QAD Adaptive Applications

Technical Reference

QAD EQMS Integration

70-3474-2025.1

QAD EQMS 2025.1

September 2025

Copyright

QAD Inc.

100 Innovation Place

Santa Barbara, CA 93108

Phone: +1 (805) 566-6100

http://www.qad.com

QAD EQMS Integration Technical Reference
Change Summary

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

Date/Version	Description	Reference	Changed By
SEPT 2020/v2020.1	Initial upload	--	J3O
SEPT 2021/v2021.1	Updated the Appendix	p. 42	J3O
SEPT 2022/v2022.1	Updated versioning	--	BHH
MAR 2023/v2023	Updated versioning	--	BHH
MAR 2024/v2024	Updated versioning	--	BHH
JUN 2024/v2023	Updated Step 6	p. 23	BHH
JUN 2024/v2024	Updated EQMS screenshots	--	BHH
SEPT 2024/v2024.1	Updated versioning	--	BHH
MAR 2025/v2025	Updated versioning; Modified the "Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration" section. It is now Level 1; two additional levels have been added; Rearranged sections to expand the final chapter, which is now titled Additional Information.	p. 12, p. 29, p. 38, p. 39	NQC
SEPT 2025/v2025.1	Updated versioning; Modified Step 4 of the Level 1 instructions, "Set Up QAD EQMS for Readiness to Configure Integration in QXtend".	p. 17	NQC

Chapter 1

Introduction to QAD EQMS Integration

Introduction

This document is meant to serve as a comprehensive guide for setting up and configuring out-of-the-box QAD EQMS <=> QAD Adaptive ERP integration, as well as creating and troubleshooting custom integrations for QAD EQMS.

About This Guide

This user guide focuses on:

- Prerequisites for setting up custom and OTB (out-of-the-box) integrations.
- Limitations of QAD EQMS integration services.
- Integration capabilities for QAD EQMS
    - Esync
    - SOAP
    - REST
- Standard out-of-the-box integrations specifications for QAD EQMS <=> QAD Adaptive ERP integration.
    - See an integration specification sample here.
- Creating custom integrations.
- Troubleshooting integrations and common problems.

Prerequisites for QAD EQMS Integration

User Roles

Users should be capable of installing QAD EQMS, interpreting technical reports, and following a proper SDLC process. Users should have an intermediate knowledge of QAD EQMS Integration Configuration, QXtend Outbound UI, IIS, Microsoft SQL Server, TSQL, and Excel.

Access and Prerequisite Information Needed

To force publishing of financial data, the following information is required:

- Access to the QAD .net UI clients
- Admin login credentials (to access the QXtend related screens, event screens, and all regular data screens)

Additionally, you must have QXtend inbound and outbound URLS, as well as the credentials to log in to these applications.

> ***Note:*** The inbound URL is necessary even if you are not configuring integration from QAD EQMS to QXtend; the method used to automatically configure QXtend Outbound is based on a web service call to send to QXtend Inbound.

Integration Workflows

See "Appendix" on page 42 for links to design documentation for some of the more complex 2.0 integrations. This documentation includes workflow information and explains which parameters

and settings must be customized or changed in order to make the integration work.

Compatibility Constraints

QAD EQMS web services (including both custom and standard services) support integration with SOAP 1.1 or 1.2. QAD EQMS also supports custom outbound REST integration.

QAD QXtend version 1.8.3 is required in order for the automated QXtend outbound configuration commands to work. QXtend must be installed correctly and linked to the appropriate databases and source applications prior to running the commands, or they will not work.

QAD EQMS has standard interfaces that support integration with most versions of both QAD EE and QAD SE. It is imperative that QXtend 1.8.3 or later is installed and linked to the source application correctly.

It is possible to integrate with earlier versions of QXtend (as long as they support SOAP 1.1 or 1.2) but the automated configuration commands are not usable with these versions. Therefore, QXtend would need to be configured manually.

There are two versions of QAD EQMS' outbound service in the Inspection Events process that transfer inventory in QAD. Which one is used depends on which standard QDoc is loaded on the QXtend inbound side (This must be checked manually, as it cannot be assumed which version is used based on the QXtend version).

- If QXtend inbound has the QDoc “transferInvSingleItem-ERP3_1” loaded, then the QAD EQMS service “qadInvTransferSingleV2” would be compatible.
- If QXtend inbound has the QDoc “inventoryTransSingle-ERP3_1” loaded, then the QAD EQMS service “qadInvTransferSingle” would be compatible.

Capabilities of QAD EQMS Integration

QAD EQMS offers two forms of configurable integration, which serve two specific needs: Web Services and Esync.

> ***Note:*** Almost all core services are SOAP web services. Currently, configuring REST services to replace the SOAP ones requires a custom services engagement.

Web Services

Web Services are designed to move data to and from QAD EQMS on a push/pull basis.

Call Service

A call service can be configured to call a SOAP or REST service from an external service or application.

Mappings can be established between the service’s input and output parameters to and from the QAD EQMS process making the call.

Call services are executed (called) on, except for the process for which they are configured.

Create Service

A create service can be configured to create QAD EQMS SOAP web services that can be called from an external service or application.

Methods can be configured for insert, update, insert or update, remove, and select actions. Input parameters can be defined and mappings can be configured.

RESTful APIs

Every QAD EQMS business object (or process, as they are referred to within QAD EQMS) has exposed RESTful APIs.

For each API, the following request types are exposed:

- Search Results (a collection of records)
    - Get
- Details (a specific record)
    - Get
    - Put (update existing)
    - Post (add new)
    - Delete

Each request requires an authentication token, which can be retrieved through a call to the QAD EQMS authentication service. Once obtained, the token (a JWT token) is stored on the client for the length of the session and passed with any subsequent REST service requests. All user permissions are applied based on the authentication token.

Error codes are sent, but detailed errors are not sent for security purposes.

Despite the fact that the application uses RESTful APIs for its own architecture, these APIs are not optimized for use other than by the QAD EQMS application.

Esync

Esync is used for initial and scheduled data loads. It is designed to pull large quantities of data into QAD EQMS on a schedule. As with the web service integrations, Esync events are configured by QAD EQMS processes.

Esync events are one way integrations. The available actions are:

- Insert
- Update
- Update or Insert

Esync events can be configured to pull information via:

- ODBC
- OLEDB
- Oracle
- SQL Client

- Web Method
- LDAP

Esync events can be scheduled to run on daily or weekly patterns at certain times of day or multiple times in a day.

Encryption

In Transit

All integration communications are encrypted over Transport Layer Security (TLS) otherwise known as HTTPS (HTTP over TLS).

Tokens

For authentication, QAD EQMS uses a JSON Web Token (JWT). JWT is a JSON-based open standard (RFC 7519) for passing claims between parties in web application environments. JWT relies on other JSON-based standards: JWS (JSON Web Signature) RFC 7515 and JWE (JSON Web Encryption) RFC 7516. The tokens are Base64 encoded.

Limitations of QAD EQMS Integration

Version 2.0/3.0 integrations will not work out of the box; they can only be a template due to the specifics of each customer implementation. These integrations require a discovery by a Services consultant and a Services engineer. Additionally, they require a QXtend SME to help set up the Qdoc and any back-end (e.g. Linux, SSL certificate).

Chapter 2

Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration (Level 1)

Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration (Level 1)

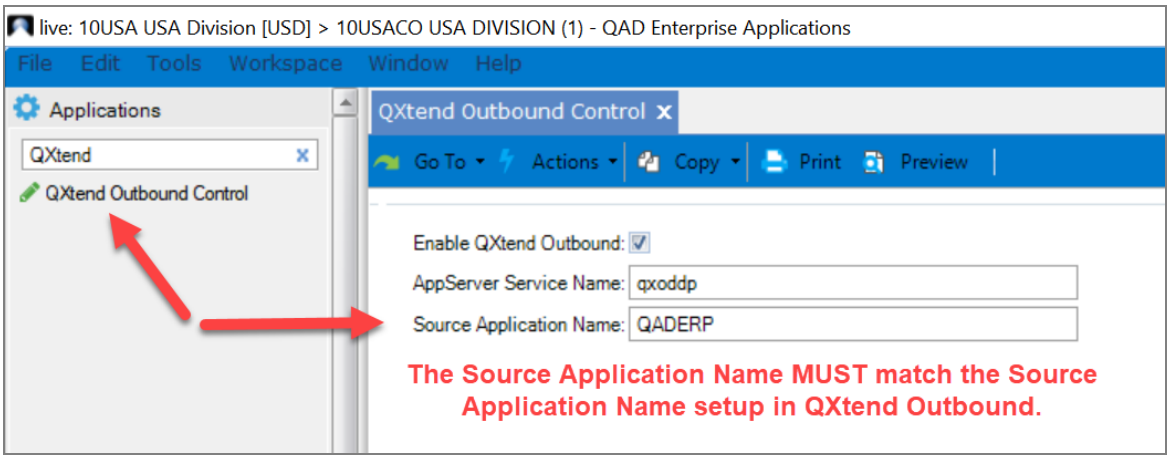
Step 1

Ensure QXtend Outbound Source Application Name in QAD .Net UI Client Matches Information in QXtend Outbound

The ERP application and QXO must be communicating before you can proceed.

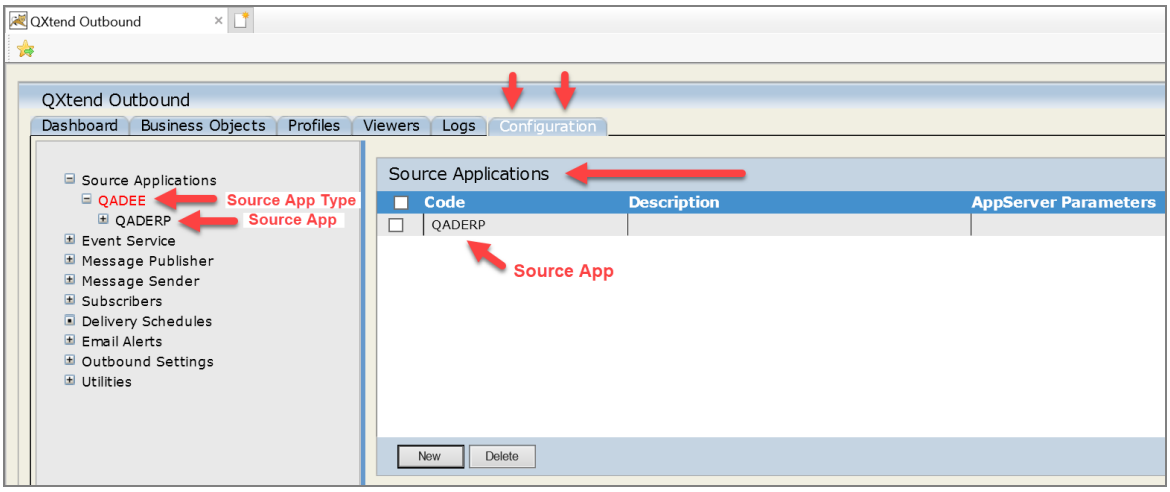
To begin, find the source application name in the QAD Net UI client in the QXtend Outbound Control. See *Fig. 1: QAD Net UI Client, Source App Name location* below.

Fig. 1: QAD Net UI Client, Source App Name location



In QXtend Outbound, navigate to the Configuration tab to locate the source application that corresponds to the previous screen shot. The source application name can be anything, as long as it matches between these two locations.

