User Guide

QAD QXtend Excelerator

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

70-3191-1.5
Version 1.5
December 2025

Excelerator_UG_v015.pdf/qgl/qgl/f6j

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

Contents

QAD QXtend Excelerator User Guide Change Summary

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

Date/Version	Description	Reference
December 2025/1.5	Added information on LogInAuth feature.	page 17
	Updated QAD QXtend Excelerator Settings screenshot	page 17
	Added information on new Auth0 fields	page 18
July 2016/1.4	Updated the steps for installing Excelerator and prerequisite components	page 12
July 2013/1.3	Updated information on the Excelerator license	page 9
	Updated the steps for installing Excelerator and prerequisite components	page 12
	Added Excel 2013 in Configuring Excel Settings	page 15
	Added new configuration options: Debug Level and Scope Transaction	page 19
	Added the use of Export Errors button in the update steps	page 33
September 2011/1.2	Modified the introductory text to Excelerator	page 8
September 2011/1.2	Updated Excelerator's prerequisite components	page 12
September 2011/1.2	Included steps to show the Developer tab in Excel 2010	page 15
September 2011/1.2	Added instructions on upgrading Excelerator	page 26
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 17
September 2011/1.2	Updated the steps for designing data integration worksheets to include new worksheet template type settings	page 20
September 2011/1.2	Updated the section on working with CSV files	page 24
September 2011/1.2	Updated the section on troubleshooting data queries	page 29

Chapter 1

QAD QXtend Excelerator Overview

This section provides an overview of QAD QXtend Excelerator.

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 with QXtend Inbound and Outbound, which must be properly implemented before you can use this tool.

Seamlessly integrated with Microsoft Excel 2007, 2010, 2013, and 2017 as a document-level customization, Excelerator provides the following functions all in a familiar and intuitive user interface:

- Designing the worksheet template
- Populating the worksheet with data
- Viewing and modifying data in the worksheet
- Synchronizing data with the target QAD application

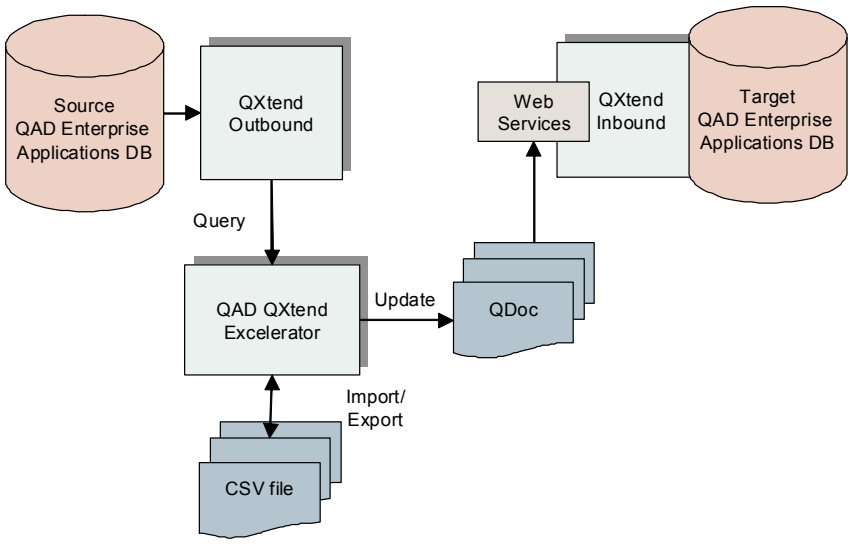
You first create an Excelerator data integration worksheet template by downloading and importing a QDoc (QXtend API schema) from a QXtend Inbound instance. You can further configure the worksheet to hide tables and columns to simplify data entry.

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 displays errors and highlights records with errors. You can reprocess failed records. This data integration tool supports all QAD products that expose APIs through QXtend Inbound.

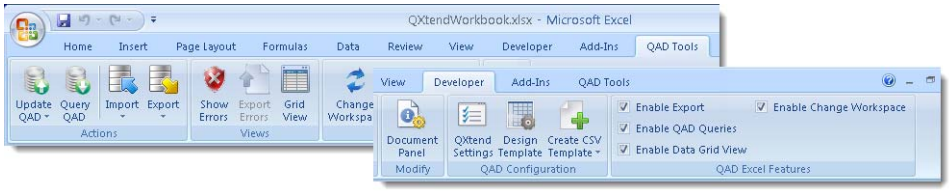
When you save a worksheet, its state is also saved. That is, if you have ten errors when you save and close the file, these ten errors are still there when you reopen the file.

