export safety stock numbers for use in QAD. With any data extract involving data going to a destination other than Forecast Maintenance, the extract of data will be a custom effort (potentially on both the DM and QAD sides).

Populating QAD_ExportDates

- During the implementation of DM, it is recommended to populate this table with a standard dataset
- An Excel file named "QAD_ExportDates data.xlsx" has been created for this purpose
- Use the "Import/Export Wizard" to easily import this spreadsheet into the QAD_ExportDates table



Next Steps

Next Steps

- At this point the table infrastructure is in place to support forecast exports
- In the next session you will
 - Configure the QAD_ExportParameters table
 - Create/modify the forecast export stored procedure

Chapter 17

DM-to-QAD: Building the DM Export Procedure

DM Export Procedure

DM Export Procedure

- Building the DM forecast export procedure involves several key steps
 - **Export Parameters** – defines variables that drive the contents of the forecast export file
 - **Export Stored Procedure** – this procedure summarizes the forecast data and creates the output file
 - **Custom Task Creation** - the stored procedure must be installed as a custom task inside of DM to be utilized by the Batch Scheduler



DM-to-QAD-Build-DM-Export-020

Note that the DM forecast output file must be named “dmtrfcst.csv”. This is not a requirement of DM but of the QAD menu function that imports the forecast file.

Defining the Export Parameters

Defining the Export Parameters

- The QAD_ExportParameters table houses variables used in the export process

Field Name	Purpose	Required?
WorkDir	Defines the file directory on the DM server for the output file	No
FileName	The desired file name for the output file	No
HostServer	The QAD server name/address	No
HostDir	The directory on the QAD server for the output file	No
HostUserName	The user name to connect to the QAD server	No
HostPassword	The password to connect to the QAD server	No
FTPScript	A file name representing an FTP script to be used in the export process	No
FTPLog	A location for the FTP log file	No
WholeMonth	0 = start export from current week 1 = start export from beginning of month	Yes
TimeFence	Number of weeks from current week to skip in the forecast export file	Yes
Nperiods	Number of weeks that should be included in the forecast export file	Yes



DM-to-QAD-build-DM-Export-030

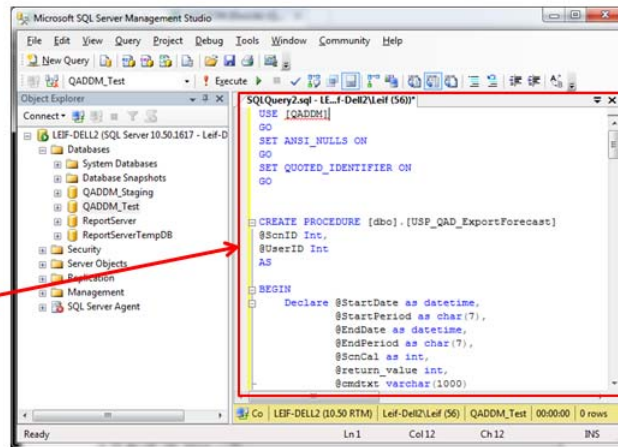
The only fields required are the last three: Whole Month, TimeFence and Nperiods. The other fields can be utilized if they are incorporated in the forecast export stored procedure. To populate this table, follow this procedure:

- 1 Open the table using SQL Server Management Studio.
- 2 If no records exist in this table, add one record and populate the fields as necessary. If a record already exists, update the fields as necessary.

Creating the Forecast Export Procedure

Creating the Forecast Export Procedure

- Use the "Forecast Export Stored Procedure.sql" script as a starting point



DM-10-QAD-build-DM-Export-040

Open the "Forecast Export Stored Procedure.sql" script in SQL Server Management Studio.

