



User Guide

QAD QXtend Excelerator

QAD QXtend Excelerator Overview
Implementing QAD QXtend Excelerator
Using QAD QXtend Excelerator

70-3191A
Version 1.2
September 2011

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 ©2011 by QAD Inc.

Excelerator_UG_v0102.pdf/hes/hes

QAD Inc.

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

Contents

What's New?	v
About This Guide	1
What Is in This Guide?	2
Audience	2
Related Documentation	2
QAD QXtend Documentation	2
Other Documentation	2
Conventions	3
Menu References	3
Typographic Conventions	3
QAD Support Services	4
Chapter 1 QAD QXtend Excelerator Overview	5
Introduction	6
Data Integration Process Map	8
Chapter 2 Implementing QAD QXtend Excelerator	9
Overview	10
Installing Excelerator and Prerequisite Components	10
Setting Up QXtend Outbound Query Service	11
Setting Up Authentication for Query QAD Requests	12
Configuring Excel Settings	12
Configuring Excelerator Settings	15
Designing Data Integration Worksheet Templates	18
Modifying Worksheet Templates	21
Working with CSV Files	21
Creating CSV Templates	21
Importing and Exporting CSV Files	21
Performing Initial Data Load	22
Performing Data Conversion	22
Upgrading Excelerator	23

Chapter 3	Using QAD QXtend Excelerator	25
Overview		26
Querying Data from the Source Application		26
Troubleshooting Data Queries		27
Viewing and Modifying Data		28
Changing Workspace		30
Updating Data to the Target QAD Application		30
Index		33

What's New?

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

Date/Version	Description	Reference
September 2011/1.2	Modified the introductory text to Excelerator	page 6
September 2011/1.2	Updated Excelerator's prerequisite components	page 10
September 2011/1.2	Included steps to show the Developer tab in Excel 2010	page 12
September 2011/1.2	Added instructions on upgrading Excelerator	page 23
September 2011/1.2	Updated the section on configuring Excelerator settings to include new features such as Enable Export and worksheet template type settings	page 15
September 2011/1.2	Updated the steps for designing data integration worksheets to include new worksheet template type settings	page 18
September 2011/1.2	Updated the section on working with CSV files	page 21
September 2011/1.2	Updated the section on troubleshooting data queries	page 27

About This Guide

What Is in This Guide? 2

Related Documentation 2

Conventions 3

QAD Support Services 4

What Is in This Guide?

Use this guide to configure and use QAD QXtend Excelsior, which is an optional data integration tool within the QAD QXtend interoperability framework that provides additional data visibility and manageability to the data integration process between QAD products.

For information about QAD QXtend, see *User Guide: QAD QXtend*.

Audience

This guide is intended for developers who design data integration worksheets and configure QAD QXtend Excelsior settings, as well as end users who perform actual data integration activities, including retrieving data from the source QAD application, editing data in the data integration worksheet, and synchronizing data with the target QAD application.

The developer should be familiar with the QAD Enterprise Applications and QXtend configurations. The user should have experience working with Microsoft Excel worksheets.

If you do not have this expertise within your company, contact QAD Support for information on the customization offerings supplied by QAD's Global Services.

Related Documentation

QAD QXtend Documentation

QAD QXtend Excelsior is designed to be used in conjunction with QAD QXtend to streamline the data integration and communication process.

- For instructions on installing QAD QXtend, see *Installation Guide: QAD QXtend*.
- For information on configuring and using QAD QXtend, see *User Guide: QAD QXtend*.

Other Documentation

QAD QXtend Excelsior is designed to interoperate with QAD Enterprise Applications.

For information on installing QAD Enterprise Applications or converting to a more recent release, refer to the appropriate installation guide for your system or the appropriate QAD Enterprise Applications conversion guide.

For information on using the various modules of QAD Enterprise Applications, refer to the user guides for your version.

For users with a QAD Web account, product documentation is available for viewing or downloading from the QAD Online Support Center at:

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

You can register for a QAD Web account by accessing the Web site. Your customer ID number is required. Access to certain areas is dependent on the type of agreement you have with QAD.

Most user documentation is available in two formats:

- Portable document format (PDF) files can be downloaded from the QAD Web site to your computer. You can view and print them with the free Adobe Acrobat Reader.
- HTML files let you view user documentation through your Web browser and use search tools for easily locating topics of interest.

Features of the Web site include an online solution database to help you answer questions about setting up and using the product. Additionally, the QAD Web site has information about training classes and other services that can help you learn about QXO and MFG/PRO.

Conventions

Menu References

This guide applies to all releases of QAD Enterprise Applications from eB through to the current release (QAD EE and QAD SE). Several menus have been reorganized between these releases. Differences in menu numbers are noted, when necessary, using the following convention:

User Tool Maintenance (36.20.4; 36.20.2 in eB)

The initial menu number identifies the program in the most recent release. The second menu number applies to the release specified and all earlier releases.

Typographic Conventions

This document uses the conventions listed in the following table.

If you see:	It means:
monospaced text	A command, path, or file name.
<i>italicized monospaced text</i>	A variable name for a value you enter as part of an operating system command; for example, <i>YourCDROMDir</i> .
<i><italicized monospaced text></i>	A variable for a system value such as a drive letter or machine name; for example <i><hostmachine></i> .
indented command line	A long command that you enter as one line, although it appears in the text as two lines.
Note	Alerts the reader to exceptions or special conditions.
Important	Alerts the reader to critical information.
Warning	Used in situations where you can overwrite or corrupt data, unless you follow the instructions.

QAD Support Services

To take full advantage of the flexibility and potential of QAD's interoperability framework in your specific environment, contact your QAD Support representative for information on the installation and customization offerings supplied by QAD Support Services. These offerings include performance enhancements as well as technical and administration training. For details, contact your nearest QAD office, or go to the QAD Web site and click the Global Services link.

QAD QXtend Excelerator Overview

This section provides an overview of QAD QXtend Excelerator.

Introduction **6**

Outlines the processes of Excelerator and discusses the Excelerator menu

Data Integration Process Map **8**

Illustrates the workflow used in the data integration process

Introduction

QAD QXtend Excelerator is an optional data integration tool within the QAD QXtend interoperability framework that provides additional data visibility and manageability to the data integration process between QAD products.

Excelerator helps you perform the following data integration tasks more easily and efficiently:

- Perform initial data load to implement QAD Enterprise Applications.
- Perform data migration to upgrade QAD Enterprise Applications.
- Transfer and convert business transaction data from one QAD application to another.

Excelerator is used in conjunction with QXtend Inbound and Outbound, which must be properly implemented before you can use this tool.

Seamlessly integrated with Microsoft Excel 2007 and 2010 as a document-level customization, Excelerator provides functions for easily designing the worksheet template, populating the worksheet with data, viewing and modifying data in the worksheet, and synchronizing data with the target QAD application, all in a familiar and intuitive user interface.