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

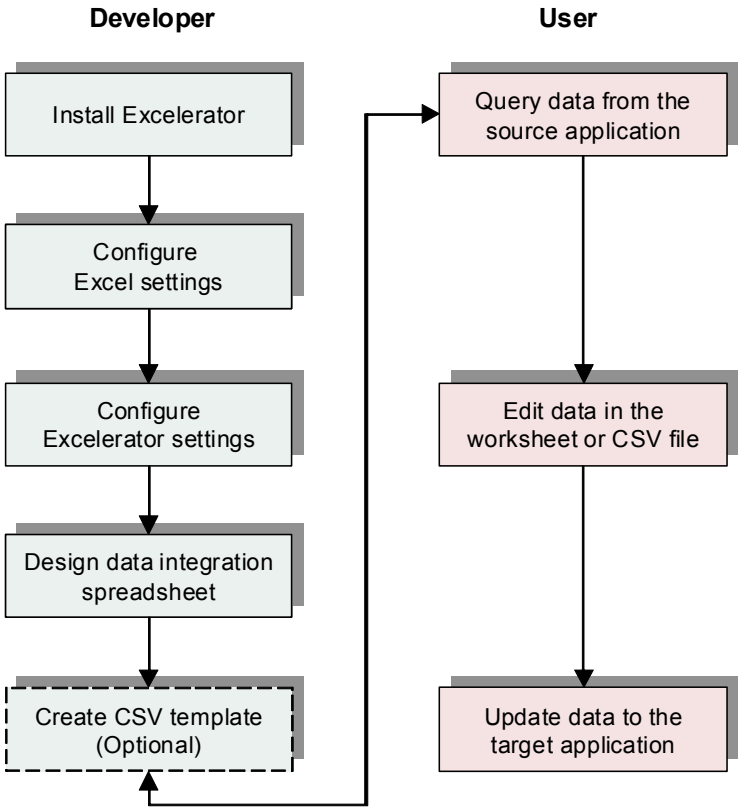


Contact your Sales Representative to get an Excelerator license for initial data load and go-live implementation. If you need more time than the license validity period for the implementation, request a new license from your sales representative. If you need further use of Excelerator for data integration and communication on a day-to-day basis, purchase a separate license.

Data Integration Process Map

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

Fig. 1.3
Data Integration Process Map



Chapter 2

Implementing QAD QXtend Excelerator

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

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, 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

To install Excelerator, make sure that you have administrator privileges. If not, you can ask your system administrator to install Excelerator into your user account. If multiple users on the system want to use Excelerator, the system administrator must install Excelerator for each user account respectively. Installing once on the system cannot enable all users to use Excelerator.

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 Vista, Windows Server 2008, Windows 7, Windows 8, or Windows 10.
   - Microsoft Excel 2007, 2010, 2013, or 2016 is installed on the client.
   - QXtend Inbound and Outbound (version 1.6.3 or later) have already been properly installed and configured to successfully communicate data between source and target QAD applications.
2. Download the Excelerator package from QAD Release Fulfillment.
3. Extract the downloaded zip file. You can see the `data.zip` file in the extracted folder.
4. Extract the `data.zip` file.

   Note There are three folders in the `data.zip` file. Each folder contains the installer for a specific Microsoft Excel version. Ensure that you get the right installer that supports the Microsoft Excel version installed on your system.

Folders	Supported Excel Versions
Excel2007	Excel 2007
Excel2010	Excel 2010
Excel2013	Excel 2013; Excel 2016

5. Open the corresponding folder and execute `setup.exe` to install all prerequisite components and the Excelerator workbook. The components include Microsoft .NET Framework 4.5, Microsoft Office Primary Interop Assemblies, and Microsoft Visual Studio 2010 Tools for Office Runtime. Follow the on-screen instructions to complete the setup.

Important For Excel 2010, if Microsoft Visual Studio Tools for the Microsoft Office System 3.0 is installed on your client machine, uninstall it before you execute `setup.exe` and remove the registry key under `HKEY_CURRENT_USER\Software\Microsoft\VSTO\SolutionMetadata`. Otherwise, you cannot open any Excelerator worksheet.

6 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 to bundle the configuration file with the set of worksheets you are working with. If the configuration file is not exposed in the local directory where the worksheets are stored, the settings are 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

To use the data query function in Excelerator to retrieve data from the source QAD Enterprise Applications, first set up the query service and 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*.

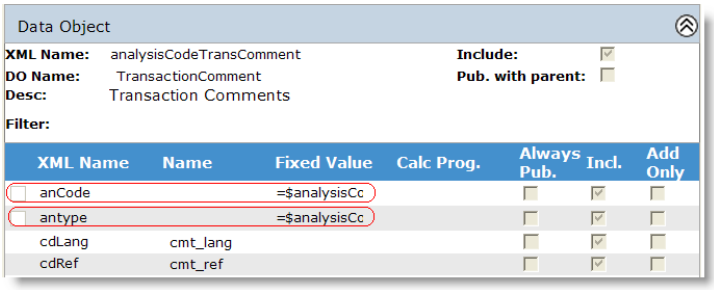
1 Set up the query service.

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.

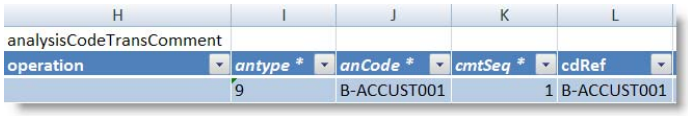
- Customize profiles as needed. Depending on the QDoc API you are using, in some cases, further configuring of the profiles for use with Excelerator may be required. For example, when querying data containing transaction comments, define the foreign key fields in the transaction comments object. For 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



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. It 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



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, configure a connection pool of type SIAPI in QXtend Inbound. Ensure that the name of the connection pool is the 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. Once done, the settings are applied to all the Excel files 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.