Creating the Forecast Export Procedure

- Review the section of the script with the "Calculation of Export Data" heading
- Determine if the first two fields in the SELECT statement reflect the "item" and "site" fields from the DM Series table (if not, modify the statement to match your configuration)

1

```

-- Calculation of export data
Insert Into QAD ExportForecast
Select trim(SR.Sku)
      ,SR.Other
      ,CAL.Monday
      ,Round(Sum(isnull(OB.UserValue04,0) * (CAL.Value / 100)), 0)
From QAD_ExportCalendar CAL Inner Join
     Observations OB On OB.Description = CAL.Month Inner Join
     Series SR With (RowLock) ON SR.SeriesID = OB.SeriesID
Where SR.ScenarioID = @ScnID
Group by SR.Sku, SR.Other, CAL.Monday
Order by SR.Sku, SR.Other, CAL.Monday

```



DM-to-QAD-Build-DM-Export-050

In the "Calculation of export data" section of the script, follow this procedure:

- 1 Determine if the first two fields of the SELECT statement match the fields for "item" and "site" from the Series table of the DM database. In this example, the SKU field represents "Item" and the Other field represents "Site".
- 2 Modify the calculation line if necessary to utilize a different observations field (opinion line).

Creating the Forecast Export Procedure

- Review the formula that begins with "Round"
 - Determine what observations field (opinion line) you require as part of the forecast export
- The example below shows the use of the "UserValue04" field as the "forecast value"

2

```

-- Calculation of export data
Insert Into QAD_ExportForecast
Select rtrim(SR.Sku)
, SR.Other
, CAL.Monday
, Round(Sum(Isnull(OB.UserValue04, 0) * (CAL.Value / 100)), 0)
From QAD_ExportCalendar CAL Inner Join
Observations OB On OB.Description = CAL.Month Inner Join
Series SR With (RowLock) ON SR.SeriesID = OB.SeriesID
Where SR.ScenarioID = @ScnID
Group by SR.Sku, SR.Other, CAL.Monday
Order by SR.Sku, SR.Other, CAL.Monday

```



DM-10-QAD-Build-DM-Export-060

In the "Calculation of export data" section of the script, follow this procedure:

- 1 Determine if the first two fields of the SELECT statement match the fields for "item" and "site" from the Series table of the DM database. In this example, the SKU field represents "Item" and the Other field represents "Site".
- 2 Modify the calculation line if necessary to utilize a different observations field (opinion line).

Creating the Forecast Export Procedure

- Review the section of the script with the "Creation of .csv file" heading
- The creation of the forecast export file can be done in a variety of ways
 - **BCP function** – dumps the contents of a table into a text file
 - **SSIS package** – performs an export into a text file

```

1  -- Creation of .csv file
   /* Two methods exist: BCP or SSIS package. Pick one method that best works
   for your customer. */
   exec master.dbo.xp_cmdshell 'bcp qaddm.dbo.qad_forecastexport out
   "D:\QAD_Databases\ForecastExport\dmtrfcst.csv" -T -t, -C ACP -w'
2  --exec master.dbo.xp_cmdshell 'dtexec /sql ExportForecastCSV'
   /*
  
```

The export of a flat file can occur through several methods. Here are two common examples:

- 1 Use of the BCP function. This dumps the contents of a table into a text file. Be sure to specify the source table and destination file name.
- 2 Build a SSIS package that exports the contents of the QAD_ForecastExport table to a text file. The name of the package would be used in this "dtexec" statement (in the example, the package is named "ExportForecastCSV").

Additional Considerations

Additional Considerations

- Depending on the method used to transfer the forecast export file to the QAD server, you may add that functionality to the forecast export stored procedure
- The forecast export file should be named "dmtrfcst.csv" – this is a requirement of QAD
- You may add additional functionality to the forecast export process depending on your customer's business requirements

Next Steps

Next Steps

- At this point, DM has been configured to allow for the creation of a forecast export file
- As part of your DM implementation, you may explore these next steps
 - Setup of the Batch Scheduler to execute a forecast export custom task
 - Setup of the QAD Demand Planning import functions (36.5.9)



DM-to-QAD-Build-DM-Export-100

The QAD Demand Planning import/export functions (which are accessed via menu function 36.5.9) are fully documented in a separate workaid.