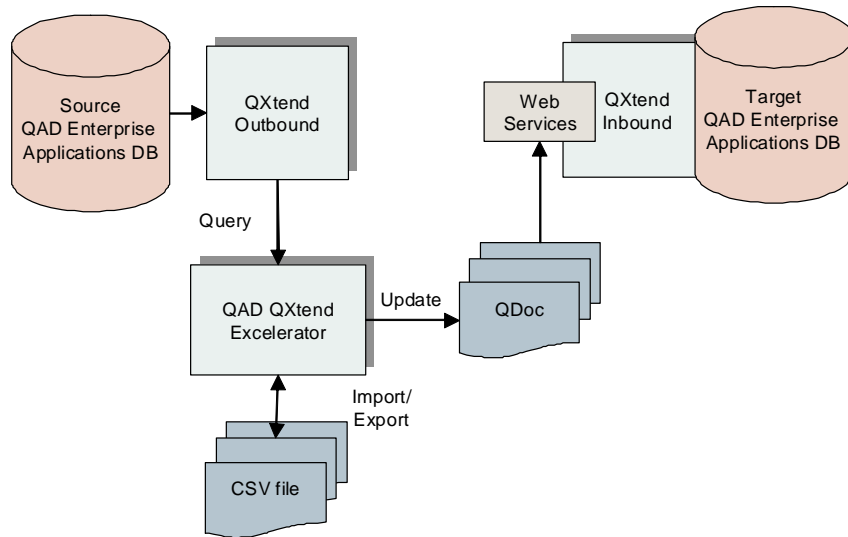
You first create an Excelerator data integration worksheet template by downloading and importing a QDoc (QXtend API schema) from a QXtend Inbound instance. You can further configure the worksheet to hide tables and columns to simplify data entry.

To prepare data to load into the target QAD application, you can import data into the worksheet from three data sources: manual data entry, CSV files, and the Query Service in QXtend Outbound.

You then update data from the Excelerator worksheet to the target QAD application through QXtend Inbound. You select a QXtend instance and receiver to receive the data and can specify whether to use SSL to call the QXtend Web service. The system performs authentication when processing records using the QXtend Web service, and you can stop and restart the load process. If errors occur during the load process, the tool will display errors and highlight records with errors, and you can reprocess failed records. This data integration tool supports all QAD products that expose APIs through QXtend Inbound.

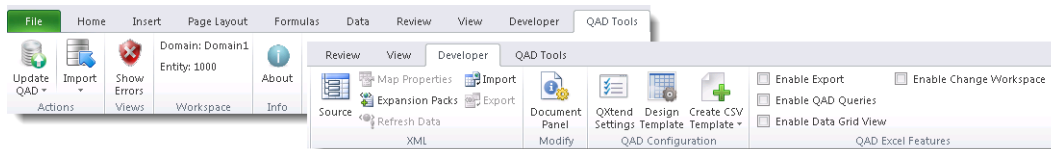
When you save a worksheet, its state is also saved; for example, if you have 10 errors when you save and close the file, you will still have these 10 errors when you reopen the file.

Fig. 1.1
Excelerator Framework



Excelerator dovetails into the Microsoft Excel user interface, and you access all of its functions the same way you use Excel features. The tool’s configuration settings are grouped under the Developer tab in the Ribbon, and all of its data integration features are grouped under the QAD Tools tab. You can enable/disable some of its data integration features as needed.

Fig. 1.2
Excelerator Menus

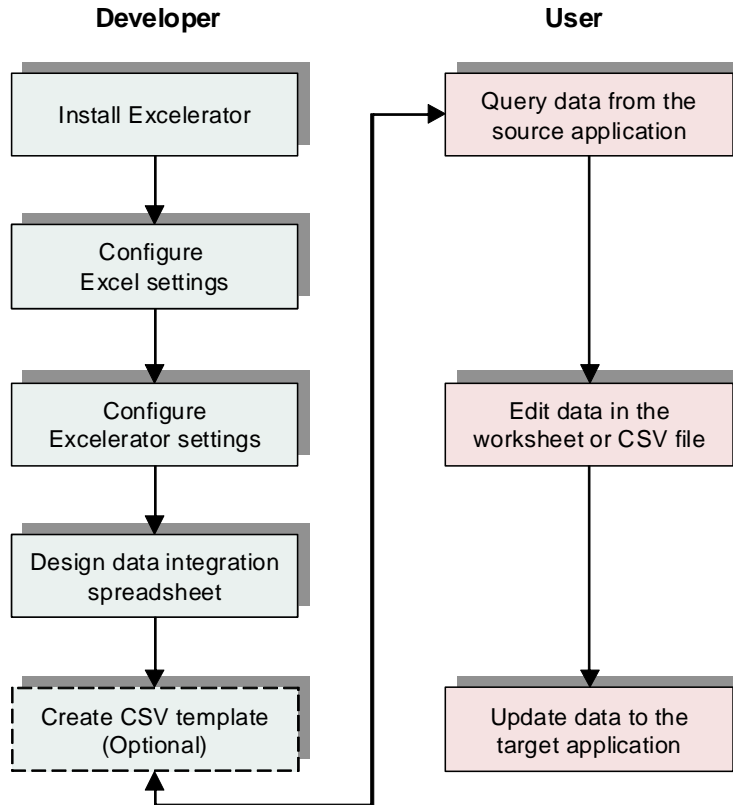


A free trial version of Excelerator ships with QAD QXtend for initial data load and go-live implementation. However, further use of this tool for data integration and communication on a day-to-day basis requires a separate license. Contact QAD Services for information on purchasing licenses for Excelerator.

Data Integration Process Map

The following diagram illustrates the basic data integration workflow using Excelerator.

Fig. 1.3
Data Integration Process Map



Implementing QAD QXtend Excelerator

This section provides detailed instructions for developers on setting up Excelerator for data integration.

Overview 10

Discusses required setup steps

Installing Excelerator and Prerequisite Components 10

Lists prerequisite steps and describes the Excelerator setup and installation procedure

Configuring Excel Settings 12

Lists the steps required to configure Excel settings

Configuring Excelerator Settings 15

Lists the steps required to configure Excelerator settings and discusses the different settings available

Designing Data Integration Worksheet Templates 18

Lists the steps required to design data integration worksheet templates

Working with CSV Files 21

Explains how to import data from and export data into CVS files

Performing Initial Data Load 22

Explains how to perform the initial data load and discusses how to address potential errors

Performing Data Conversion 22

Lists the steps required to perform data conversion

Overview

This section assumes that you are a developer familiar with QXtend configurations. Before Excelerator can be used to perform data integration and communication between QAD products, you must use the following general tasks to set up the tool:

- Installing Excelerator and Prerequisite Components
- Configuring Excel Settings
- Configuring Excelerator Settings
- Designing Data Integration Worksheet Templates

Installing Excelerator and Prerequisite Components

1 Make sure that the following prerequisites are met before you implement Excelerator on your client machine:

- The operating system on the client machine must be Windows XP, Windows 7, or Windows Server 2003.
- Microsoft Excel 2007 or 2010 is installed on the client.
- QXtend Inbound and Outbound (version 1.6.3 and later) have already been properly installed and configured to successfully communicate data between source and target QAD applications.

2 From the installation media, execute `setup.exe` to install the Excelerator workbook and all prerequisite components—Microsoft .NET Framework 3.5 SP1, Microsoft Office Primary Interop Assemblies, and Visual Studio Tools for the Office System 3.0 Runtime Service Pack 1. Follow the on-screen instructions to complete the setup.