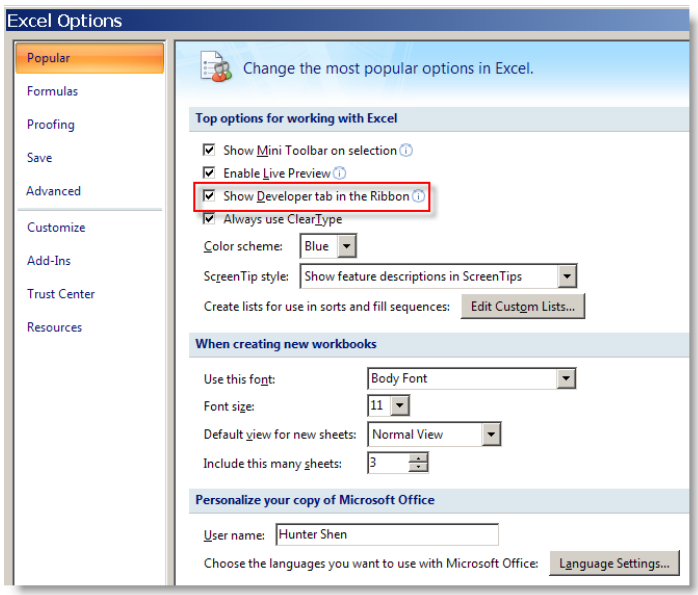
   For Excel 2007:

   a. Click the Microsoft Office Button and then click Excel Options.

   b. On the Popular tab, 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)

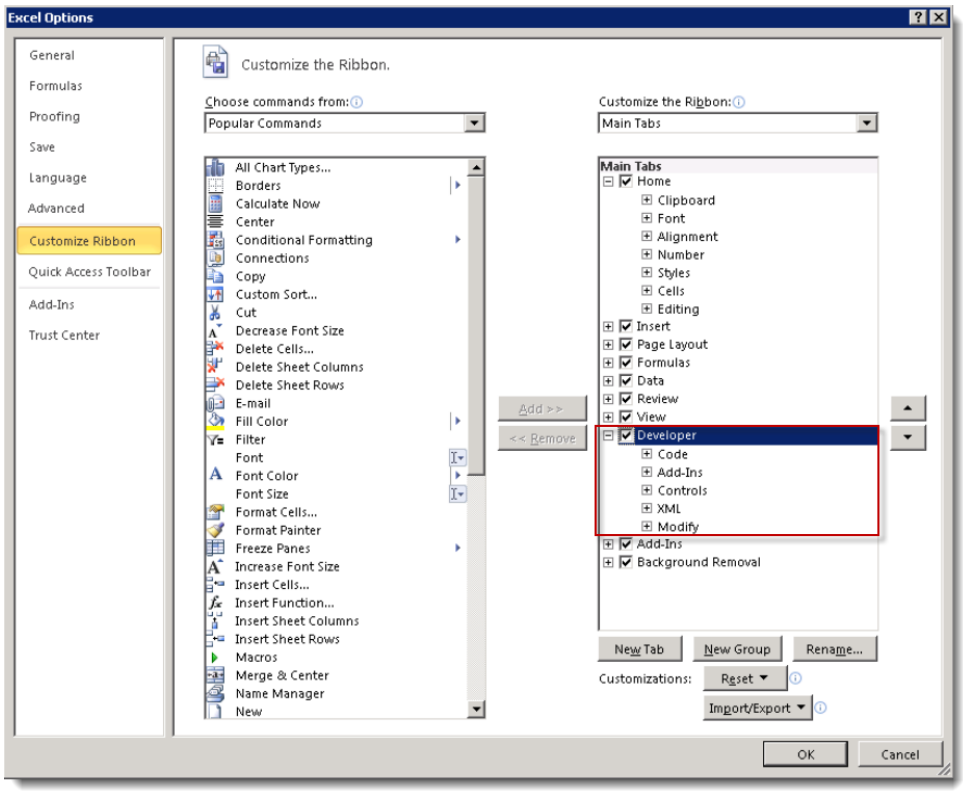


For Excel 2010, Excel 2013, and Excel 2016:

a. Click the File|Options|Customize Ribbon.

b. Under Customize the Ribbon, select the Developer check box.

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

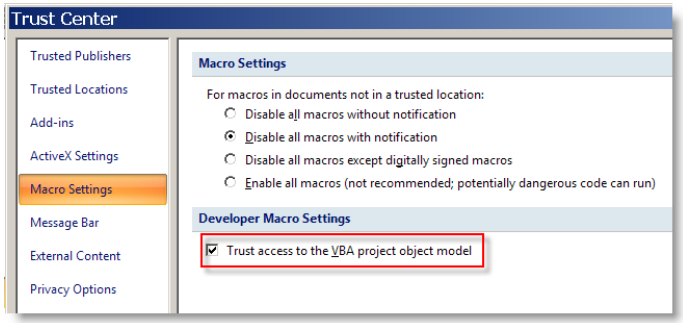


Note Display the Developer tab only for the users who are required to access the developer functions.