Fig. 2: QXtend outbound, Source App Name location



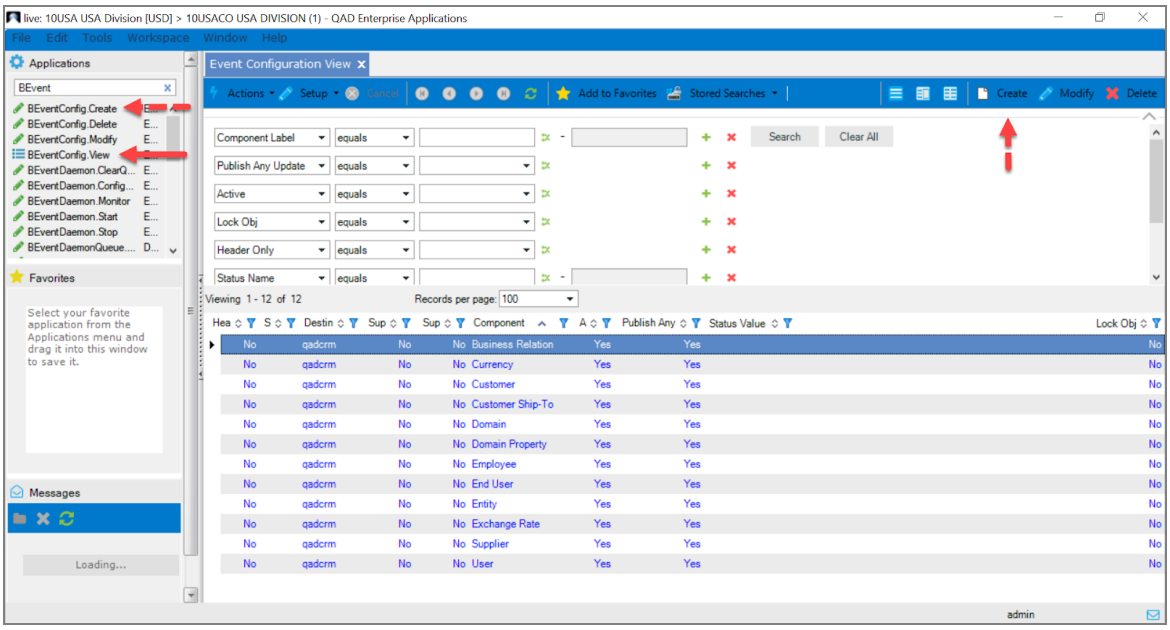
Step 2

(EE ONLY) Verify Business Events are Configured

If you are configuring integration for QAD EE, continue this step. If you are configuring integration for QAD SE, skip this step.

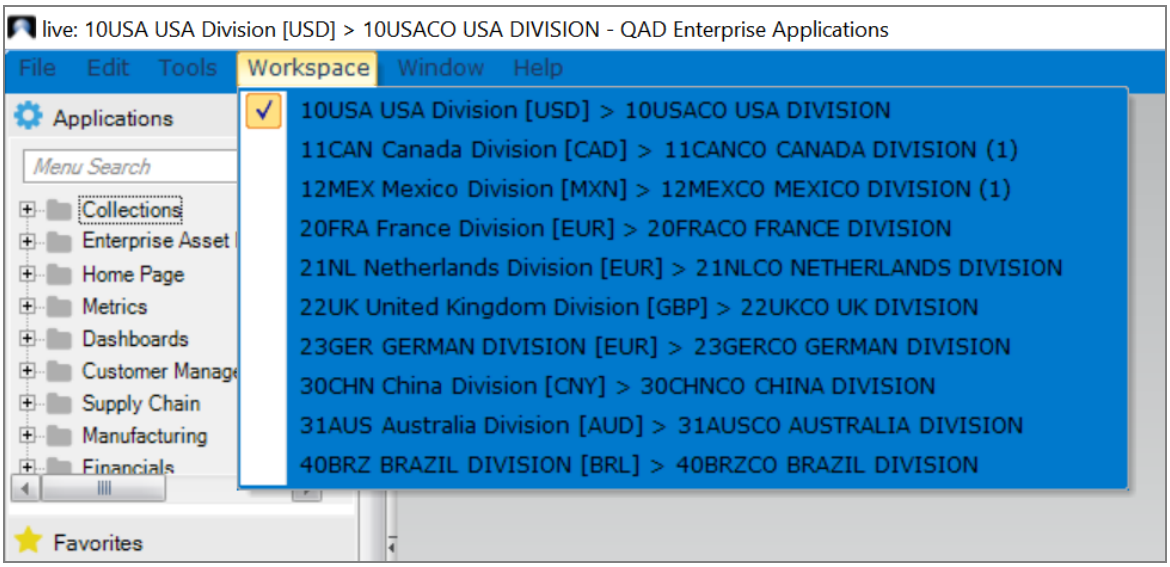
Navigate to the "BEventConfig. View" application in QAD .Net UI. Verify that the Business Events are configured so that the financial data will work.

Fig. 3: QAD .Net UI Client, Event Configuration View tab



These BEvent records must be added separately in each QA workspace (i.e. domain). Click the Workspace drop-down menu at the top of the QAD .Net UI screen to view a list of available domains/workspaces. When you select an item from the list, you will log into that specific workspace.

Fig. 4: QAD .Net UI screen, Workspace menu

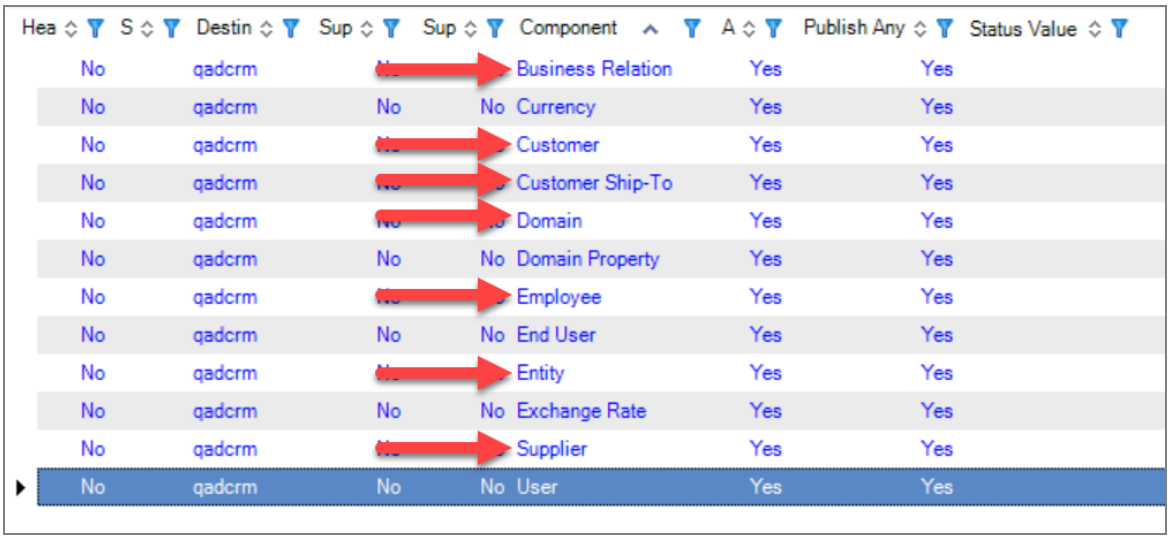


Several interfaces must be performed separately in each workspace from which QXtend should send data. This includes the following interfaces:

- Business Relations (BBusinessRelation)
- Customers (BDebtors)
- Customer Ship-To (BDebtorShip-To)
- Domains (BDomains)
- Employee (BEmployee)
- Entity (BCompany)
    - Entity is the latest standard interface
- Shared Sets (BSharedSet)
- Supplier (BCreditors)

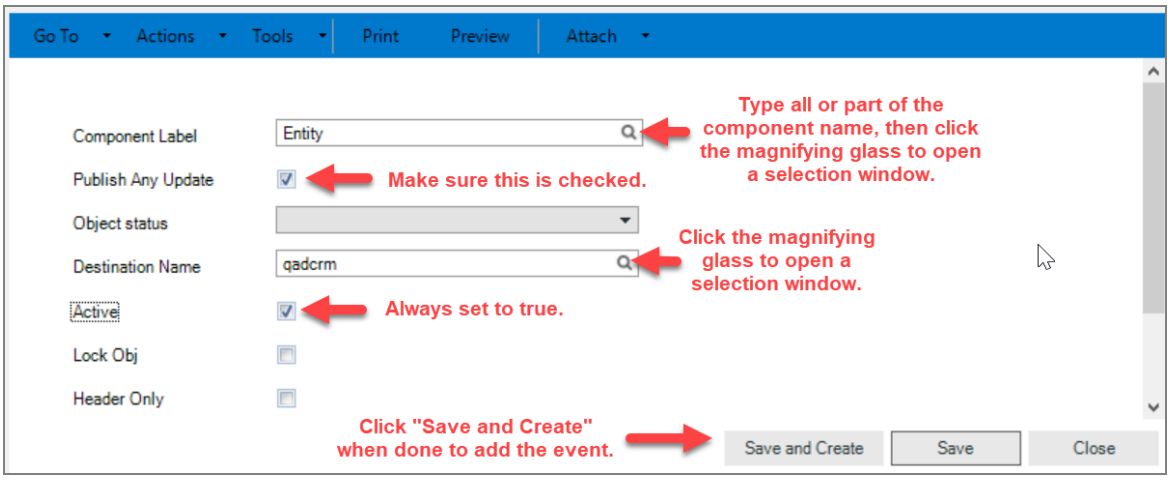
Most of these interfaces should be set up when QAD installs QXtend, but this does not include Entity, which should be manually added. Additionally, the other records may need to be manually added if something was missed during the install process. See *Fig. 5: QAD .Net UI Client, Event Configuration View tab* below, which shows which records should be configured.

Fig. 5: QAD .Net UI Client, Event Configuration View tab



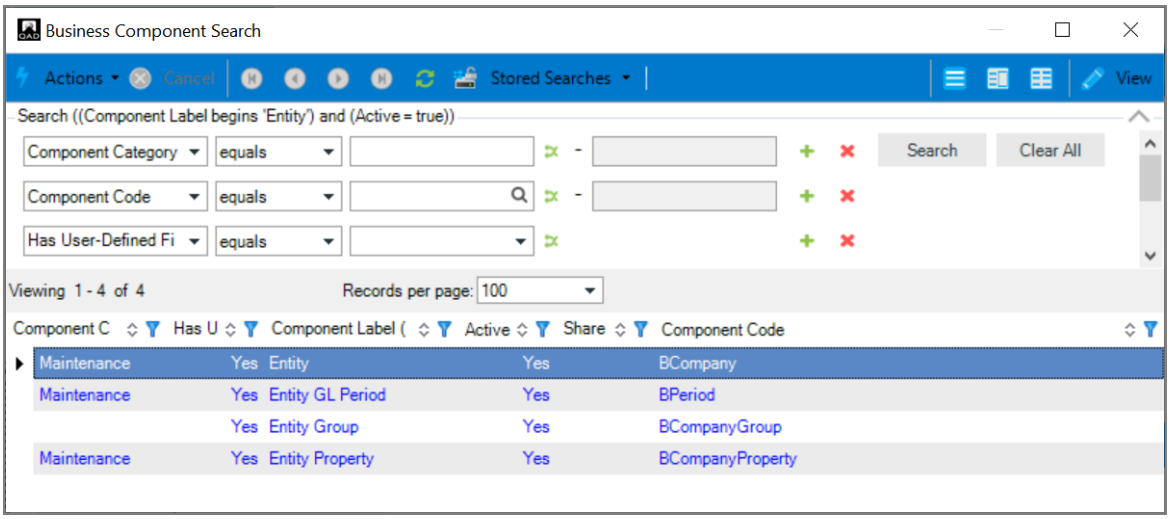
If you are missing events in the BEventConfig. View screen, you can easily add them by clicking the Create button in the BEventConfig. View screen, or by navigating to the BEventConfig. Create screen. Doing so will present the following information, which should be completed as shown below.

Fig. 6: BEventConfig. Create screen



Click the magnifying glass look-up on the Component Label field to open the Business Component Search. Double-click the row you wish to add as your event component.