3 When setup is complete, you can find a `QXtendWorkbook.xlsx` file in your `My Documents\QAD\QAD QXtend Workbook` folder. This file serves as a template, so do not directly use this file but rather create new working Excel files from the file for data integration.

A default `QXtendWorkbookConfig.xml` file is also deployed to the `My Documents\QAD\QAD QXtend Workbook` folder. The file contains the global Excelerator settings that are set in the QAD QXtend Excelerator Settings window. When the Excelerator is launched, the configuration details (QXtend Location, Update Receiver, QAD Application Settings) are loaded from the `QXtendWorkbookConfig.xml` file. The tool searches the following locations in order of preference when loading the configuration:

- Local directory where the worksheet was opened from.
- The `config` subdirectory under the local directory where the worksheet was opened from.
- Your ClickOnce cache directory. The ClickOnce cache is where required DLLs are installed to. The location is a subdirectory under your `Local Settings\Apps\2.0` directory.
- The `config` directory under the ClickOnce cache.

This mechanism allows you to decide whether or not to bundle the configuration file with the set of worksheets you will be working with. If the configuration file is not exposed in the local directory where the worksheets are stored, the settings will be stored in your ClickOnce cache. The administrator can then restrict access to the cache directory, which would restrict a user from tampering with the QXtend settings.

Setting Up QXtend Outbound Query Service

In order to use the data query function in Excelerator to retrieve data from the source QAD Enterprise Applications, you must first set up the query service as well as configure QDoc APIs in QXtend Outbound. Perform the following steps in QXtend Outbound.

Note For details on setting up the query service, modifying query profiles, and configuring and deploying QDoc APIs, see *User Guide: QAD QXtend*.

- 1 Set up the query service.
- 2 Configure corresponding QDoc APIs.
 - If you are working with standard QDoc APIs, some already have corresponding synchronization profiles defined in QXtend Outbound, so you do not need to do anything for them; for example, for the maintainCustomer QDoc API, there is a corresponding profile called MaintainCustomerData already defined in QXtend Outbound.
 - If the QDoc API being used does not contain the default profile, define the required profile in QXtend Outbound.
 - Customize profiles as needed. Depending on the QDoc API you are using, in some cases, further configuring of the profiles for use with Excelerator may be required. For example, when querying data containing transaction comments, the foreign key fields need to be defined in the transaction comments object. In the case of Analysis Code Master, the following fields have been added to the Transaction Comments object:
 - anCode contains a fixed value of “=\$analysisCode.anCode\$”
 - antype contains a fixed value of “=\$analysisCode.antype\$”

Fig. 2.1
Customizing Profiles

XML Name	Name	Fixed Value	Calc Prog.	Always Pub.	Incl.	Add Only
anCode		=\$analysisCc		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
antype		=\$analysisCc		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cdLang	cmt_lang			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cdRef	cmt_ref			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

This type of assignment uses the fixed value expressions functionality available in QXtend. In this example, the value is replaced by the value in the parent record. This is required because the foreign key definition in the maintainAnalysisCode schema references these values when linking the child transaction comments data to the parent analysis code record.

Fig. 2.2
Reference Example

H	I	J	K	L
analysisCodeTransComment				
operation	antype *	anCode *	cmtSeq *	cdRef
	9	B-ACCUST001		1 B-ACCUST001

The standard replication profiles supplied with QXtend Outbound already include this field assignment when dealing with Transaction Comments data. When designing profiles for use with the Excelerator worksheets, it is important to ensure that this type of field assignment is included in the definition.

- 3 To expose the query profiles to the worksheet, use the Deploy Query button to publish any newly created or modified query profile definitions to QXtend Inbound. This allows QXtend Outbound to accept query requests for the published profiles.

Setting Up Authentication for Query QAD Requests

Query QAD works for all QAD Enterprise Applications versions QAD supports, but Query QAD requests are only authenticated for 2009 and later releases of QAD EE and QAD SE.

To enable this functionality, you must configure a connection pool of type SI-API in QXtend Inbound, and the name of the connection pool must be same as the Update Receiver specified in the Excelerator Settings panel.

For details on configuring connection pools, see *User Guide: QAD QXtend*.

Configuring Excel Settings

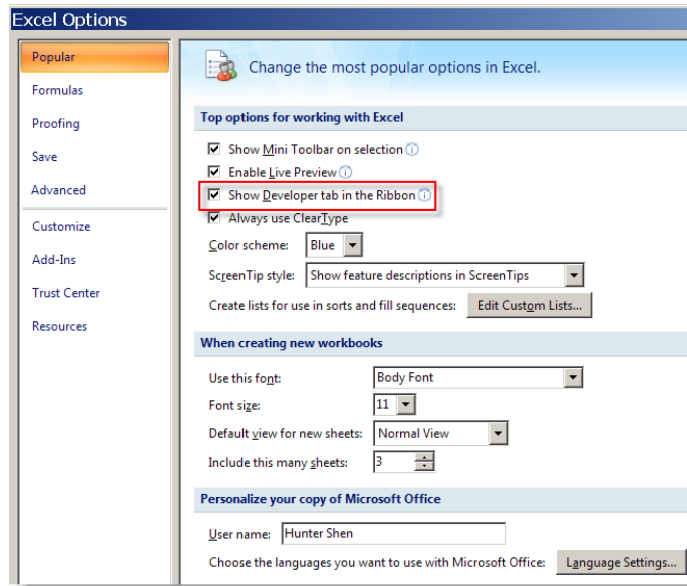
Use the following to configure Excel settings. You only need to perform this once on your machine.

- 1 Create a working Excel file by making a copy of the `QXtendWorkbook.xlsx` template file located in the `My Documents\QAD\QXtendWorkbook` folder. You can rename the new file as you want.
- 2 Open the working Excel file.
- 3 Show the Developer tab in Excel.

For Excel 2007:

- a Click the Microsoft Office Button and then click Excel Options.
- b Click Popular, and then select **Show Developer tab in the Ribbon** under **Top options for working with Excel**. The Developer tab contains configuration options for Excelerator, so you must display it to access these settings.