4 On the Trust Center tab, 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 Excelerator` folder.

1 In the Ribbon, click the Developer tab.

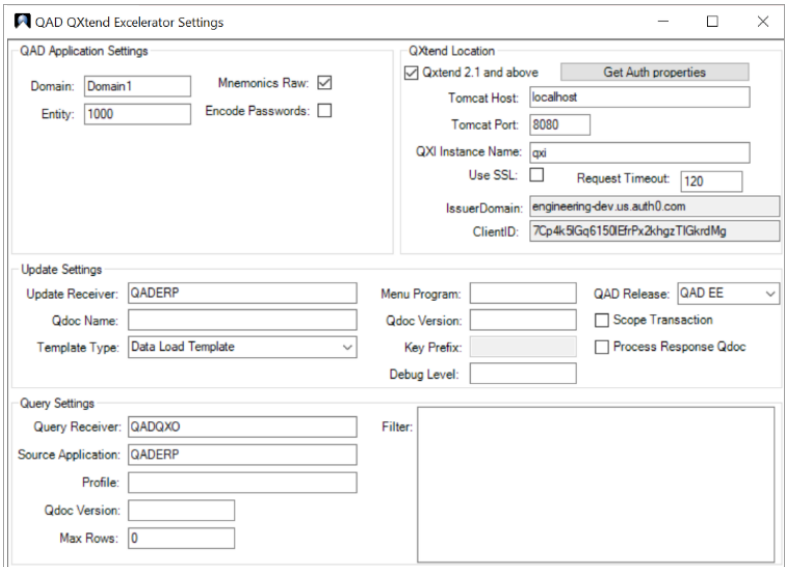
2 In the QAD Excel Features group, select the features of Excelerator that you want to turn on:

- 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.
- LogInAuth: Lets you enable secure login authentication. This feature is compatible with Excel 2013 or newer versions and requires .NET Framework 4.8. This feature is available only when the following options are configured in the QAD QXtend Excelerator Settings screen:
  - QXtend 2.1 and above checkbox is selected.
  - ClientID value is available.
  - IssuerDomain value is available.

3 In the QAD Configuration group, click QXtend Settings.

4 In the QAD QXtend Excelerator Settings screen, set up proper settings for Excelerator to 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 allow QXtend to 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

Qxtend 2.1 and above. Select this checkbox to enable Auth0 authentication. Selecting this checkbox enables the authentication property fields.

Note If this checkbox is not selected, Excelerator operates without Auth0 authentication, providing backward compatibility for older QXtend versions.

Get Auth properties. Click this button to get the IssuerDomain and ClientIdD authentication property values.

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.

IssuerDomain. Specifies the Auth0 domain URL, for example, https://dev-pi-security.eu.auth0.com. This field is automatically populated when you click the *Get Auth properties* button.

ClientID. Specifies the client ID. This field is automatically populated when you click the *Get Auth properties* button.

Note The Auth0 configurations listed above are updated based on the customer's environment using authentication properties available in the qad-qxtend-qxi.properties file, located at the following location: \servers\tomcat-qxtend\webapps\qxi\WEB-INF\conf\qad-qxtend-qxi.properties. IssuerDomain and ClientID values are used from the Auth0 native application for authentication. Excelerator uses the exceleratorIssuer and exceleratorClientId properties.

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

  It is the system default worksheet template type and 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 the target system to generate key field values or business document numbers (for example, sales order numbers).

  This feature is realized by using 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.

Debug Level. Specify the debug level for financial QDocs only. Refer to Table 2.1 for the available level options.

Table 2.1
Debug Levels

Level	Description
1	Logs the start and end of all entry-level business methods. Entry-level methods are called from outside the business logic.
2	Logs the start and end of all methods and procedures, internal and external.
4	Logs parameter values of business methods. This level is only available when level 1 or level 2 is active.
8	Logs all read or update database queries.

Level	Description
16	Logs details of all database update (create, modify, or delete).
32	Analyzes performance and logs possible memory leaks in the business code.

Scope Transaction. Select this option if you want each update from Excelerator to be treated as a whole transaction in QAD. If any error is encountered, the entire update will be backed out.

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 processes 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 ignores 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. Enter zero (0) to retrieve all matching rows.