Fig. 7: Business Component Search screen



Likewise, you must use the look-up on the Create screen's Destination Name field and double-click the proper app server connection before you can save and create the new event.

If you want Entity Integration (the latest standard interface), then you must set up the Entity event on the BEventConfig screen. It is not usually configured by default, and therefore must be manually added to the BEventConfig screen in the .Net UI client.

Step 3

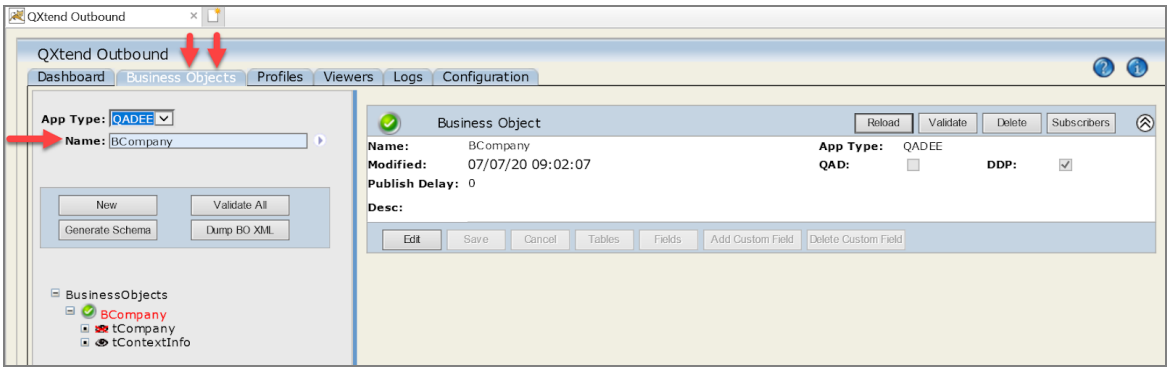
(EE Only) Verify the Entity Business Object is Included with QXtend Outbound

If you are configuring integration for QAD EE, continue this step. If you are configuring integration for QAD SE, skip this step.

You must determine if the bCompany (i.e. Entity) Business Object is included with QXtend Outbound. If it is not, then it must be imported manually; this should be done prior to running any of the commands below from QAD EQMS.

To determine if bCompany is included in QXtend Outbound type bCompany into the Name field and click the Search button. The following results will be returned if it is present:

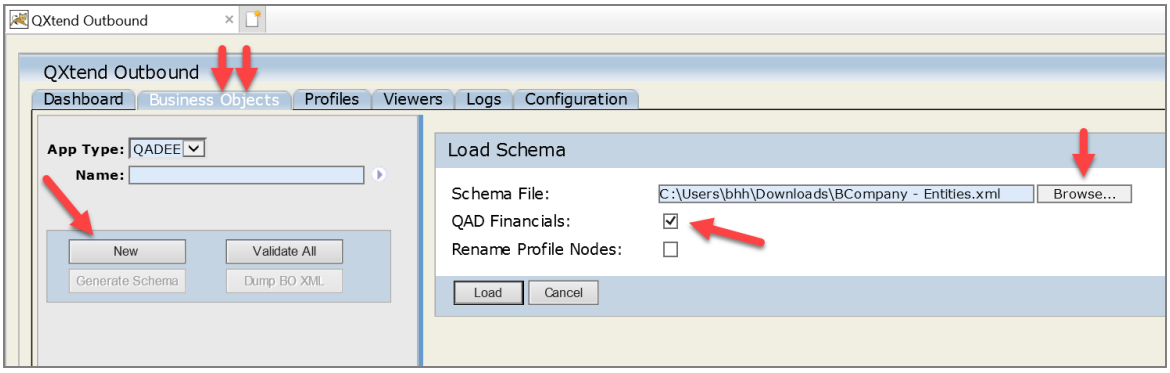
Fig. 8: QXtend Outbound screen



If bCompany is not already included, take the following steps to import it:

1. Download the bCompany (Entity) schema file from here.
2. Navigate to the Business Objects tab in QXO (QXtend outbound URL) and click the New button.
3. Click Yes when QXO prompts you asking if you want this to be a Direct Data Publish BO.
4. In the Schema File field, enter or browse to the path where you downloaded the BCompany schema file.
5. Select the check box for QAD Financials, then click the Load button. This concludes the creation of the custom Direct Data Publish Business Object for Entity.

Fig. 9: QXtend Outbound, Business Objects tab



From here, the QXtend and QAD .Net UI Client should be ready to receive the SOAP messages from QAD EQMS, which will automatically configure the integration in QXO.

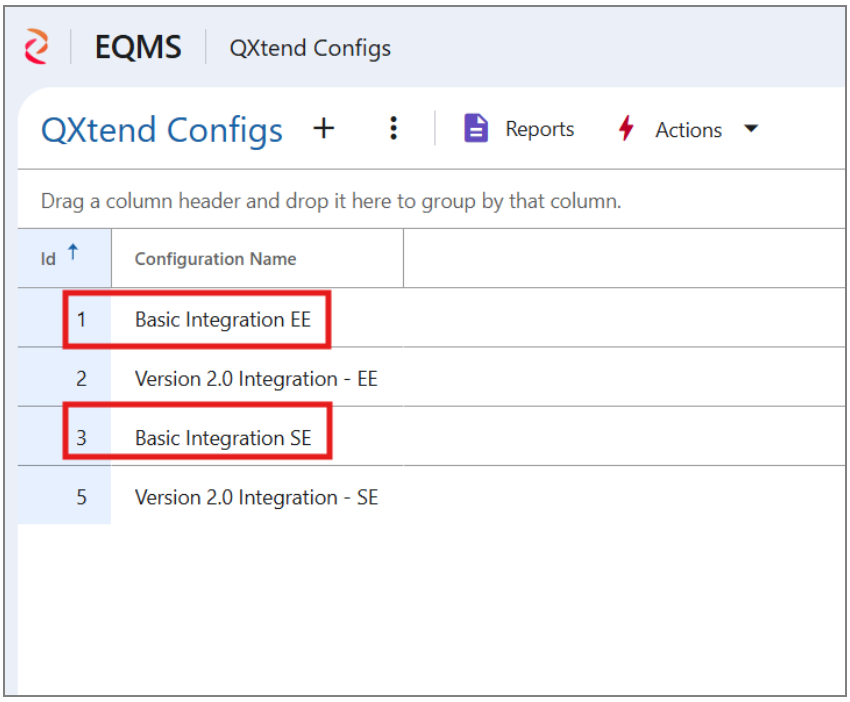
Step 4

Set Up QAD EQMS for Readiness to Configure Integration in QXtend

This step is intended to prepare the QAD EQMS for configuration integration in QXtend. First, the Source App Type and Source App Values in QAD EQMS must be updated to match the content in QXtend. Follow the instructions below to accomplish this:

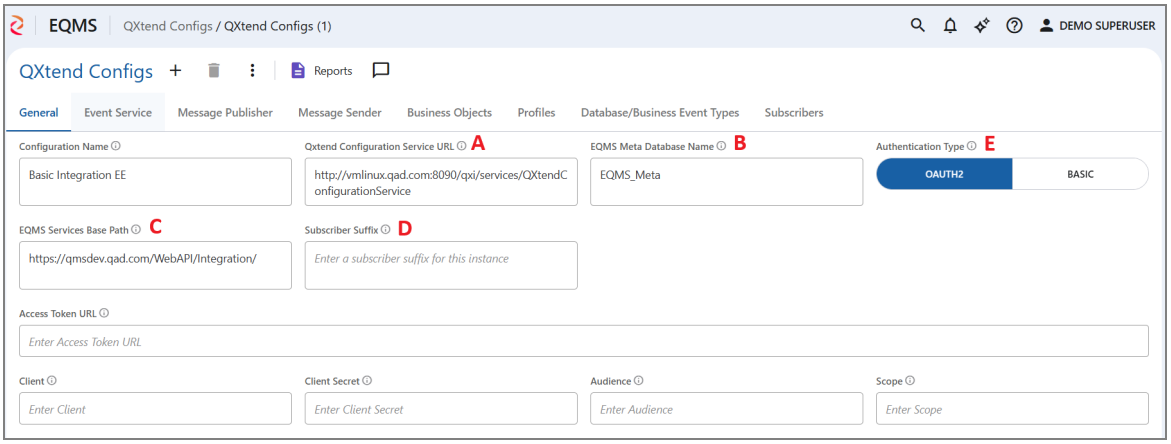
1. Open the QXtend Configs process search screen in QAD EQMS. Four records appear: two labeled for EE, two labeled for SE.
2. Choose between "Basic Integration EE" or "Basic Integration SE", depending on which you are configuring.

Fig. 10: QAD EQMS, QXtend Configs search screen



3. Open your selected record. There are six fields to verify on the General tab of this screen.

Fig. 11: QXtend Configs screen, General tab



A. Ensure the "QXtend Configuration Service URL" is set to the current QXtend integration URL.
B. Ensure the meta database name matches the actual one used by the QAD EQMS site. The meta database name can be discovered by logging into the QAD EQMS site's AdminTools. Select the three-dot menu in the upper-right corner and choose About.
C. Ensure the EQMS Service Base Path field is set to the current QAD EQMS sites integration URL. This consists of the base site URL and the WebAPI name from IIS,

followed by "/Integration/". Here is an example URL: *https://qmsdev.qad.com/webapi/Integration/*

You MUST include the slash at the end of Integration or it will not work. You can test this URL by appending "qadItems.asmx?wsdl" to the end of the URL (for example: *https://qmsdev.qad.com/webapi/Integration/qadItems.asmx?wsdl*). This should take you to an XML document (aka the wsdl). If it does, then the URL is correct and you can move on.

D. Include the subscriber suffix if used.
E. Select the authentication type to be used. Use OAuth 2.0 Authentication (for Qxtend 2.1+). This toggle allows you to select the authentication method used to connect to your Qxtend instance. The correct method depends on the version of Qxtend you are using.
    i. **Authentication Type: OAUTH2**

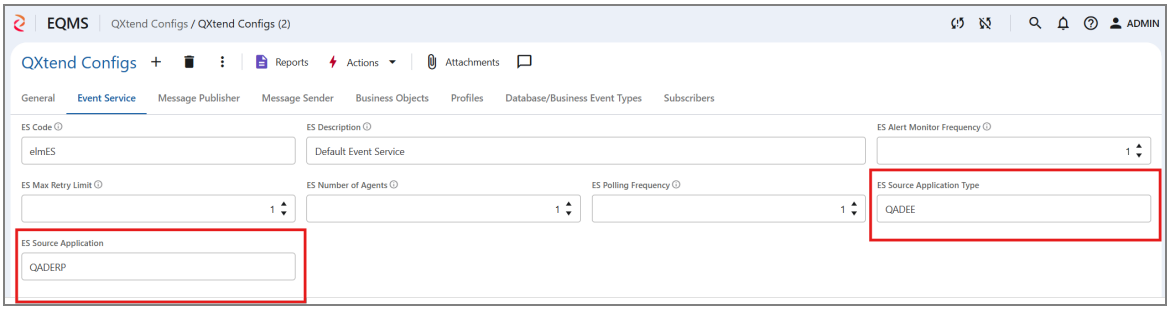
    Use when: You are connecting to a Qxtend version 2.1 or newer. The application will use the OAuth 2.0 flow to authenticate. You will be required to provide the Access Token URL, Client ID, Client Secret, and other related OQuth 2.0 fields.

    ii. **Authentication Type: Basic**

    Use when: You are connecting to a Qxtend version older than 2.0. The application will use Basic Authentication. You will be required to provide a standard Username and Password for your Qxtend account.

4. Navigate to the Event Service tab. Ensure the ES Source Application Type and ES Source Application match the values from Step 1 on page 12. These should be the Source Application Type and Source Application name from QXtend Outbound configuration tab, respectively.