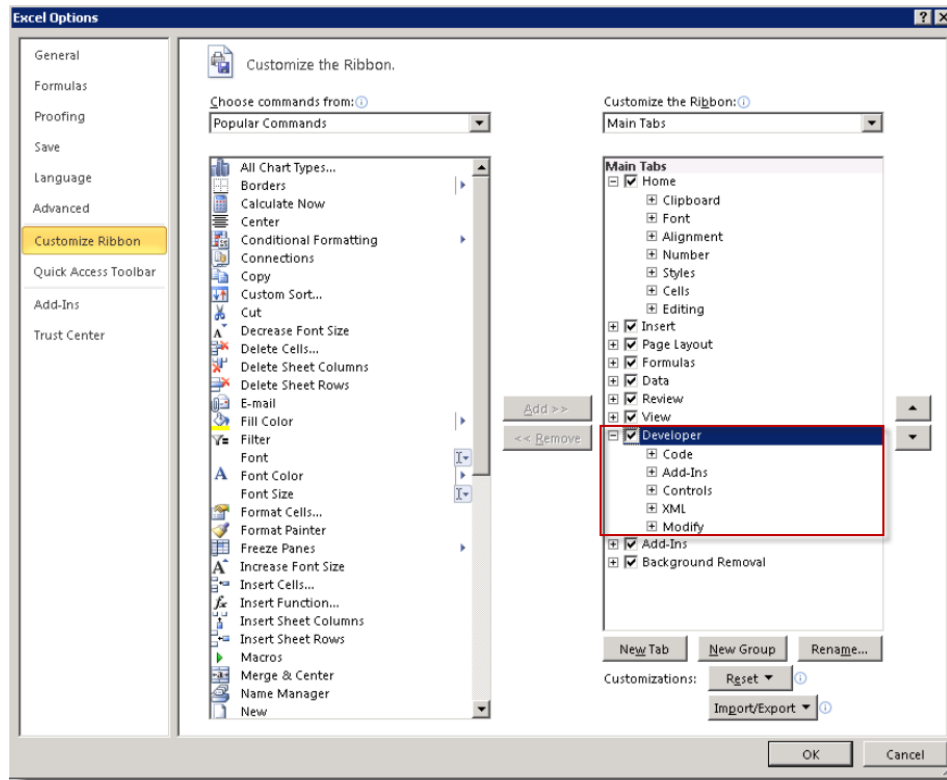
Fig. 2.3
Excel Options—Popular (Excel 2007)



For Excel 2010:

- a** Click the File tab.
- b** Under Help, click Options.
- c** Click Customize Ribbon.
- d** Under Customize the Ribbon, select the Developer check box.

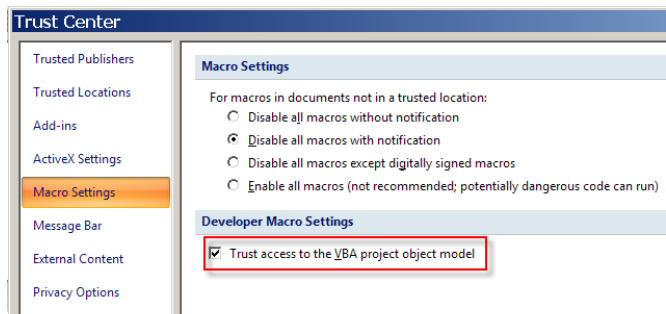
Fig. 2.4
Excel Options—Customize Ribbon (Excel 2010)



Note The Developer tab should only be displayed if the user will be required to access the developer’s functions.

- 4 Click Trust Center, and then click Trust Center Settings.
- 5 Click Macro Settings, and then select **Trust access to the VBA project object model** under **Developer Macro Settings**.

Fig. 2.5
Excel Options—Trust Center



Configuring Excelerator Settings

By default, the global Excelerator settings are stored in the `QXtendWorkbookConfig.xml` file in the `My Documents\QAD\QAD QXtend Workbook` folder.

- 1 In the Ribbon, click the Developer tab.
- 2 In the QAD Excel Features group, select the features of Excelerator that you want to turn on:
 - Enable Export: Lets you export data from the Excelerator worksheet into CSV files.
 - Enable QAD Queries: Lets you query data in QAD applications through the QXtend Outbound Query Service.
 - Enable Data Grid View: Lets you view data in a data grid view.
 - Enable Change Workspace: Lets you change the current domain and entity.
- 3 In the QAD Configuration group, click QXtend Settings.
- 4 In the QXtend Configuration screen, set up proper settings for Excelerator to correctly work with QXtend and QAD applications.

Fig. 2.6
QAD QXtend Excelerator Settings

QAD Application Settings

Domain and Entity. Specify the domain and entity of the QAD Enterprise Application you want to query data from and load data into.

Mnemonics Raw. specify whether QXtend should process the raw value of a mnemonic field or the mnemonic itself. For example, the EMT Type field in Customer Maintenance is stored as a mnemonic. The raw value is 01; however, the mnemonic displayed on the screen is NON-EMT.

Encode Password. Specify whether you want to encode your password for accessing the target QAD Enterprise Applications. Encode Passwords is used if the QXtend application is running with `<encodedPasswords>` set to true. (This value is set in the `qxtendconfig.xml` configuration file in QXtend Inbound.)

QXtend Location

Tomcat Host and Tomcat Port. Provide the host and port information for the QXI server.

QXI Instance Name. Specify the Webapp name, which is the name of the Web service used to connect to QXtend Inbound.

Use SSL. Select this option if using Secure Socket Protocol https encryption. Enabling this is only applicable if QXtend has been configured to run under SSL.

Request Timeout. Enter, in seconds, the maximum wait time for responses from QXtend Inbound.

Update Settings

Update Receiver. Specify the QXtend Inbound receiver to use to load data into the target QAD application.

QDoc Name. Displays the name of the QDoc API the worksheet relates to. The field defaults from the schema imported into the worksheet.

QDoc Version. Displays the version of the QDoc API the worksheet relates to. The field defaults from the schema imported into the worksheet.

Menu Program. The menu program the QDoc API is mapped to. (Optional: this information is used when retrieving menu-specific column labels and authenticating Query requests.)

QAD Release. Specify the QAD release the worksheet is interacting with.

Template Type. Select the default Excelerator worksheet template type to use when creating worksheet templates.

There are two Excelerator worksheet template types:

- Data load template

This is the system default worksheet template type and should be used for loading or updating master data into QAD Enterprise Applications.

When you use this template type, the Process Response QDoc option is cleared by default since no message processing is required for QXtend responses.

- Business transaction template

Use this template type when you want to create from Excelerator new business transaction records (for example, sales orders) in QAD Enterprise Applications that require key field values or business document numbers (for example, sales order numbers) to be generated by the target system.

This feature is realized through the use of the key prefix. When processing the updates to QAD, the temporary key prefix is removed and replaced with the system-generated key (e.g. the sales order number).

Key Prefix. Specify a key prefix (for example, tmp#) to identify key fields or business document numbers of business transaction records (for example, sales order numbers of sales orders) in the worksheet that you want the target QAD application to supply values for. You can then use the key prefix to create temporary key field values (for example, tmp#1, tmp#2, tmp#3 ...) for new records in the worksheet. When the data is sent for processing through the Update QAD function, QXtend replaces the temporary key field values from the requests with

values generated by the target QAD application. When Excelerator receives responses containing the system-generated key field values, it updates the worksheet with the new key field values.

The prefix must and can only be defined when the worksheet template type is Business Transaction Template.

The value you enter here defaults to the Key Prefix field in the Design Template dialog box when you design worksheet templates.

Process Response QDoc. Specify whether you want Excelerator to process response QDocs returned from QXtend and update the worksheet with any new data.

Yes (Selected): Excelerator will process response QDocs containing detailed business data (the `suppressResponseDetails` attribute is set to false) returned from QXtend. Any new data in the response messages will be updated into the worksheet.

No (Cleared): The Excelerator will ignore any response QDocs returned from QXtend. The `suppressResponseDetails` attribute is set to true in the request sent to QXtend Inbound, which limits the data contained in the response to processing status and exception messages, excluding any business data to enhance the message processing performance of QXtend Inbound. You should set the option to No for data load templates.

You can only specify this option for data load templates. For business transaction templates, the option is always set to Yes and cannot be changed.