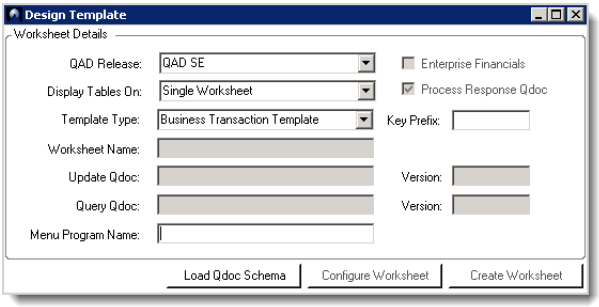
Filter. Enter a valid Progress query to use as filter criteria to filter records. The filter must be a valid Progress ABL WHERE clause and can only be defined against the top-level table of the QXO business object; for example:

```
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.

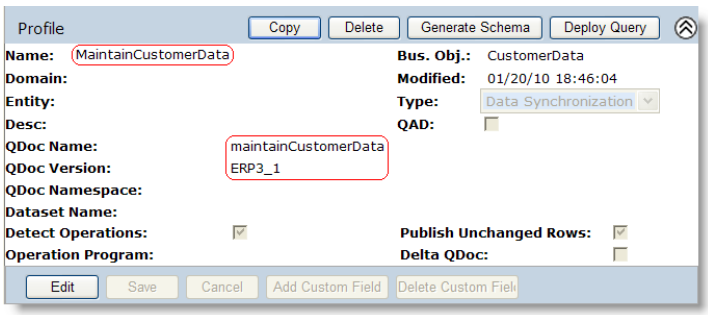
Fig. 2.7
Design Template



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. It means that 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



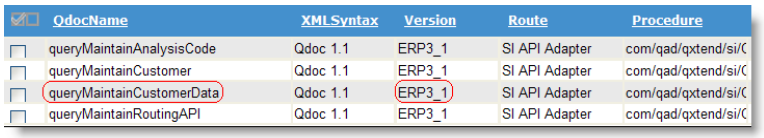
After the MaintainCustomerData profile has been deployed as a query to QXtend Inbound, it is available for use with the worksheet.

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

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



QdocName	XMLSyntax	Version	Route	Procedure
queryMaintainAnalysisCode	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/(
queryMaintainCustomer	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/(
queryMaintainCustomerData	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/(
queryMaintainRoutingAPI	Qdoc 1.1	ERP3_1	SI API Adapter	com/qad/qxtend/si/(

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 situation 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 18.

If you select the business transaction template type, specify the Key Prefix value and the Process Response QDoc option. These default from the QAD QXtend Excelerator settings. See “Key Prefix” on page 18 and “Process Response QDoc” on page 19.

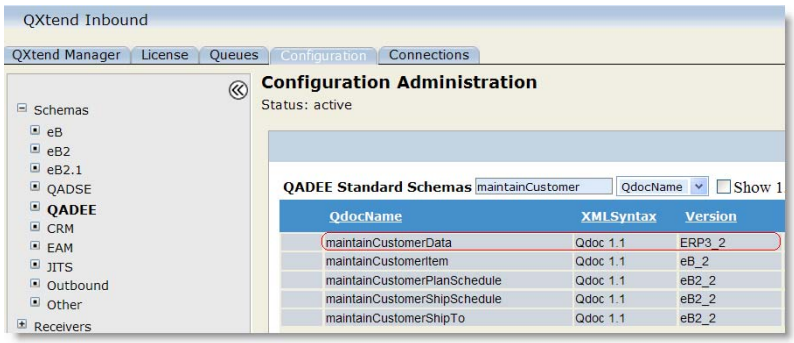
5 Click Load QDoc Schema, and then locate a QDoc file to load into the worksheet.

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 that 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).

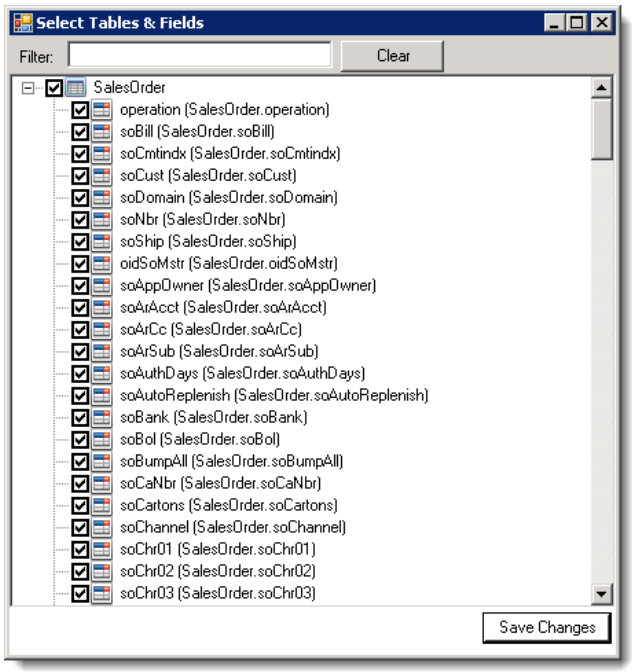