Fig. 12: QAD EQMS, QXtend Configs process screen, Event Service tab



5. Update the QAD EQMS references pointing to the QXtend Inbound URL to the correct URL. This can be done manually, or it can be done automatically by running the command "0 - Update Source Application Type" from the QXtend Configs detail screen in which you have been making changes.

    Under the Actions menu, you will see a list of several commands, but only command zero should be run. After running this command and before running any further

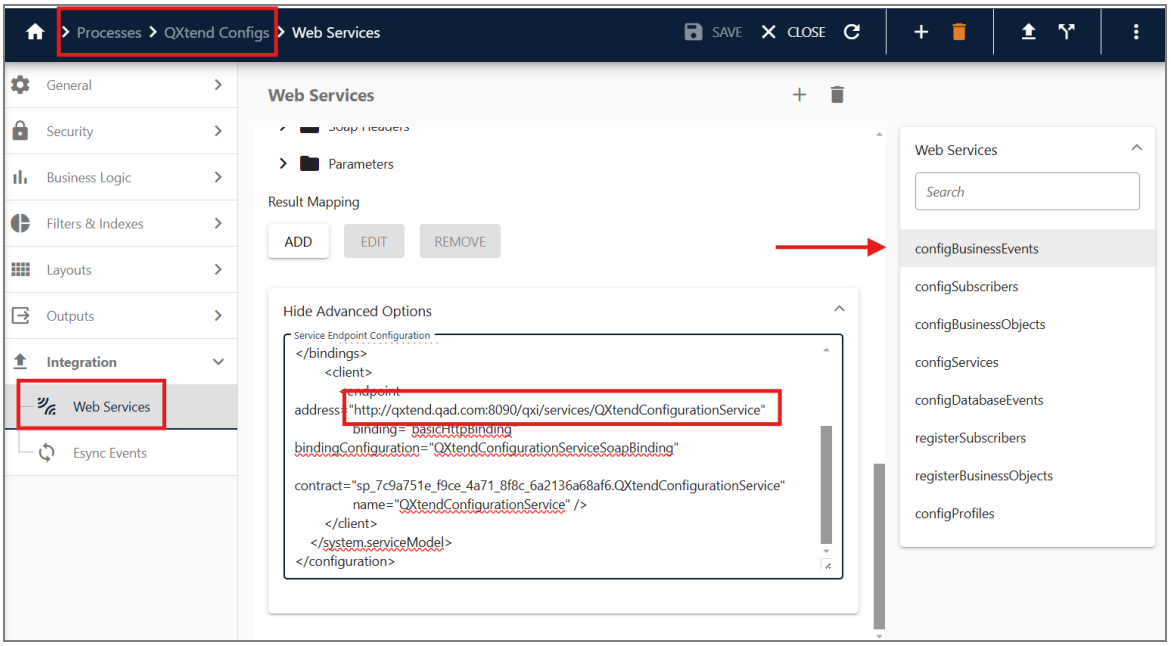
commands, you must still publish and deploy by following the manual setup directions below.

Manual Process for Updating QXtend Inbound URLs in QAD EQMS

1. Open the QXtend Configs process in QAD EQMS AppXtender.
2. Navigate to the Integration section and choose Web Services. You will see several pre-configured services.
3. Extend the process and click the paperclip icon for each service to extend them and make them editable.
4. For each service, click Show Advanced Settings, which reveals a text box labeled Service Endpoint Configuration. In this text box only, reset the existing placeholder QXtend inbound URL to the correct QXtend inbound URL, provided by the customer or QAD. Only change up to "qxi" ("http://vmlinux.qad.com:8090/qxi/"). Do not change any other URLs in the integration section or take any other action.

   Do this for each integration service in the QXtend Configs process, as well as in each of the following processes: QXtend Business Object Configs, QXtend Profile Configs, and QXtend Database Event Configs. An example is below.

Fig. 13: AppXtender, QXtend Configs process screen, Web Services tab



Whether you updated the URLs manually or automatically using command zero, you must publish the following processes in AppXtender in the following order:

1. QXtend Configs (soft publish)
2. QXtend Business Object Configs (soft publish)
3. QXtend Profile Configs (soft publish)
4. QXtend Database Event Configs (publish and deploy)

At this point, both QXtend and QAD EQMS are ready to auto-configure the integration interfaces.

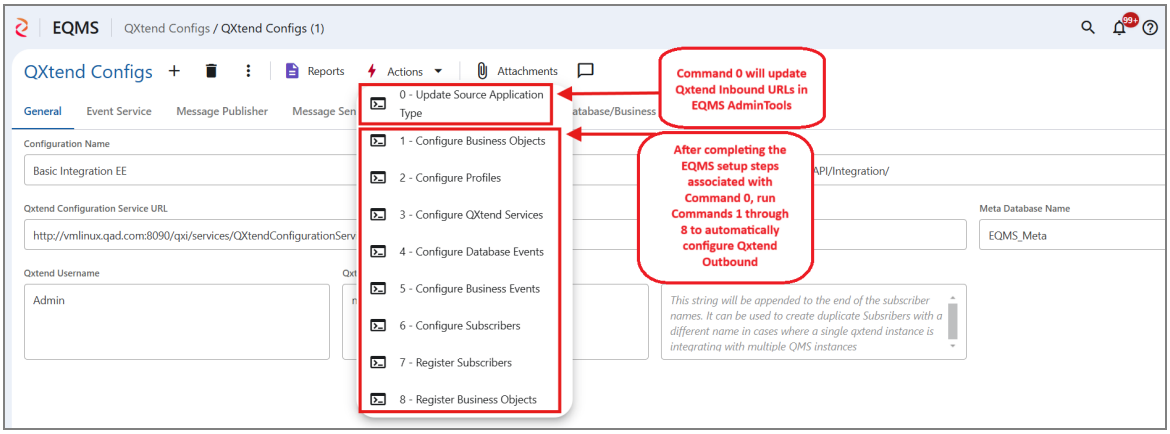
Step 5

Configure QXtend Outbound from QAD EQMS

Return to the Basic Integration EE/SE record from the QXtend Configs process search screen in QAD EQMS (see "Step 3 " on page 16). Double-click the process to enter the record's detail screen.

On this record, you will see a numbered list of commands. Run each remaining command from top to bottom (lowest to highest). This action configures QXtend Outbound automatically. Once it is finished, integration should be configured and functional. If any errors appear when running these commands, then you must troubleshoot them before you can move on.

Fig. 14: QAD EQMS, QXtend Configs process screen, Commands list



The best way to confirm success is to check the Configuration tab in QXtend Outbound URL. Look under Subscribers and ensure a subscriber exists for each of the following:

For EE:

- elmDebtorShipTo
- elmEntitiesUpsert
- elmCustomersBRelUpsert
- elmUofMUpsert
- elmWorkCentersUpsert
- elmSuppliersUpsert
- elmSupplierItemsUpsert
- elmSitesUpsert
- elmProdLinesUpsert
- elmLocationsUpsert
- elmItemsUpsert
- elmItemTypesUpsert
- elmItemGroupsUpsert
- elmInvStatusCodesUpsert

- elmDomainsUpsert
- elmDepartmentsUpsert
- elmProcessesUpsert
- elmSuppliersBRelUpsert
- elmCustomersUpsert
- elmCustomerItemUpsert

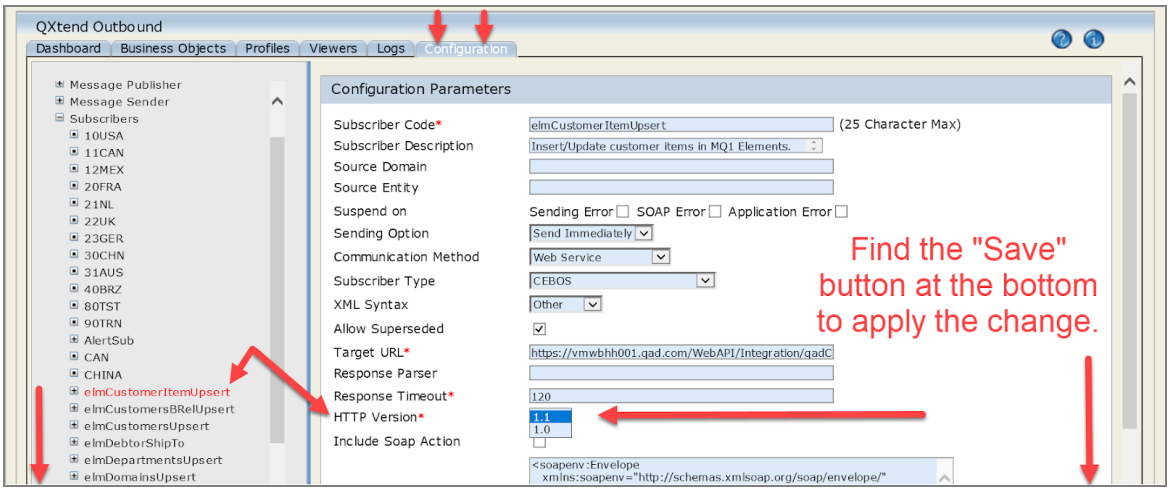
For SE:

- ElmEntities
- ElmCustomerItemUpsert
- ElmProcessesUpsert
- ElmDepartmentsUpsert
- ElmInvStatusCodesUpsert
- ElmItemGroupsUpsert
- ElmItemTypesUpsert
- ElmItemsUpsert
- ElmLocationsUpsert
- ElmProdLinesUpsert
- ElmSitesUpsert
- ElmWorkCentersUpsert
- ElmSupplierItemsUpsert
- ElmUofMUpsert
- ElmDomains
- ElmShipToUpsert
- ElmSESupplierUpsert
- ElmSECustomerUpsert

Subscriber HTTP Version

The latest version of QAD EQMS requires use of HTTP version 1.1. By default, the subscribers noted above are likely using HTTP version 1.0. Navigate to the configuration tab and select each of the above noted subscribers. Each has a long parameter configuration detail screen. The HTTP Version setting is located here. Change it from 1.0 to 1.1 and then click the Save button at the bottom of the detail screen to apply your change.

Fig. 15: QXtend Outbound screen, Configuration tab



Step 6

"Force Publish" Data from QAD Adaptive ERP to QAD EQMS

Be sure to read through the entirety of this step before starting the changes mentioned within it.

This step is intended for the initial population of data into QAD EQMS and must be performed in a specific order to avoid errors. When you are finished force publishing a given interface, it is important that you check the EQMS Integration log and the QXtend Subscriber Messages log for errors before force publishing the next interface. If too many errors are generated, QXtend will crash, which requires QAD Support to clear out the subscriber messages queue in order to make the tool functional again. As recovery is difficult and requires extra resources, do not rely solely on the single success message from the QAD interface after force publishing; it does not accurately represent the status of each individual message.

The best way to check for errors is to consult the Viewers tab within QXO. Open the subscriber log and filter the screen for one of the following:

- Status = "SENDER"
- Status = "HOLD"

If any records for the subscriber you just published are returned with one of these statuses, it is likely that something did not work and you should investigate before proceeding with the next publish. Status = "PEND" may appear if there are several messages being processed. If it does not eventually become Status = "DLV" then this is also evidence of a problem. Manually refresh the Subscriber Messages screen to see the updated status.

The first step you should take for troubleshooting is to highlight one of the records in the subscriber log. Then scroll to the bottom and click the Response button. This will show you the response/error that was returned.

Most interfaces can be force published by performing the following steps for each interface*:

1. Navigate to the Configuration tab in QXtend Outbound.
2. Click Subscribers.
3. Find the correct interfaces in the list.
4. Click Force Publish.
5. Complete the Register Profiles wizard (shown below).

*Note: EE has a few exceptions, which are listed below.