Chapter 18

Developing Custom Tasks

Custom Task Overview

Custom Task Overview

- A custom task is a SQL stored procedure in the DME database that provides customized functionality that does not exist in the standard DME package
- Custom tasks are designed to manipulate data contained within the DME database or other data repository based upon business rules defined by the implementation team
- Examples include:
 - Opinionline A * Series Attribute Price = Revenue
 - Copy Opinion Line A to Opinion Line B

Stored Procedure Requirements

Stored Procedure Requirements

- In order to be called by a custom task, a stored procedure should satisfy the following conditions
 - It is stored within the DME database
 - Has "execute" permissions granted
 - Has the same owner as the DME database
 - The first two parameters are "INTEGER" type
- Stored procedures can be developed using SQL Server Management Studio

Stored Procedure Parameters

Stored Procedure Parameters

- A custom task can have an unlimited number of parameters - there are three special parameters
 - **@ScenarioID** – the DME will pass the current ScenarioID to this parameter
 - It is required in all custom tasks and should be the first parameter listed
 - **@UserID** – the DME passes the current UserID to this parameter
 - It is also required in all custom tasks and should be listed second
 - **@SeriesID** – this optional parameter passes the current SeriesID if executed from the Viewer
 - If called from the Batch Scheduler, a zero is passed to this parameter

Pre-Installed Stored Procedures

Pre-Installed Stored Procedures

- A number of stored procedures designed to execute certain Viewer functions are part of the standard installation

Stored Procedure	Purpose
USP_JGS_ApplyHierarchy_CustomTask	Applies a static hierarchy to the specified scenario
USP_JGS_CopyHierarchy_CustomTask	Copies a hierarchy definition to another user
USP_JGS_DeleteTree_CustomTask	Removes the static hierarchy from the specified scenario
USP_JGS_Rebalance_CustomTask	Performs the "rebalance" function to the specified scenario
USP_JGS_Reforecast_CustomTask	Performs the "reforecast" function to the specified scenario
USP_JGS_Reindex_CustomTask	Reindexes the DME database



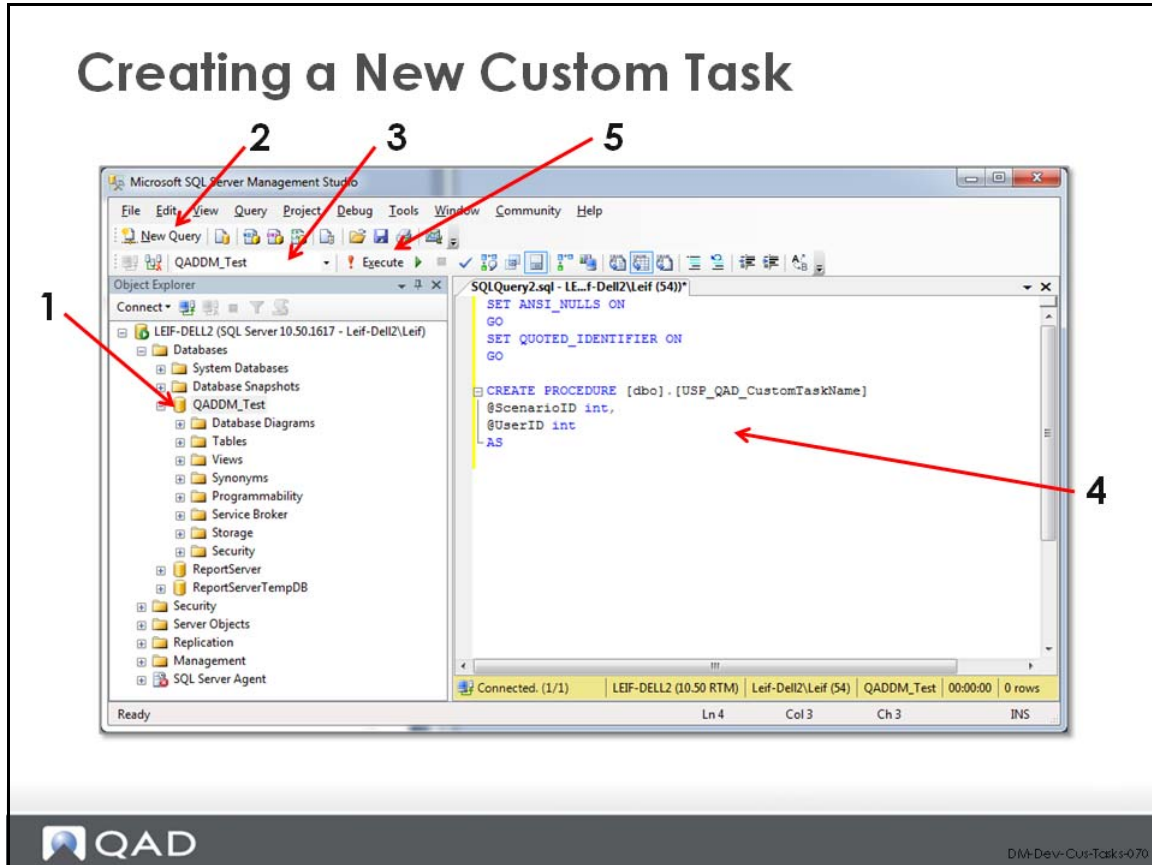
DW-Dev-Cur-Task-050

This list is not meant to be all-inclusive. Generally speaking, a custom task exists for most Viewer functions.