6 Optionally, click Configure Worksheet to customize the schema in the worksheet. In the Select Tables and Fields window, select the tables and fields you want in the worksheet. You can use the Filter to quick search for fields. The tool automatically select table primary keys.

Note Column labels are only retrieved for 2009 and later releases of QAD EE and QAD SE.

To enable this functionality, configure a connection pool of type SIAPI in QXtend Outbound. The name of the connection pool must be same as the Update Receiver specified in the Excelerator Settings panel.

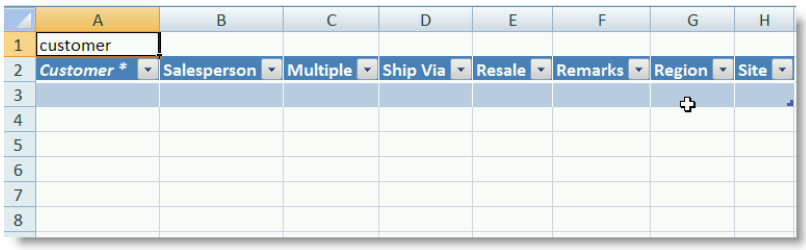
For details on configuring connection pools, see *User Guide: QAD QXtend*.

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 in the QDoc; all other fields are sorted in alphabetical order.

Fig. 2.12
Worksheet with Schema



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. You can import the CSV file into the Excel worksheet for data synchronization with the target QAD application.

To create a CSV template:

1 Under the Developer tab in the Ribbon, click Create CSV Template.

2 Select one of the following:

- Single CSV Template: creates one CSV file for all the tables in the worksheet.
- Template Per Table: creates multiple CSV files, one for each table in the worksheet.

3 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.

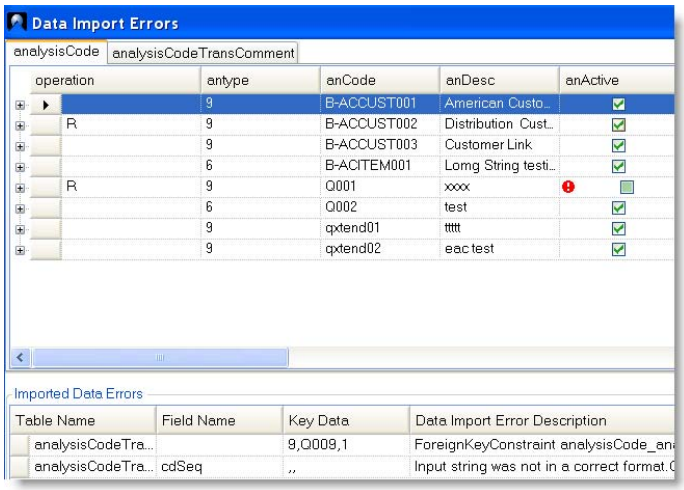
Using the CSV export tool, you can export data from the Excelerator worksheet into a single CSV file or multiple CSV files, one for each table. The structure of the exported CSV file strictly follows that defined in the imported CSV file.

Performing Initial Data Load

For go-live implementation of QAD Enterprise Applications, 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. Correct these errors before you can proceed with data import.

Fig. 2.13
Data Import Errors



5. If necessary, 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.