Fig. 16: QXtend Outbound screen, Configuration tab

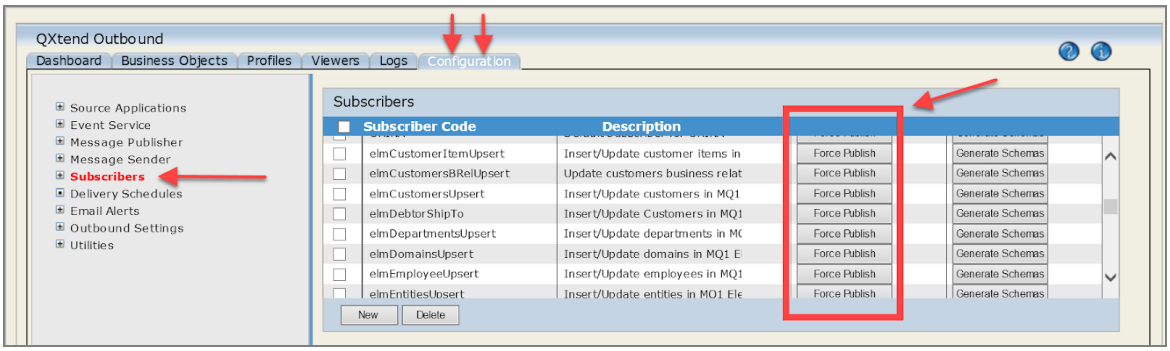
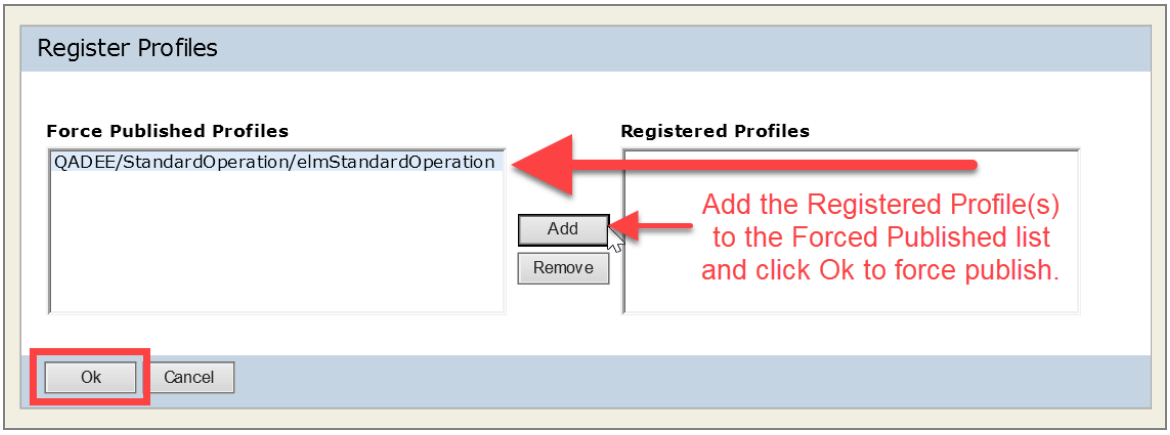


Fig. 17: Register Profiles screen



For EE, the order is as follows, with interfaces requiring a different publishing method using the QAD .Net client noted with an asterisk and described below:

- Shared Sets (BsharedSet)*
- Domain (BDomain)*
- Entity (BCompany)*
- Site
- Department
- Inventory Status Code
- Item Group
- Item Type
- Product Line
- Unit of Measure (UofM)
- Inventory Location

- Work Center
- Process
- Customers (BDebtor)*
- Customer Data (CustomerData)
- Customer Ship-To (BDeborShipTo)*
- Suppliers (BCreditor)*
- Supplier Data (SupplierData)
- Business Relation (BBusinessRelation)*
- Item
- Customer Item
- Supplier Item

For SE, the order is as follows:

- Domain
- Entity
- Site
- Department
- Inventory Status Code
- Item Group
- Item Type
- Product Line
- Unit of Measure (UofM)
- Inventory Location
- Work Center
- Process
- Customer
- Supplier
- Ship To
- Item
- Customer Item
- Supplier Item

For EE only, the following interfaces must be force published from within the QAD .Net Client and cannot be force published from QXO:

- Domain (BDomain)
- Entity (BCompany)
- Customer (BDebtor)
- Customer Ship-To (BDeborShipTo)
- Supplier (BCreditor)
- Business Relation (BBusinessReleation)

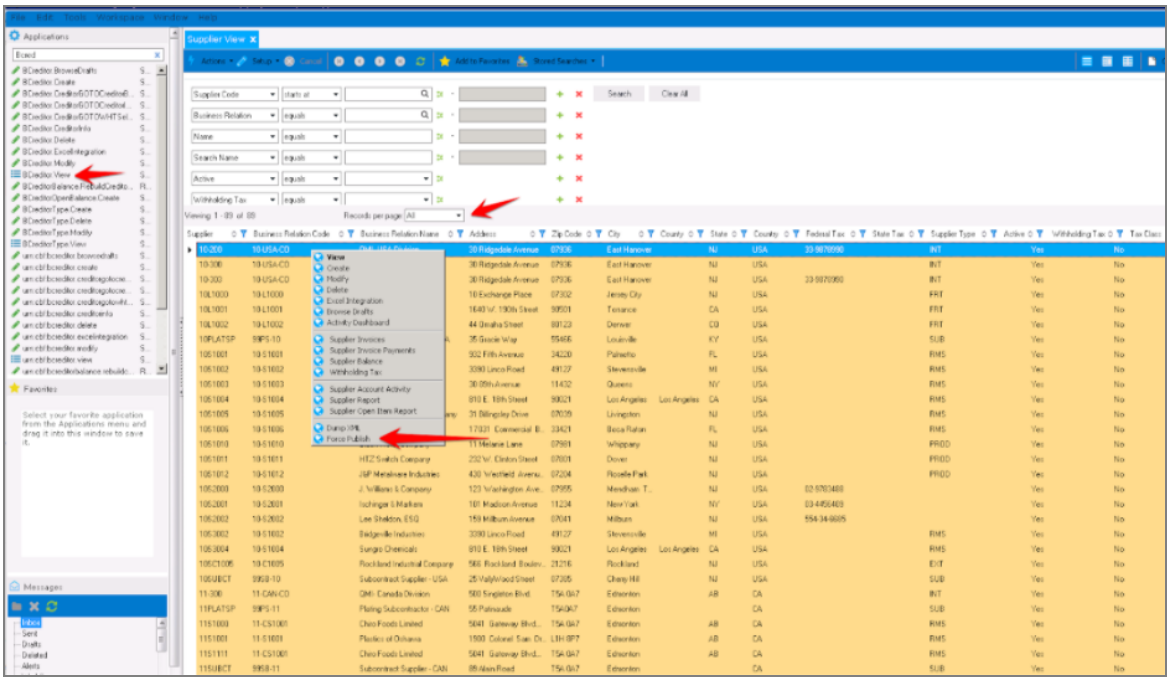
For these interfaces, the following steps must be performed to publish financial data from EE:

- Open the interface's View screen in the QAD .Net UI client.
- Set the "Records per page" value to "All".
- Highlight all records, right-click, then select Force Publish.

Data must be force published from each QAD Workspace separately, as the data in each workspace is different. If this is not done correctly, then you will drive errors when you run the force publishes for non-financial data from the QXtend UI, as this will send data for all workspaces and there are dependencies. For example, a given customer must first be sent to QAD EQMS in order for QAD to successfully receive the related Customer Item record. See *Fig. 18: QAD .Net UI Client, force publishing suppliers* below for an example.

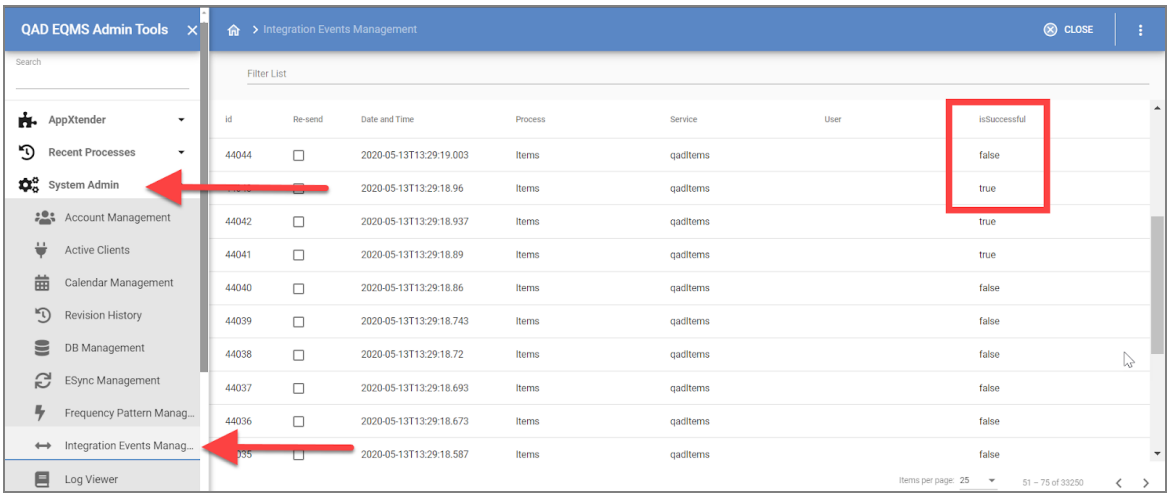
> ***Reminder:*** To access a given workspace, you must select it from the Workspace drop-down menu in the top screen toolbar while logged into the QAD .Net UI Client.

Fig. 18: QAD .Net UI Client, force publishing suppliers



After all force publishes are done executing, integration is configured and the initial data population is complete. It is a best practice to check the QAD EQMS side integration log to ensure there are no illegitimate failures. To do this, navigate to the Integration Events Management process search screen in the System Admin application (see *Fig. 19: QAD EQMS, Integration Events Management process search screen* below). Filter results for "False". Only records whose "issuccessful" column reads "False" should be returned.

Fig. 19: QAD EQMS, Integration Events Management process search screen



Chapter 3

Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration (Level 2)

Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration (Level 2)

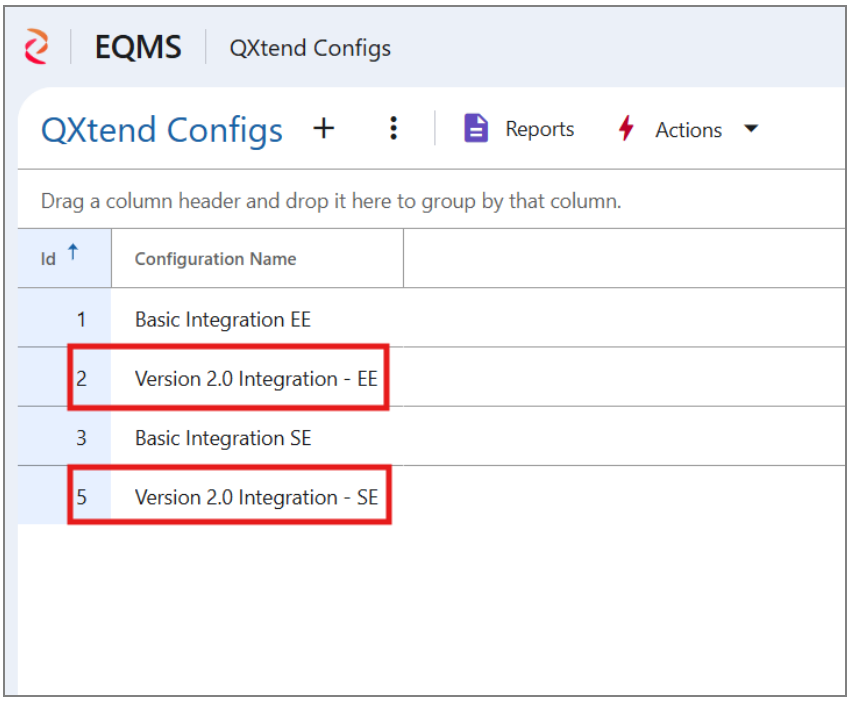
Step 1

Set Up QAD EQMS for Readiness to Configure Integration in QXtend

This step is intended to prepare the QAD EQMS for configuration integration in QXtend. First, the Source App Type and Source App Values in QAD EQMS must be updated to match the content in QXtend. Follow the instructions below to accomplish this:

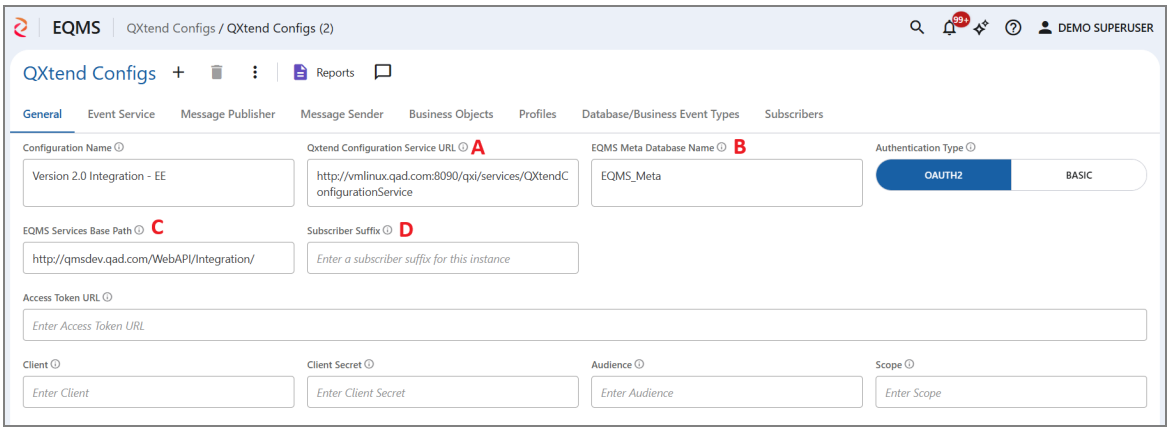
1. Open the QXtend Configs process search screen in QAD EQMS. Four records appear: two labeled for EE, two labeled for SE.
2. Choose between "Version 2.0 Integration EE" or "Version 2.0 Integration SE", depending on which you are configuring.

Fig. 20: QAD EQMS, QXtend Configs search screen



3. Open your selected record. There are six fields to verify on the General tab of this screen.

Fig. 21: QXtend Configs screen, General tab



A. Ensure the "QXtend Configuration Service URL" is set to the current QXtend integration URL.
B. Ensure the meta database name matches the actual one used by the QAD EQMS site. The meta database name can be discovered by logging into the QAD EQMS site's AdminTools. Select the three-dot menu in the upper-right corner and choose About.
C. Ensure the EQMS Service Base Path field is set to the current QAD EQMS sites integration URL. This consists of the base site URL and the WebAPI name from IIS, followed by "/Integration/". Here is an example URL: *https://qmsdev.qad.com/webapi/Integration/*

   You MUST include the slash at the end of Integration or it will not work. You can test this URL by appending "qadItems.asmx?wsdl" to the end of the URL (for example: *https://qmsdev.qad.com/webapi/Integration/qadItems.asmx?wsdl*). This should take you to an XML document (aka the wsdl). If it does, then the URL is correct and you can move on.

D. Include the subscriber suffix if used.
E. Select the authentication type to be used. Use OAuth 2.0 Authentication (for Qxtend 2.1+). This toggle allows you to select the authentication method used to connect to your Qxtend instance. The correct method depends on the version of Qxtend you are using.
   i. **Authentication Type: OAUTH2**

      Use when: You are connecting to a Qxtend version 2.1 or newer. The application will use the OAuth 2.0 flow to authenticate. You will be required to provide the Access Token URL, Client ID, Client Secret, and other related OQuth 2.0 fields.

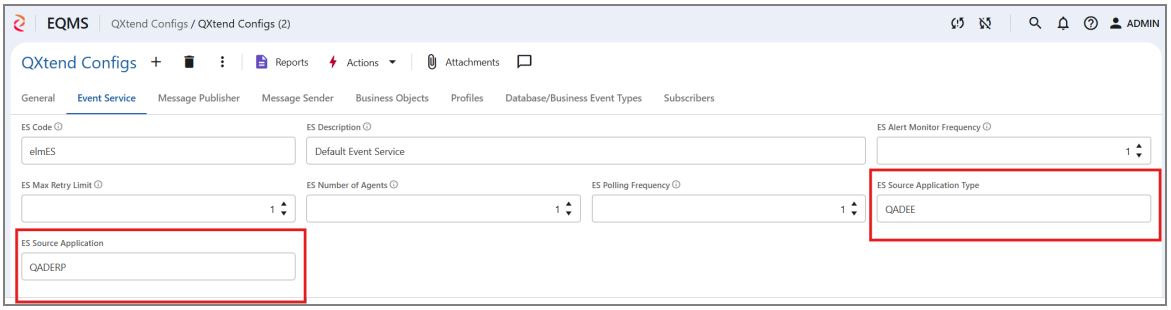
   ii. **Authentication Type: Basic**

      Use when: You are connecting to a Qxtend version older than 2.0. The application will use Basic Authentication. You will be required to provide a standard Username and Password for your Qxtend account.

F. Include the subscriber suffix if used.

4. Navigate to the Event Service tab. Ensure the ES Source Application Type and ES Source Application match the values from Step 1 on page 12. These should be the Source Application Type and Source Application name from QXtend Outbound configuration tab, respectively.

Fig. 22: QAD EQMS, QXtend Configs process screen, Event Service tab



5. Update the QAD EQMS references pointing to the QXtend Inbound URL to the correct URL. This can be done manually, or it can be done automatically by running the command "0 - Update Source Application Type" from the QXtend Configs detail screen in which you have been making changes.

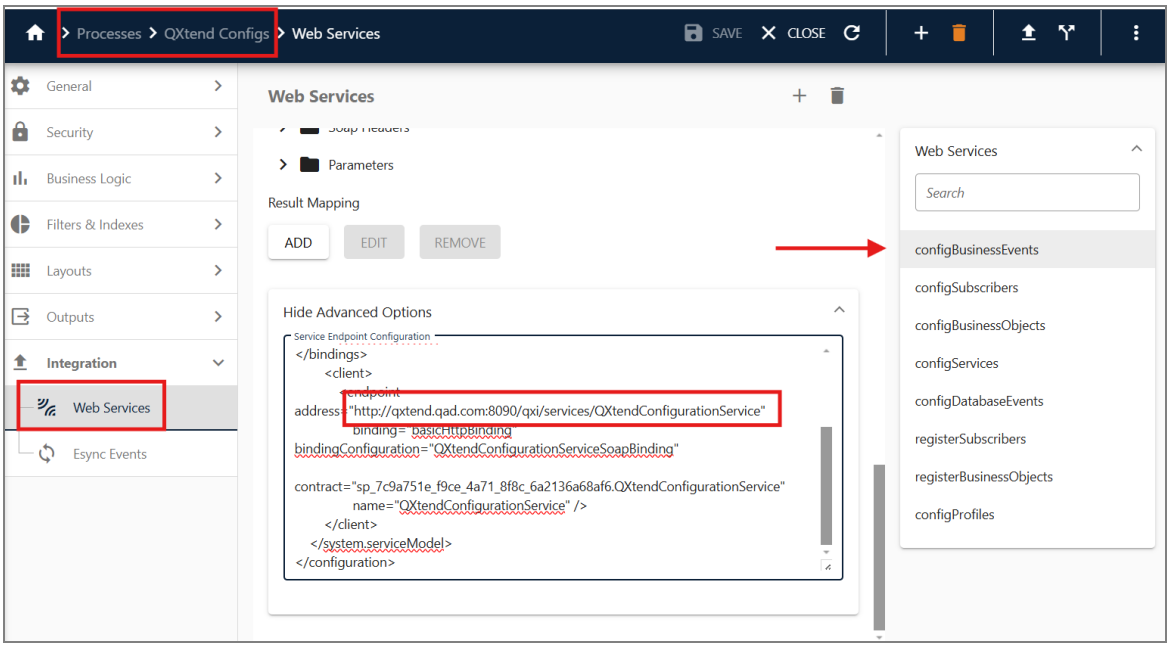
   Under the Actions menu, you will see a list of several commands, but only command zero should be run. After running this command and before running any further commands, you must still publish and deploy by following the manual setup directions below.

Manual Process for Updating QXtend Inbound URLs in QAD EQMS

1. Open the QXtend Configs process in QAD EQMS AppXtender.
2. Navigate to the Integration section and choose Web Services. You will see several pre-configured services.
3. Extend the process and click the paperclip icon for each service to extend them and make them editable.
4. For each service, click Show Advanced Settings, which reveals a text box labeled Service Endpoint Configuration. In this text box only, reset the existing placeholder QXtend inbound URL to the correct QXtend inbound URL, provided by the customer or QAD. Only change up to "qxi" ("http://vmlinux.qad.com:8090/qxi/"). Do not change any other URLs in the integration section or take any other action.

   Do this for each integration service in the QXtend Configs process, as well as in each of the following processes: QXtend Business Object Configs, QXtend Profile Configs, and QXtend Database Event Configs. An example is below.

Fig. 23: AppXtender, QXtend Configs process screen, Web Services tab



Whether you updated the URLs manually or automatically using command zero, you must publish the following processes in AppXtender in the following order:

1. QXtend Configs (soft publish)
2. QXtend Business Object Configs (soft publish)
3. QXtend Profile Configs (soft publish)
4. QXtend Database Event Configs (publish and deploy)

At this point, both QXtend and QAD EQMS are ready to auto-configure the integration interfaces.

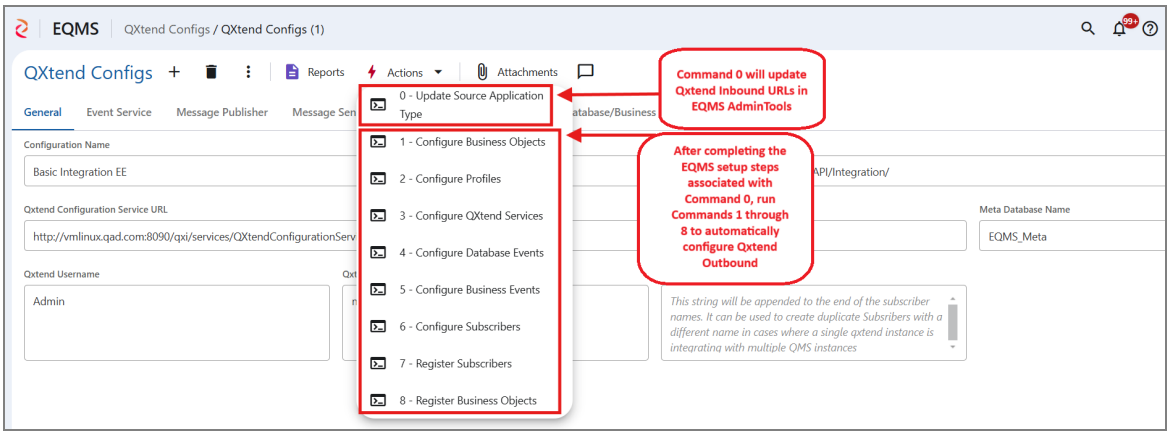
Step 2

Configure QXtend Outbound from QAD EQMS

Return to the Version 2.0 EE/SE record from the QXtend Configs process search screen in QAD EQMS (see "Step 3 " on page 16). Double-click the process to enter the record's detail screen.