Query Settings

Query Receiver. Specify the receiver to use to query data from the source QAD application. The receiver must be a receiver listed under the Outbound module receivers defined in QXtend Inbound.

Source Application. Specify the source application instance name.

Profile. Specify the name of the profile in QXtend Outbound that defines the format of the queried data.

Note The format must be compatible with the request XSD that the worksheet was built from. This is important because the format of the query response data must map to the table and field names defined in the worksheet. The table name and field names are case sensitive.

QDoc Version. Specify the QDoc version.

Max Rows. Enter the default maximum number of records to retrieve into a worksheet during data query. Entering zero (0) indicates all matching rows will be returned.

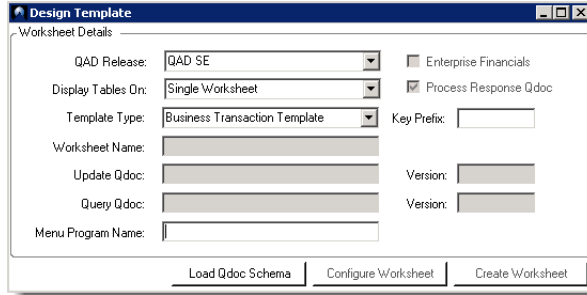
Filter. Enter a valid Progress query to use as filter criteria to filter records. The filter must be a valid Progress ABL WHERE clause and can only be defined against the top-level table of the QXO business object; for example:

```
cm_site = 'SITE1' and (cm_addr >= '0100' and cm_addr <= '100')
```

Designing Data Integration Worksheet Templates

- 1 Under the Developer tab in the Ribbon, click Design Template. The pop-up window displays worksheet details settings.

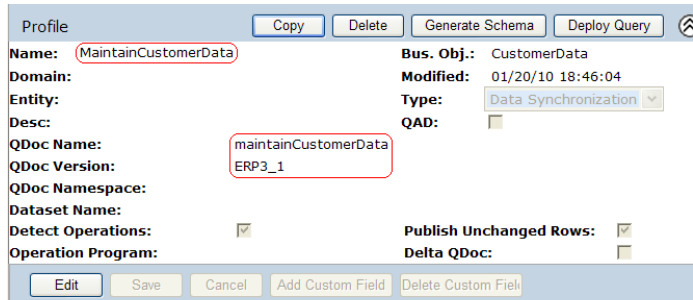
Fig. 2.7
Design Template



- 2 In the QAD Release field, select the version of the source and target QAD Enterprise Applications.

The Query QDoc name and version in the screen map to a corresponding profile defined in QXtend Outbound. For example, a worksheet has been designed to work with the Customer Data QDoc. This means the maintainCustomerData QDoc API has been imported to the worksheet. In QXtend Outbound there is a corresponding profile defined called MaintainCustomerData.

Fig. 2.8
Profile in QXO



After the MaintainCustomerData profile has been deployed as a query to QXtend Inbound, it is available for use with the worksheet.

The version of the query QDoc depends on what QAD release has been selected, as set below:

QAD Enterprise Applications Release	Default Version
QAD EE	ERP2_1
QAD SE	eB21_1
eB2.1	eB21_1
eB2	eB2_1
eB	eB_1

The versions for QAD EE QDocs are displayed in QXtend Inbound as follows:

Fig. 2.9
QAD EE QDoc Versions

<input checked="" type="checkbox"/>	QdocName	XMLSyntax	Version	Route	Procedure
<input type="checkbox"/>	queryMaintainAnalysisCode	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/C
<input type="checkbox"/>	queryMaintainCustomer	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/C
<input type="checkbox"/>	queryMaintainCustomerData	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/C
<input type="checkbox"/>	queryMaintainRoutingAPI	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/C

- 3 In the Display Tables On field, specify whether you want to display all data in a single worksheet or create a separate worksheet for each table.

Note When custom add-ins are deployed in your Excel, the application may become unstable and unresponsive when the number of worksheets in the file exceeds three. When this occurs, use the Single Worksheet option or uninstall the custom add-ins.

- 4 From the Template Type list, select a worksheet template type. This defaults from the QAD QXtend Excelerator settings. See "Template Type" on page 16.

If you select the business transaction template type, you should also specify the Key Prefix value and the Process Response QDoc option. These also default from the QAD QXtend Excelerator settings. See "Key Prefix" on page 16 and "Process Response QDoc" on page 17.

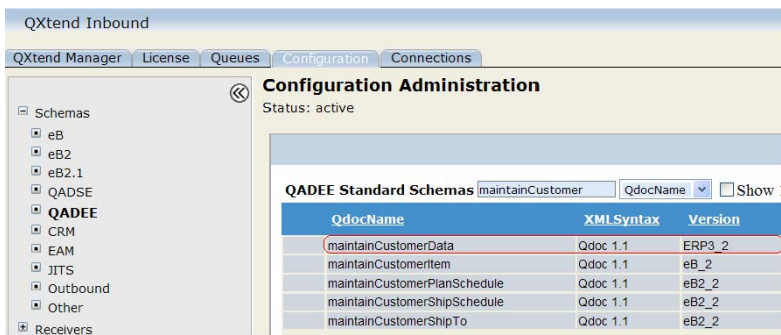
- 5 Click Load QDoc Schema, and then locate a QDoc file to load into the worksheet.

Only QDoc 1.1 syntax schemas are compatible with Excelerator.

The schemas are located in the WEB-INF/schemas directory of the QXtend Inbound application. To identify the correct version of the schema to import, view the schema information displayed in the Schema View in Configuration Manager in QXtend Inbound.

For example, for the QAD EE maintainCustomerData QDoc, the Schema View shows its version number is ERP3_2, so you need to retrieve maintainCustomerData-ERP3_2.xsd from the TOMCAT_HOME/webapps/<QXI webapp>/WEB-INF/schemas/QADEE directory.

Fig. 2.10
Schema View



The tool processes the QDoc schema, imports all the tables and fields into the worksheet, and automatically fills the appropriate values into the Worksheet Name, Update QDoc, and Query QDoc fields based on the QDoc data.

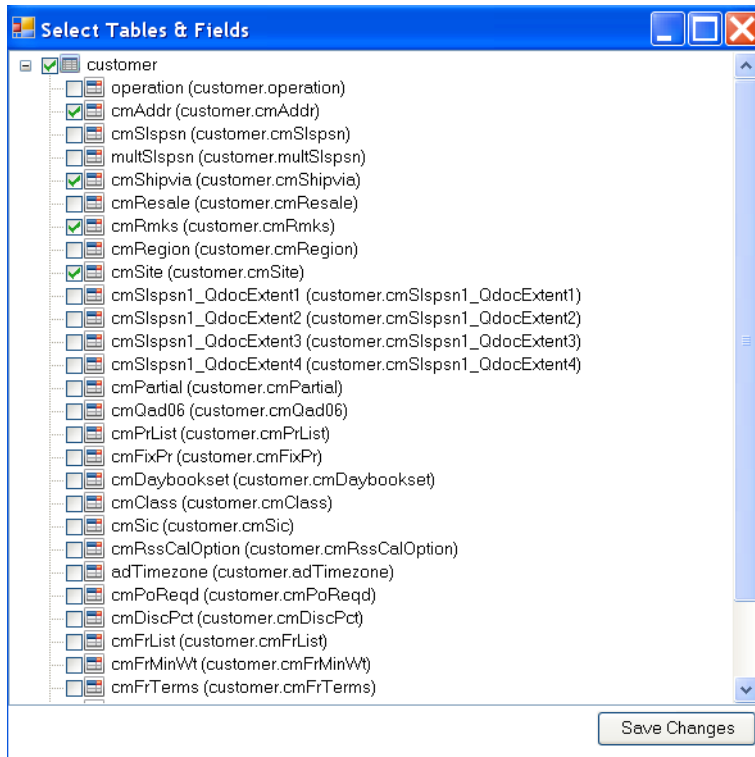
The Menu Program Name field displays the menu program the QDoc API relates to. This information is also used when retrieving menu specific column labels and authenticating Query QAD requests (QAD SE and QAD EE only).