Creating a new Custom Task

Creating a New Custom Task

- Outline the purpose of the custom task
- Define the specific fields and/or calculations that will be a part of the custom task
- Open SQL Server Management Studio and click "New Query"
- Develop the SQL procedure and add it to the DME database
- Within the DME Viewer or Admin Tool, add the stored procedure as a custom task



Once you open SQL Server Management Studio, follow this procedure to create a new stored procedure:

- 1 Highlight your DME database.
- 2 Click the “New Query” button.
- 3 If the database drop-down list doesn’t display the name of your DME database, select it.
- 4 Build your SQL procedure within the SQL Query window.
- 5 To store the new procedure within the SQL database, click “! Execute”

Custom Task Example

Custom Task Example

- In this example, we will develop a custom task that will populate an opinion line with the following calculation
 - $U\text{value}01 = F\text{value} * \text{Price}$
- To simplify the stored procedure creation, start with this "stored procedure" template

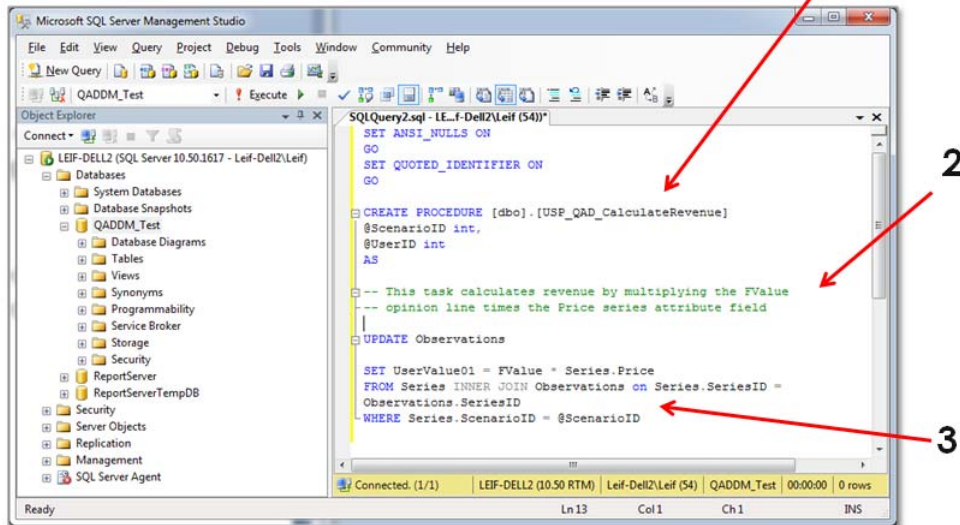
```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[USP_QAD_CustomTaskName]
@ScenarioID int,
@UserID int
AS

-- Custom task code here:
```