On this record, you will see a numbered list of commands. Run each remaining command from top to bottom (lowest to highest). This action configures QXtend Outbound automatically. Once it is finished, integration should be configured and functional. If any errors appear when running these commands, then you must troubleshoot them before you can move on.

Fig. 24: QAD EQMS, QXtend Configs process screen, Commands list



The best way to confirm success is to check the Configuration tab in QXtend Outbound URL. Look under Subscribers and ensure a subscriber exists for each of the following:

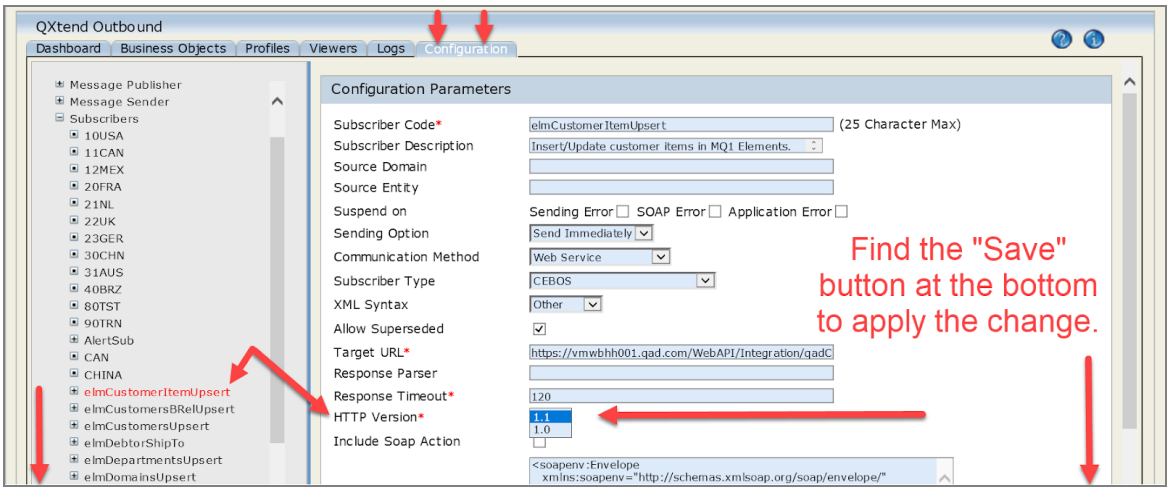
For EE or SE:

- elmEmployeeUpsert
- elmInvTrans

Subscriber HTTP Version

The latest version of QAD EQMS requires use of HTTP version 1.1. By default, the subscribers noted above are likely using HTTP version 1.0. Navigate to the configuration tab and select each of the above noted subscribers. Each has a long parameter configuration detail screen. The HTTP Version setting is located here. Change it from 1.0 to 1.1 and then click the Save button at the bottom of the detail screen to apply your change.

Fig. 25: QXtend Outbound screen, Configuration tab



Step 3

"Force Publish" Data from QAD Adaptive ERP to QAD EQMS

Be sure to read through the entirety of this step before starting the changes mentioned within it.

This step is intended for the initial population of data into QAD EQMS and must be performed in a specific order to avoid errors. When you are finished force publishing a given interface, it is important that you check the EQMS Integration log and the QXtend Subscriber Messages log for errors before force publishing the next interface. If too many errors are generated, QXtend will crash, which requires QAD Support to clear out the subscriber messages queue in order to make the tool functional again. As recovery is difficult and requires extra resources, do not rely solely on the single success message from the QAD interface after force publishing; it does not accurately represent the status of each individual message.

The best way to check for errors is to consult the Viewers tab within QXO. Open the subscriber log and filter the screen for one of the following:

- Status = "SENDER"
- Status = "HOLD"

If any records for the subscriber you just published are returned with one of these statuses, it is likely that something did not work and you should investigate before proceeding with the next publish. Status = "PEND" may appear if there are several messages being processed. If it does not eventually become Status = "DLV" then this is also evidence of a problem. Manually refresh the Subscriber Messages screen to see the updated status.

The first step you should take for troubleshooting is to highlight one of the records in the subscriber log. Then scroll to the bottom and click the Response button. This will show you the response/error that was returned.

Most interfaces can be force published by performing the following steps for each interface*:

1. Navigate to the Configuration tab in QXtend Outbound.
2. Click Subscribers.
3. Find the correct interfaces in the list.
4. Click Force Publish.
5. Complete the Register Profiles wizard (shown below).

*Note: EE has a few exceptions, which are listed below.

Fig. 26: QXtend Outbound screen, Configuration tab

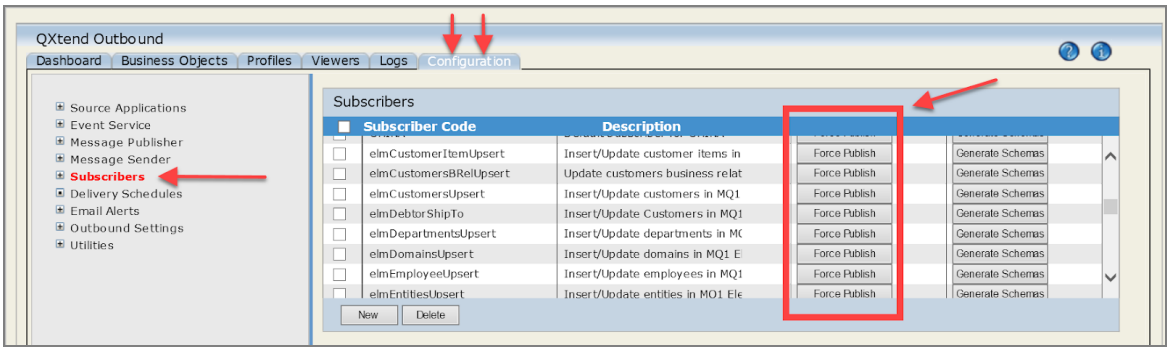
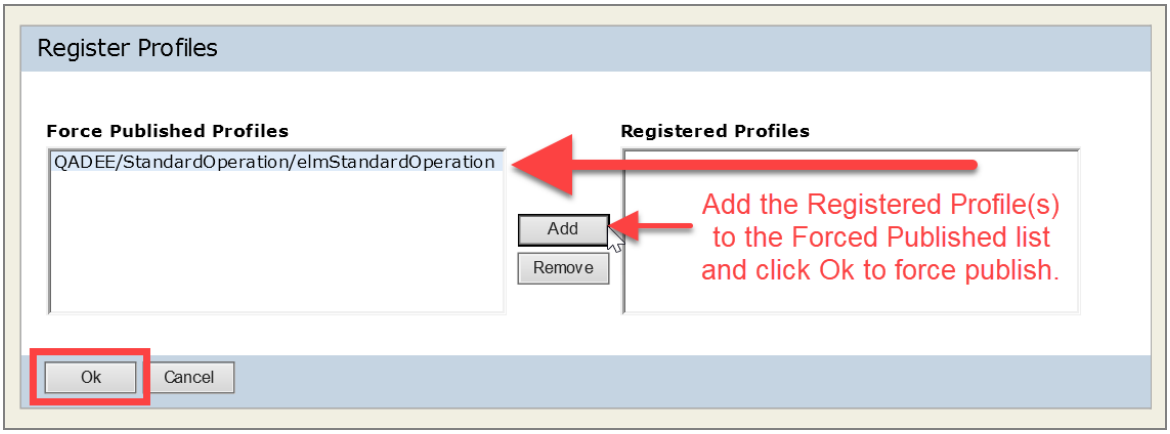


Fig. 27: Register Profiles screen



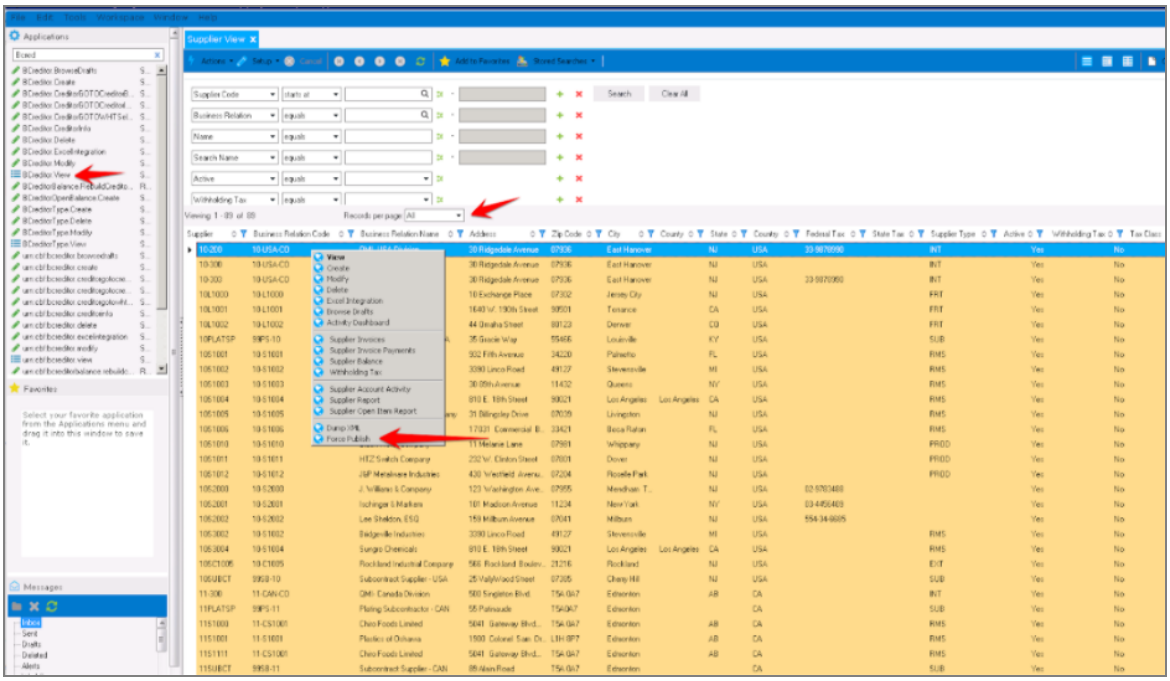
For EE or SE, the order is as follows:

- Employees (BEmployee)
- Inventory Transactions (InventoryTransaction)

Data must be force published from each QAD Workspace separately, as the data in each workspace is different. If this is not done correctly, then you will drive errors when you run the force publishes for non-financial data from the QXtend UI, as this will send data for all workspaces and there are dependencies. For example, a given customer must first be sent to QAD EQMS in order for QAD to successfully receive the related Customer Item record. See *Fig. 28: QAD .Net UI Client, force publishing suppliers* below for an example.

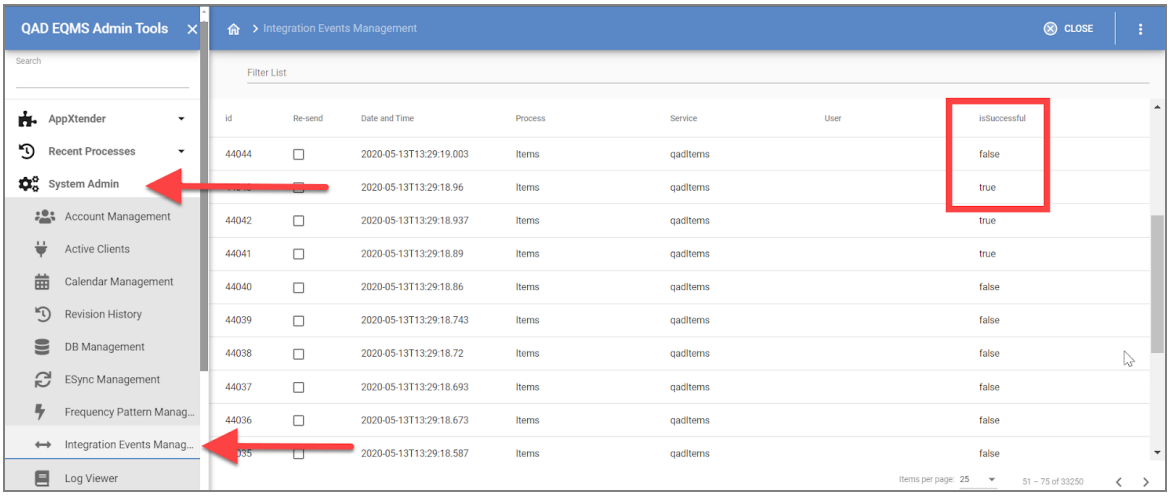
> ***Reminder:*** To access a given workspace, you must select it from the Workspace drop-down menu in the top screen toolbar while logged into the QAD .Net UI Client.