- 6 Optionally, click Configure Worksheet to customize the schema in the worksheet. In the Select Tables and Fields window, select the tables and fields you want in the worksheet. Table primary keys are automatically selected by the tool.

Note Column labels are only retrieved for 2009 and later releases of QAD EE and QAD SE. To enable this functionality, you must configure a connection pool of type SIAPI in QXtend Outbound, and the name of the connection pool must be same as the Update Receiver specified in the Excelerator Settings panel.

For details on configuring connection pools, see *User Guide: QAD QXtend*.

Fig. 2.11
Select Tables and Fields



- 7 Click Create Worksheet. If you chose to create a single worksheet, the tool creates a worksheet with tables and fields from the schema translated into rows and columns; if you chose to create multiple worksheets, the tool creates a separate worksheet for each table from the schema. Primary keys in tables are marked with an asterisk (*). In the worksheet, operation and index fields are displayed in the same sequence as that in the QDoc; all other fields are sorted in alphabetical order.

Fig. 2.12
Worksheet with Schema

	A	B	C	D	E	F	G	H
1	customer							
2	Customer *	Salesperson	Multiple	Ship Via	Resale	Remarks	Region	Site
3							+	
4								
5								
6								
7								
8								

- 8 Save the Excel file. You can now use the blank worksheets for entering data, querying data from the QAD application, and updating data to the QAD application.

Modifying Worksheet Templates

You can use Configure Worksheet to modify the schema structure in the worksheet template you have already created. So, if you forgot to include a field, you can simply go back to Configure Worksheet to select the field you missed.

However, if you have already entered data in the worksheet, when you modify the template design and update the worksheet, the system prompts you to retain existing data in the worksheet. If you choose to retain the existing data, all the rows in the worksheet are marked as Processed and you must use Update QAD|Process All later to update data to the target QAD application.

Working with CSV Files

Creating CSV Templates

After you design the data integration worksheet, you can create a CSV template from the worksheet. You can then populate the CSV file with data from QAD or third-party applications either manually or programmatically, and import it into the Excel worksheet for data synchronization with the target QAD application.

To create a CSV template, under the Developer tab in the Ribbon, click Create CSV Template, and select Single CSV Template to create one CSV file for all the tables in the worksheet, or select Template Per Table to create multiple CSV files, one for each table in the worksheet; then specify a location to save the file.

Importing and Exporting CSV Files

Under QAD Tools, use the Import and Export functions to use CSV files for data entry.

You can import the CSV file—either a single CSV file containing all the tables or separate CSV files each containing one table—into the Excelerator worksheet.

Using the CSV export tool, you can export data from the Excelerator worksheet into a single CSV file or multiple CSV files, one for each table. The structure of the exported CSV file strictly follows that defined in the imported CSV file.

Performing Initial Data Load

For go-live implementation of QAD Enterprise Applications, you need to load some initial data into the system to quickly get it up and running. Use the following general steps to perform the initial data load using Excelerator:

- 1 Design a set of data integration worksheets for the initial data load.
- 2 Use the CSV template creation function to generate CSV templates that outline the expected formats of CSV files for use with the worksheets.
- 3 Use the CSV templates to convert data from a legacy system into the structure defined in the templates and load it into the worksheets. You populate data in the CSV templates either programmatically or through data output from the legacy system. The format of the CSV data must match the CSV template.
- 4 Use the Import From CSV function to load the data from the CSV files into the worksheets. Any errors that occur during CSV file import are captured and displayed in the grid view. You must correct these errors before you can proceed with data import.

Fig. 2.13
Data Import Errors

operation	anType	anCode	anDesc	anActive
	9	B-ACCUST001	American Custo...	<input checked="" type="checkbox"/>
R	9	B-ACCUST002	Distribution Cust..	<input checked="" type="checkbox"/>
	9	B-ACCUST003	Customer Link	<input checked="" type="checkbox"/>
	6	B-ACITEM001	Lomg String testi..	<input checked="" type="checkbox"/>
R	9	Q001	xxxx	<input checked="" type="checkbox"/>
	6	Q002	test	<input checked="" type="checkbox"/>
	9	qxtend01	tttt	<input checked="" type="checkbox"/>
	9	qxtend02	eac test	<input checked="" type="checkbox"/>

Table Name	Field Name	Key Data	Data Import Error Description
analysisCodeTra...		9,Q009,1	ForeignKeyConstraint analysisCode_an
analysisCodeTra...	cdSeq	,,	Input string was not in a correct format.C

- 5 If required, modify the imported data in the worksheets.
- 6 Use the Update QAD function to load the data from the worksheets into target QAD Enterprise Applications.

Performing Data Conversion

Use the following general steps to perform data conversion when you upgrade your QAD Enterprise Applications:

- 1 Design a set of data integration worksheets for the data conversion.
- 2 Use the Query QAD function to query data from the source QAD Enterprise Applications into the worksheets.

- 3 Modify the data in the worksheets as required.
- 4 Use the Change Workspace function to switch to the domain and entity of the target QAD Enterprise Applications.
- 5 Use the Update QAD function to load the data from the worksheets into target QAD Enterprise Applications.

Upgrading Excelerator

If you already have an earlier version of Excelerator installed, perform these simple steps to upgrade it to the latest version:

- 1 Uninstall the previous version of Excelerator using either one of the following methods:
 - Run `setup.exe` from the previous version of Excelerator and then choose the Uninstall option.
 - Uninstall programs QAD QXtend Workbook and QXtendWorkbook using the Windows control panel.
- 2 Run `setup.exe` from the latest version of Excelerator to install the new program. Worksheets created using earlier versions of Excelerator are compatible with the new version.

Using QAD QXtend Excelsior

Overview 26

Discusses how QXtend Excelsior can be integrated with other programs

Querying Data from the Source Application 26

Lists and describes the steps required to use the Query function accurately

Viewing and Modifying Data 28

Explains how to use different view functions and delete data

Changing Workspace 30

Explains how to use the Change Workspace feature

Updating Data to the Target QAD Application 30

Lists the steps required to update data to other applications using QXtend Inbound

Overview

This section assumes that you are a user responsible for performing data integration between QAD products using Excelerator. By importing QDoc schemas, you use Excelerator to translate database tables and fields of QAD Enterprise Applications into rows and columns in the Excel worksheets. You can then use the worksheets to hold records queried from the source QAD application, view and modify data, and finally update them to the target QAD application.