Chapter 3

Using QAD QXtend Excelerator

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. It 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.

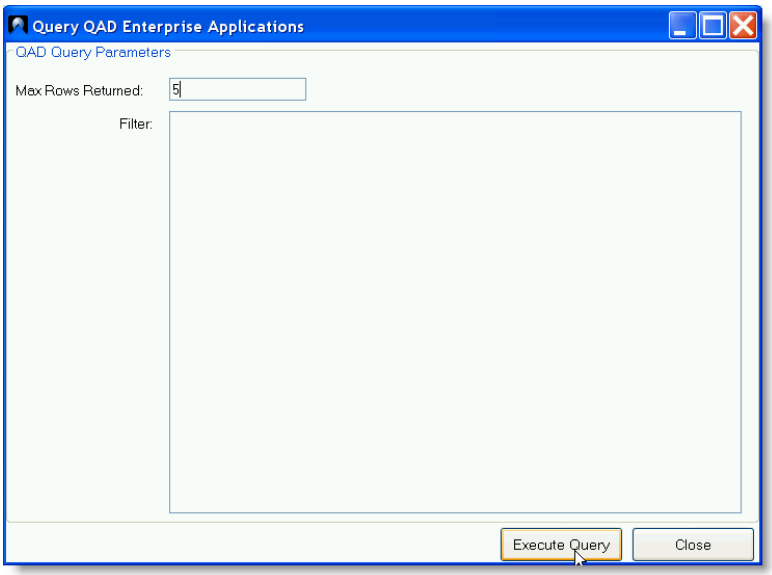
2 In the Query QAD Enterprise Applications window, define the query parameters.

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

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:

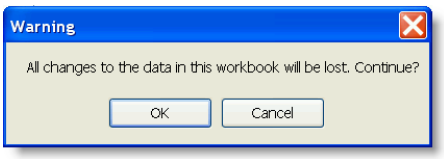
```
cm_site = 'SITE1' and (cm_addr >= '0100' and cm_addr <= '100')
```

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. The `qdocResponses.log` file records 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 the profile defined in QXtend Outbound. All these names are case-sensitive. Map them exactly 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

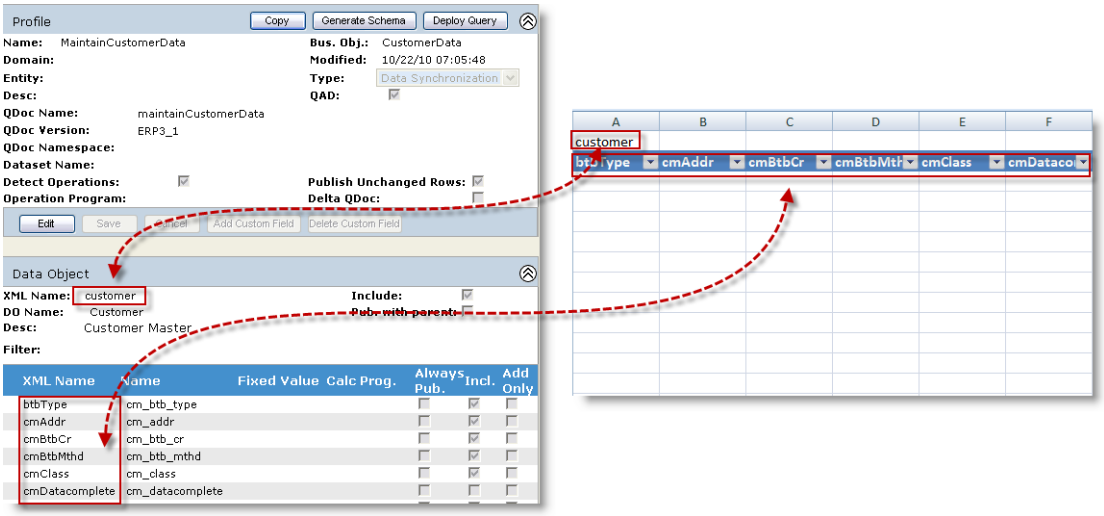
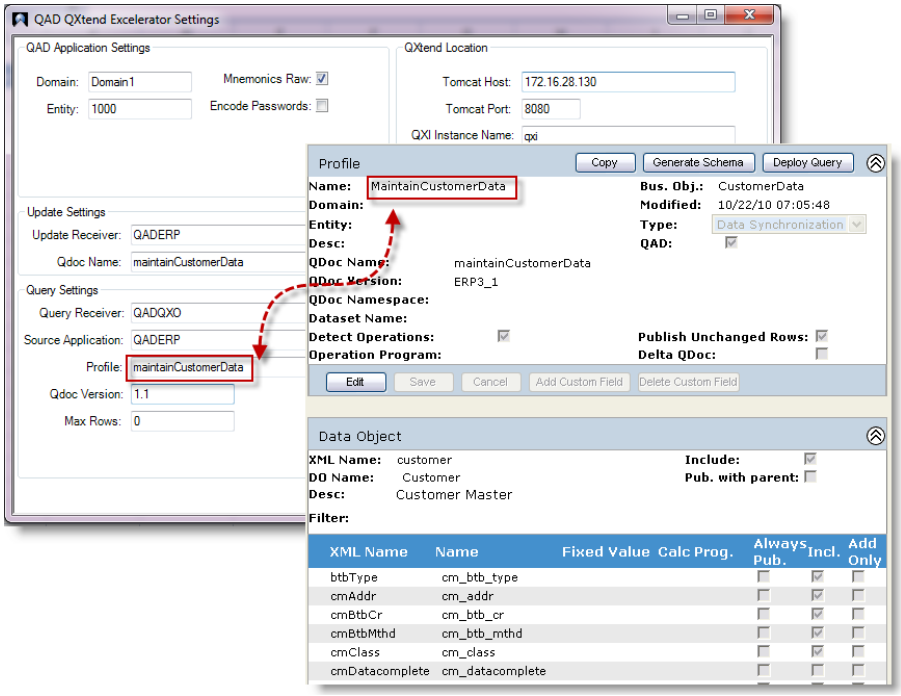


Fig. 3.4
Mapping of Profile Names



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 cause 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 that 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.

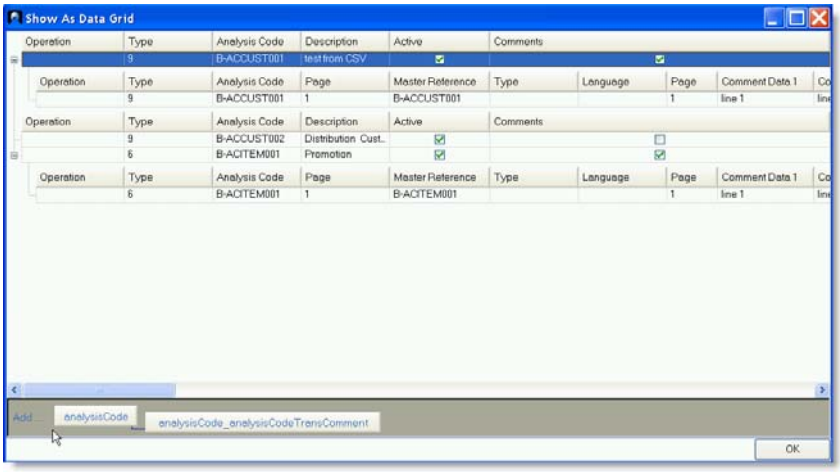
Viewing and Modifying Data

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. If you enter invalid data, error occurs when you update records to the target QAD application.

Alternatively, when the Enable Data Grid View option on 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.

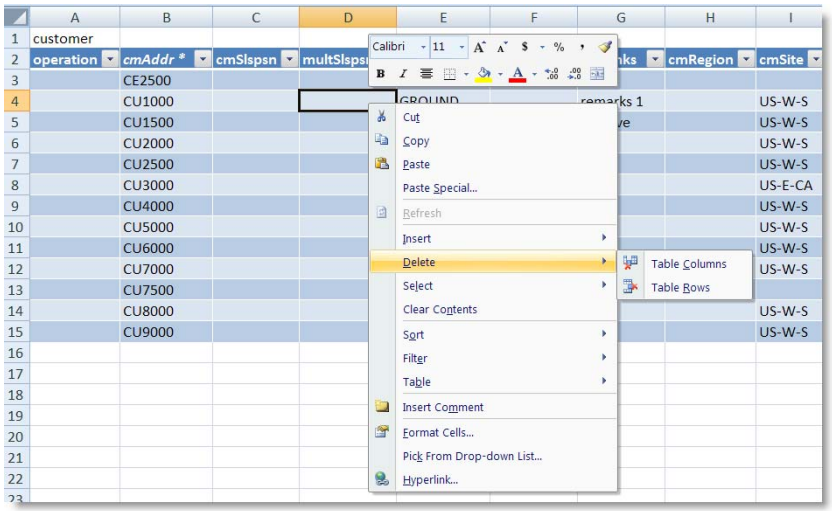
Fig. 3.5
Data Grid View



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 operation 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 cannot delete corresponding records from the target QAD Enterprise Applications when you perform data update. To remove records from the target application, populate the “operation” column with R (for Remove) in corresponding rows to indicate to QXtend Inbound to delete the records when processing Update QAD.