Custom Task Example

- Build the stored procedure

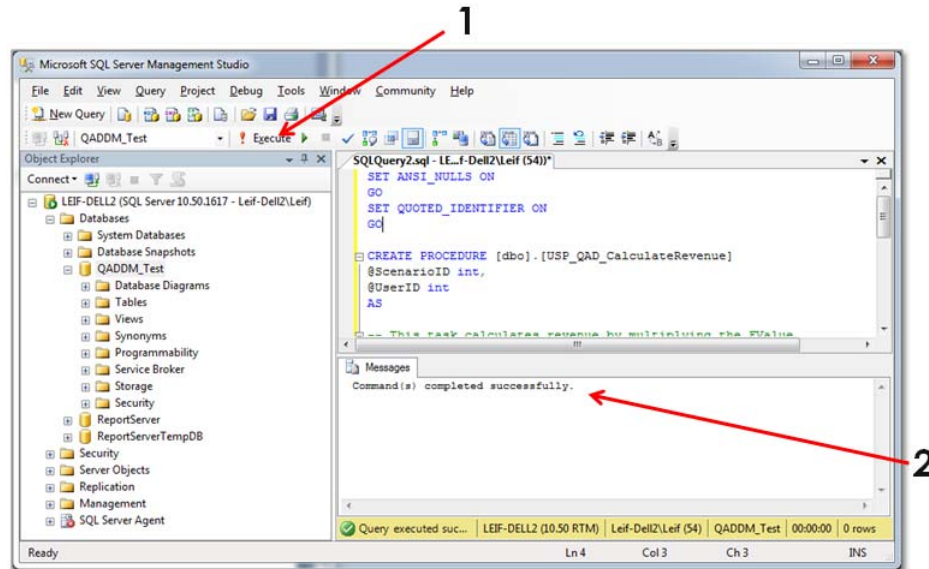


The process for building the stored procedure:

- 1 We give the stored procedure a unique name. In this example, we have named the procedure USP_QAD_CalculateRevenue (USP stands for user-stored procedure).
- 2 We have added notes to describe the purpose of the procedure. Notes in a SQL procedure begin with "--"
- 3 Enter the query that will be used to perform the desired actions.

Custom Task Example

- Execute the stored procedure

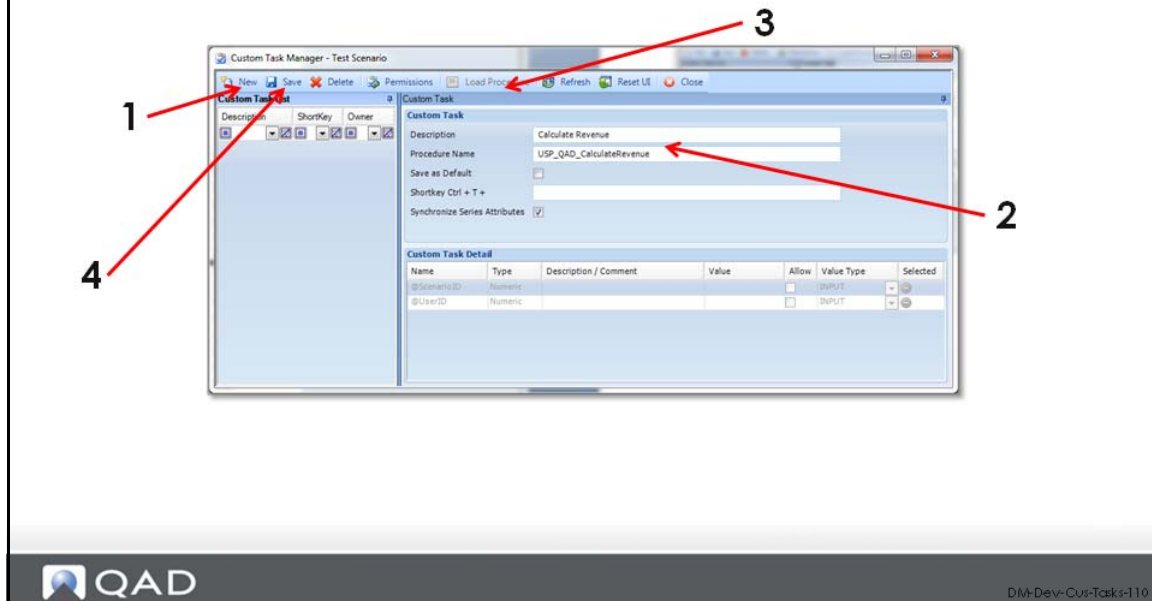


To store the newly created procedure, follow this process:

- 1 Click the “Execute” button. This creates the procedure within the database.
- 2 Look for a “Command(s) completed successfully” message in the messages window. If there were any errors within your procedure, they will be outlined in this window.

Custom Task Example

- Create the custom task within the DME Viewer or Admin Tool



To create a new custom task based on the new stored procedure, open either the DME Viewer or DME Admin Tool, navigate to the Custom Task Manager and follow this process:

- 1 Click “New”.
- 2 Enter a name for the custom task in the “Description” field. Enter the stored procedure name in the “Procedure Name” field.
- 3 Click the “Load Procedure” button. This will populate the parameter list within the “Custom Task Detail” section of the window.
- 4 Click “Save”.