Querying Data from the Source Application

Note The Query QAD function is disabled when the worksheet is designed to work with an EE Financials component schema. This is because the query service is not currently compatible with the EE Financials components.

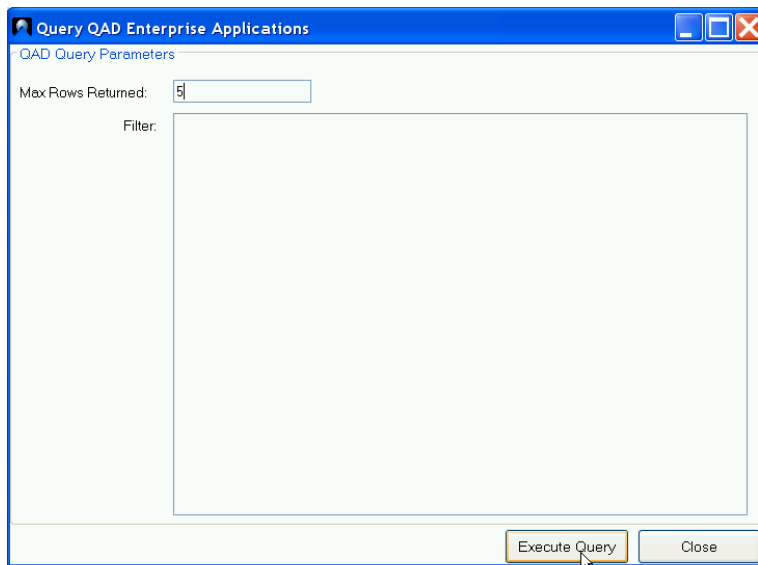
- 1 Under the QAD Tools tab in the Ribbon, click Query QAD.
- 2 In the Query QAD Enterprise Applications window, define the query parameters.

Max Rows Returned. Enter the maximum number of records to retrieve into a worksheet. Entering zero (0) indicates all matching rows will be returned.

Filter. Enter a valid Progress query to use as filter criteria to filter records. The filter must be a valid Progress ABL WHERE clause and can only be defined against the top-level table of the QXO business object; for example:

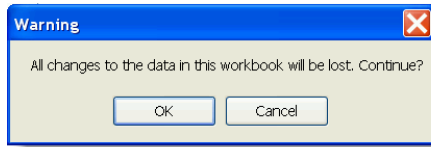
```
cm_site = 'SITE1' and (cm_addr >= '0100' and cm_addr <= '100')
```

Fig. 3.1
QAD Query Parameters



- 3 Click Execute Query. A warning message appears. Click OK to proceed.

Fig. 3.2
QAD Query Warning



- 4 The query pulls records from the source QAD application into the worksheet. Save the file.

Troubleshooting Data Queries

No Data Returned from the Data Query

If no data comes back from the data query, examine the `qdocResponses.log` file for details on what data was returned and whether it matches the format expected in the worksheet and Excelerator settings.

Make sure that worksheet and column header names in the Excelerator worksheet match data object and field XML names in the profile definition in QXtend Outbound and that the profile name specified in the Excelerator settings is identical to that defined in QXtend Outbound. All these names are case-sensitive and they must be exactly mapped so that Excelerator knows where to place queried data into the corresponding cells in the worksheet.

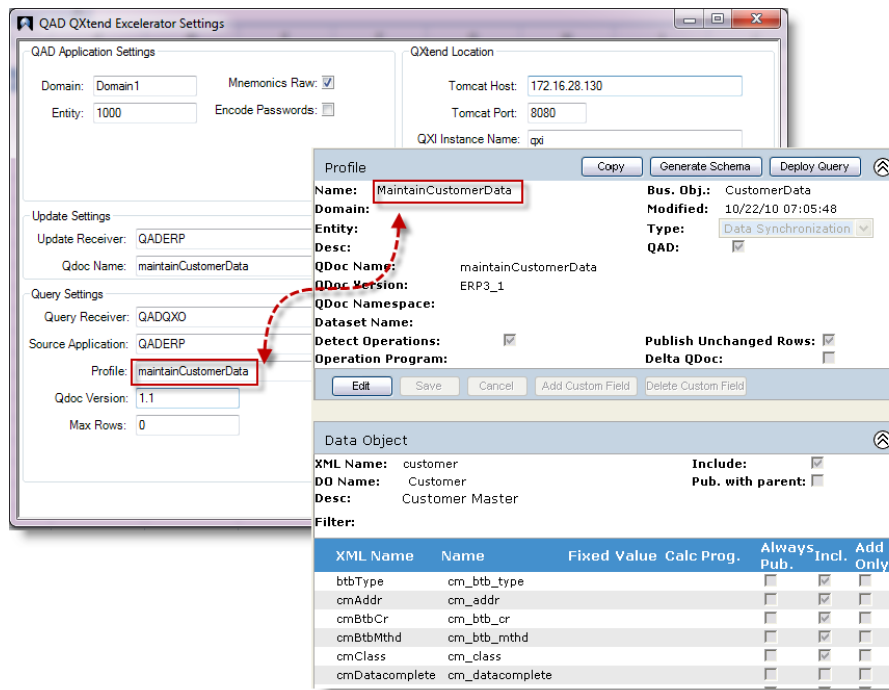
Fig. 3.3
Mapping of Data Object and Field Names

The screenshot shows the QAD QXtend Outbound configuration interface. On the left, the 'Profile' section is expanded to show 'Data Object' settings. The 'XML Name' is 'customer' and the 'Name' is 'Customer Master'. Below this is a table mapping XML names to field names:

XML Name	Name	Fixed Value	Calc Prog.	Always Pub.	Incl.	Add Only
bttType	cm_btt_type			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cmAddr	cm_addr			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cmBttCr	cm_btt_cr			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cmBttMthd	cm_btt_mthd			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cmClass	cm_class			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
cmDatacomplete	cm_datacomplete			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

On the right, an Excel worksheet is shown with columns A through F. The header row contains the following dropdown menus: 'customer', 'bttType', 'cmAddr', 'cmBttCr', 'cmBttMthd', 'cmClass', and 'cmDatacomplete'. Red dashed arrows indicate the mapping from the 'XML Name' field in the configuration to the dropdown menu in the worksheet, and from the 'Name' field to the corresponding column header.

Fig. 3.4
Mapping of Profile Names



Unhandled Exception Error

When you run into the following error message, read the details to determine the cause of the error.

An unhandled exception occurred in your application. Contact your administrator or the author of this document for further assistance

A couple of situations may give rise to this error:

- If the detailed error message reads “Object reference not set to an instance of an object,” this error is caused by a mismatch between XML names of primary key fields and the field names (not field labels in QEA) in the schema. Make sure field names in the schema are identical to corresponding XML names.
- If the detailed error message indicates that the string is not a valid boolean value, check boolean fields in the schema to see whether they have the proper values. This error often occurs to custom fields.