Fig. 3.6
Deleting Records

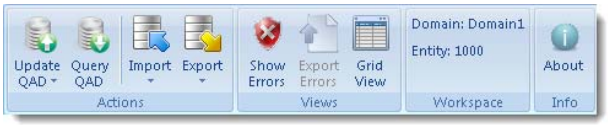


Changing Workspace

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

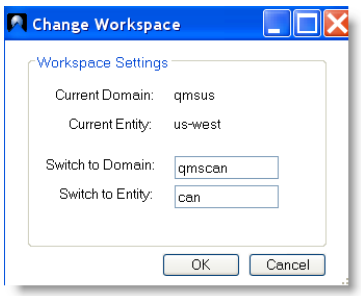
The current workspace for the worksheet is displayed in the Workspace group under the QAD Tools tab.

Fig. 3.7
Current Workspace



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

Fig. 3.8
Change Workspace

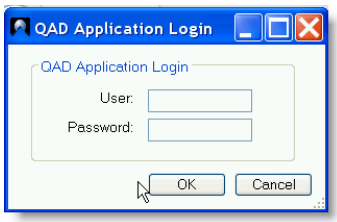


Updating Data to the Target QAD Application

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

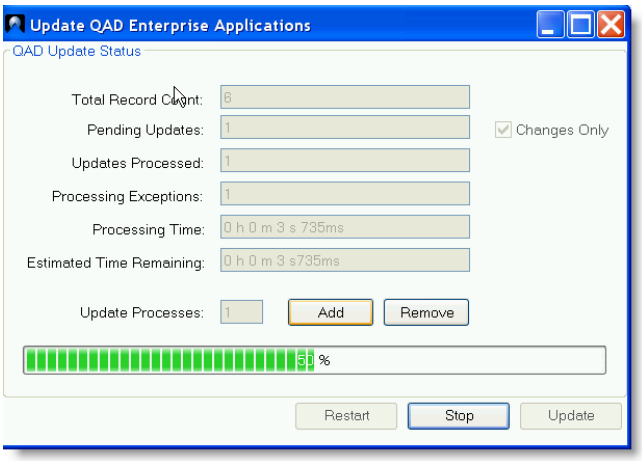
1. Under the QAD Tools tab, click Update QAD and select from one of the following options:
   - Process Changes: Only submit records that have been modified.
   - Process All: Submit all records in the worksheets.
2. A QAD Application Login window appears. Provide log-in information to access the target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

Fig. 3.9
QAD Application Log-In



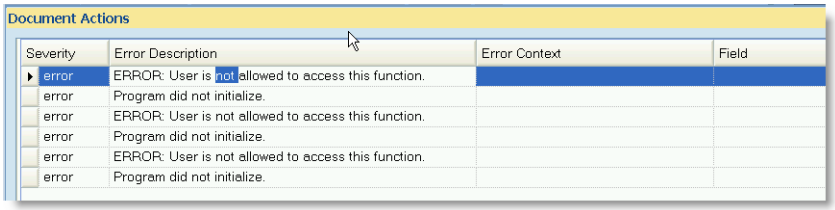
3 A QAD Update Status window appears with detailed update status information and progress bar. You can stop and restart the process as needed.

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 on and off the error information 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. You can use the Export Errors button to export errors to a text file and save it to your specified location for further investigation.

Fig. 3.11
QAD Update Status



Product Information Resources

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

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

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

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

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

*Log-in required

Index