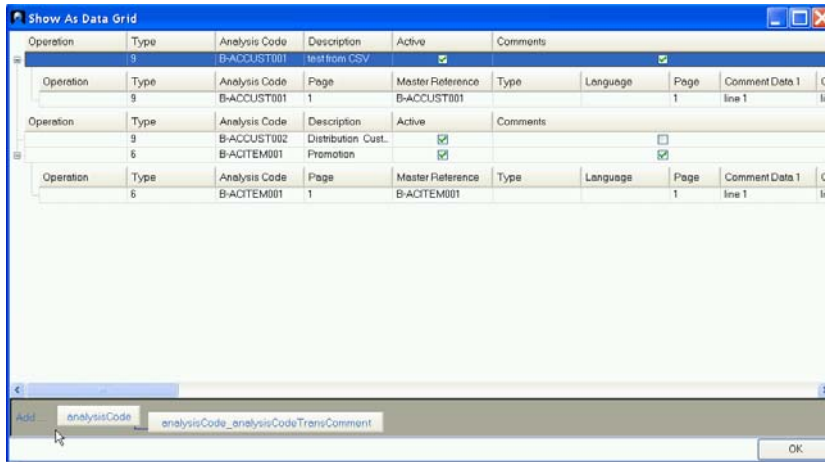
Viewing and Modifying Data

After you execute a query to import data from the source QAD application into your worksheets, you can view and modify data just like you do with any other Excel worksheets.

The tool does not perform real-time validation of the data you enter in the worksheet, but if you enter invalid data, error will occur when you update records to the target QAD application later.

Alternatively, when the Enable Data Grid View option (in the QAD Excel Feature group under the Developer tab) is selected, you can click Grid View under the QAD Tools tab to view and edit the data in the data grid view. The data grid view gives you a better view of the parent-child relationships among data.

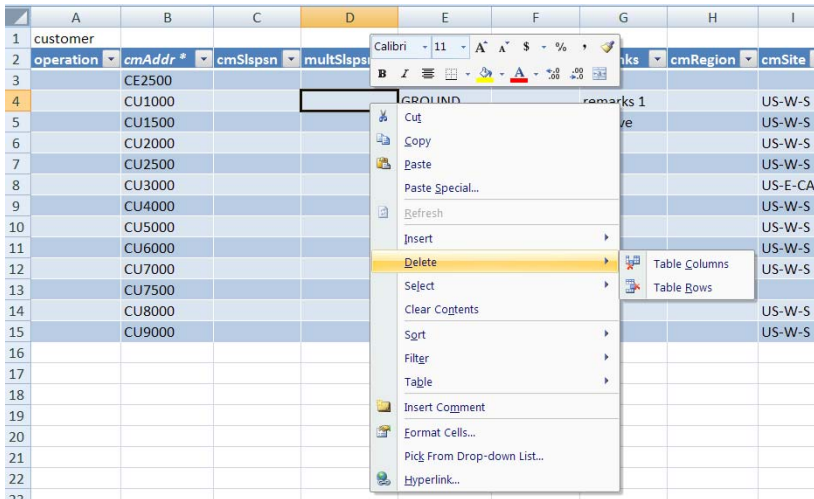
Fig. 3.5
Data Grid View



To delete records from the worksheet, select the rows and right-click; then from the pop-up menu, point to Delete and select Table Rows. This also deletes any corresponding child records in the worksheet. For example, if you delete a record in table routing, the tool automatically deletes its child records that reference the primary key of the parent record in table routingTransComment.

Deleting rows from the worksheet will not delete corresponding records from the target QAD Enterprise Applications when you perform data update. To remove records from the target application, you must populate the “operation” column with R (for Remove) in corresponding rows to indicate to QXtend Inbound to delete the records when processing the QDoc request during data update.

Fig. 3.6
Deleting Records

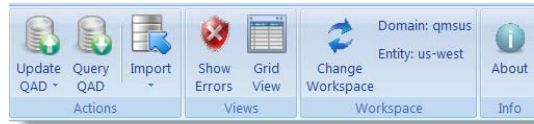


Changing Workspace

If the change workspace feature is enabled, you can switch the current domain and entity your Excelerator worksheet is currently working with.

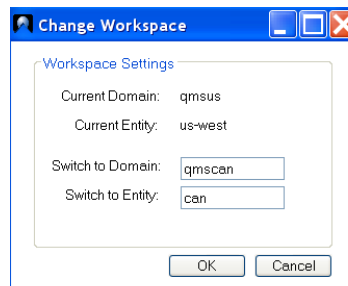
The current workspace for the worksheet is displayed in the Workspace group under the QAD Tools tab.

Fig. 3.7
Current Workspace



Click the Change Workspace button to switch to another domain and entity. Any Query QAD and Update QAD requests will use the new domain and entity combination. The workspace group will be updated to display the current domain and entity of the worksheet.

Fig. 3.8
Change Workspace

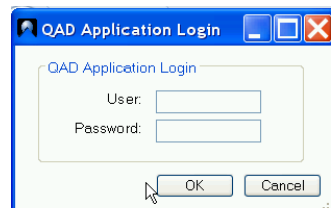


Updating Data to the Target QAD Application

When you have finished viewing and editing data in the worksheet, you can update it to the target QAD application through QXtend Inbound.

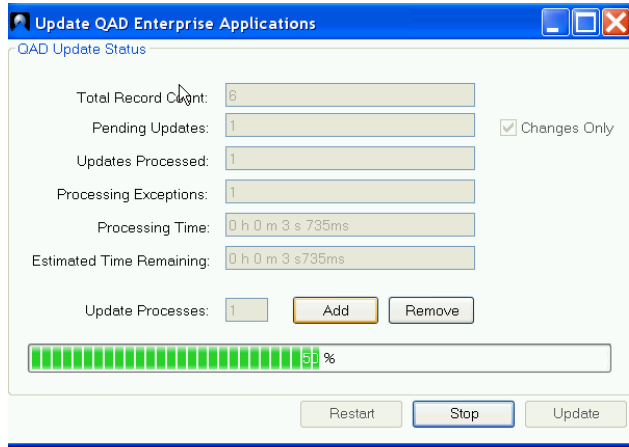
- 1 Under the QAD Tools tab, click Update QAD and select from one of the following options:
 - Process Changes: Only submit records that have been modified.
 - Process All: Submit all records in the worksheets.
- 2 A QAD Application Login window appears. Provide log-in information to access the target QAD Enterprise Applications.

Fig. 3.9
QAD Application Log-In



- 3 A QAD Update Status window appears with detailed update status information and progress bar. You can stop and restart the process as needed.

Fig. 3.10
QAD Update Status



- 4 When the update is complete, any errors that occurred show in the Document Actions pane below the worksheets. You can switch the error information on and off using the Show/Hide Errors button under the QAD Tools tab. Double-clicking on a row in the errors panel sets the focus in the worksheet to the row with the error.

Fig. 3.11
QAD Update Status

Document Actions			
Severity	Error Description	Error Context	Field
error	ERROR: User is not allowed to access this function.		
error	Program did not initialize.		
error	ERROR: User is not allowed to access this function.		
error	Program did not initialize.		
error	ERROR: User is not allowed to access this function.		
error	Program did not initialize.		

Index

D

data

- query 26
- update 30
- view and modify 28

data query

- setup 11
- troubleshooting 27

I

installation 10

K

key prefix 16

P

process map 8

process response QDoc 17

Q

QAD Support Services 4

S

settings

- Excel 12
- Excelerator 15

T

template type 16