Fig. 28: QAD .Net UI Client, force publishing suppliers



After all force publishes are done executing, integration is configured and the initial data population is complete. It is a best practice to check the QAD EQMS side integration log to ensure there are no illegitimate failures. To do this, navigate to the Integration Events Management process search screen in the System Admin application (see *Fig. 29: QAD EQMS, Integration Events Management process search screen* below). Filter results for "False". Only records whose "issuccessful" column reads "False" should be returned.

Fig. 29: QAD EQMS, Integration Events Management process search screen



Chapter 4

Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration (Level 3)

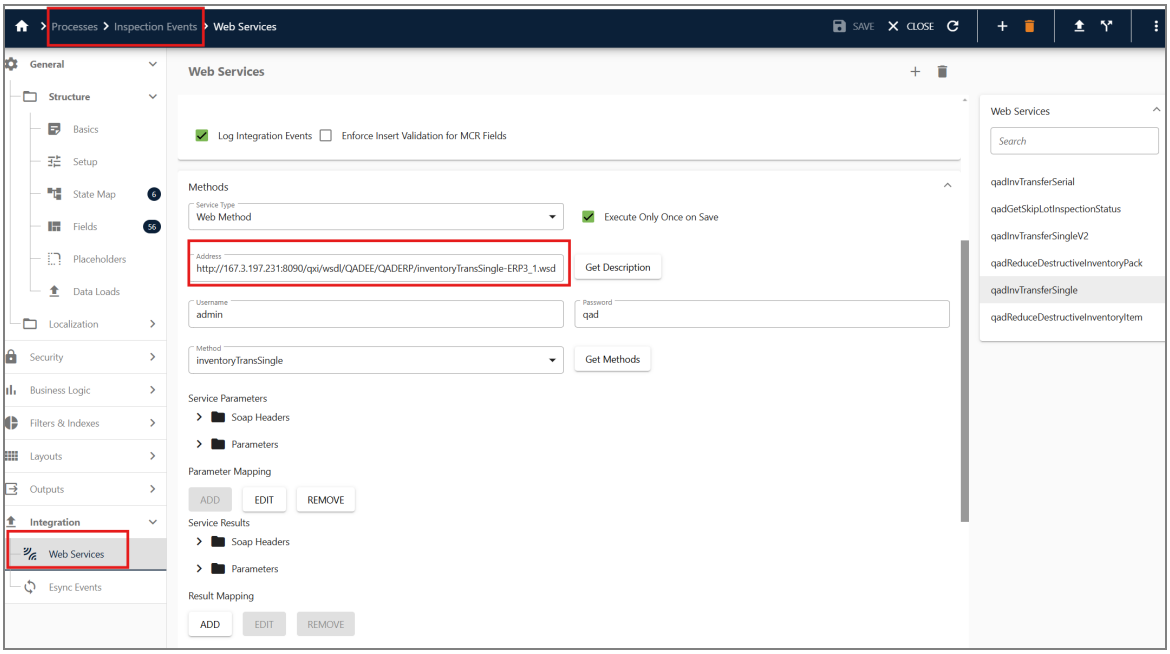
Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration (Level 3)

1. Open process "Inspection Events" in EQMS Admin tools
2. Extend the process.
3. Review all web services and their documentation to determine which service is required. Each service has a specific function.
4. Update the address field for the required service to include the Qxtend URL. Example: [https://\|https://]{qxtendURL}/wsdl/QADEE/QADERP/transferInvSingleItem-ERP3_1.wsdl?soapheaders=true. (Note http vs https)

   Note: The example above is for the "Single Item Transfer" service. The {qxtendURL} portion is what will need to be updated.

5. Save, publish and deploy.

Fig. 30: Admin Tools, Inspection Events process, Web Services



Chapter 5

Additional Information

Creating Custom Integrations

- SOAP Integration Configuration Guide 2025: *https://documentlibrary.qad.com/documents/2370665/2520006/SOAPIntConfig_TR_v2025.pdf*
- REST Integration Configuration Guide 2025: *https://documentlibrary.qad.com/documents/2370665/2520006/OutboundRESTIntConfig_TR_v2025.pdf*
- Training video covering QXtend QXO components: *https://drive.google.com/open?id=1dW_PHxyRV6Q9dX1rU-VEJCePzaoHlYyZ*

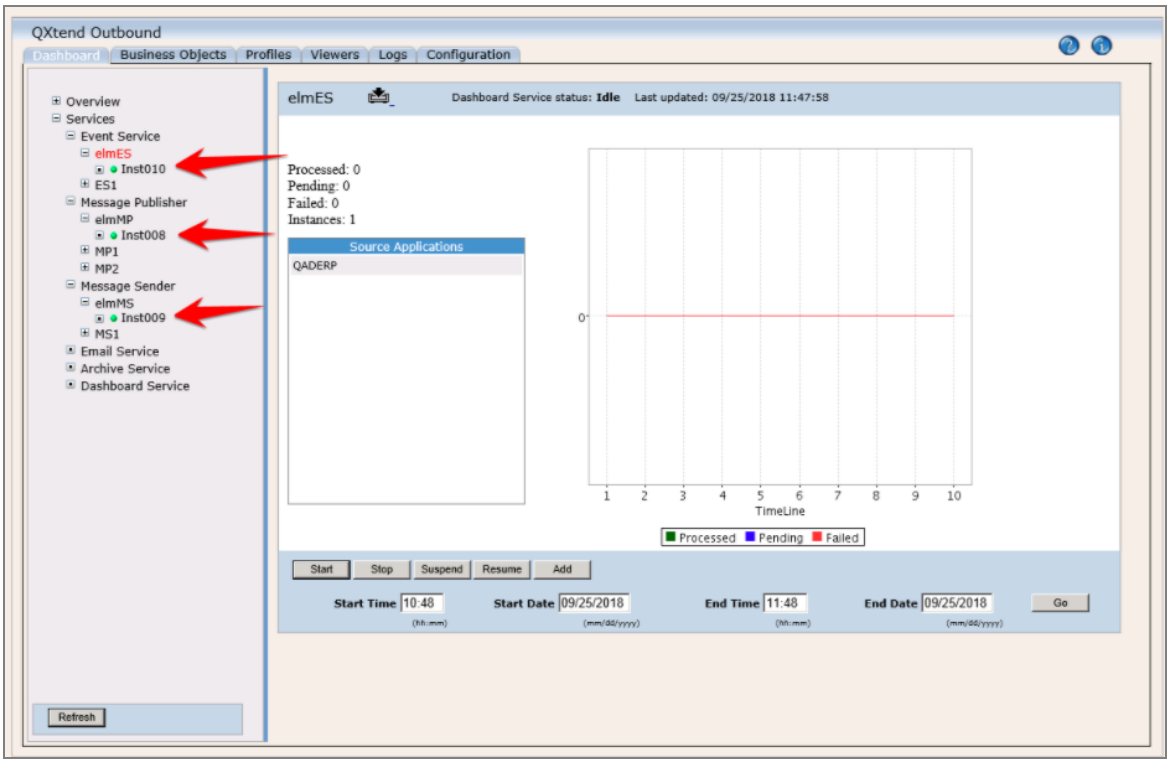
Troubleshooting Integrations / Common Problems

Force Publishing Problems

If you force publish and do not receive a message in the QAD EQMS Integration Log, it is likely that the problem is in QXtend. To resolve the issue, perform the following steps in QXO in order:

1. Navigate to the Dashboard tab. There are three types of services used in QXO that must all be running for integration to work: "ElmMS", "ElmMP", and "ElmES". If these services are running, a green dot is shown below the service (see *Fig. 31: QXtend Outbound, Dashboard tab, running services* below). If the services are not running, then select the service name and click Start.

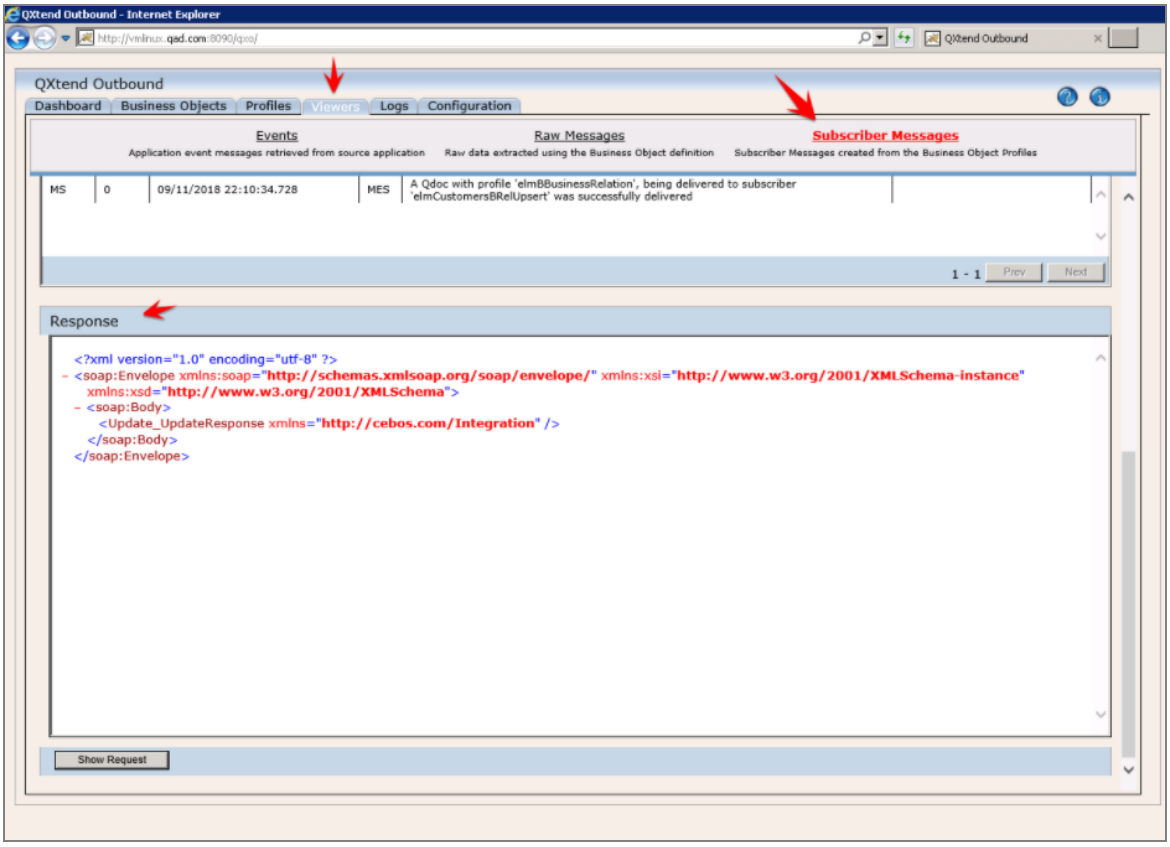
Fig. 31: QXtend Outbound, Dashboard tab, running services



2. If you are still unable to force publish, navigate to the Viewers tab, which shows most of the QXtend side integration logs. The most important item is the Subscriber log.

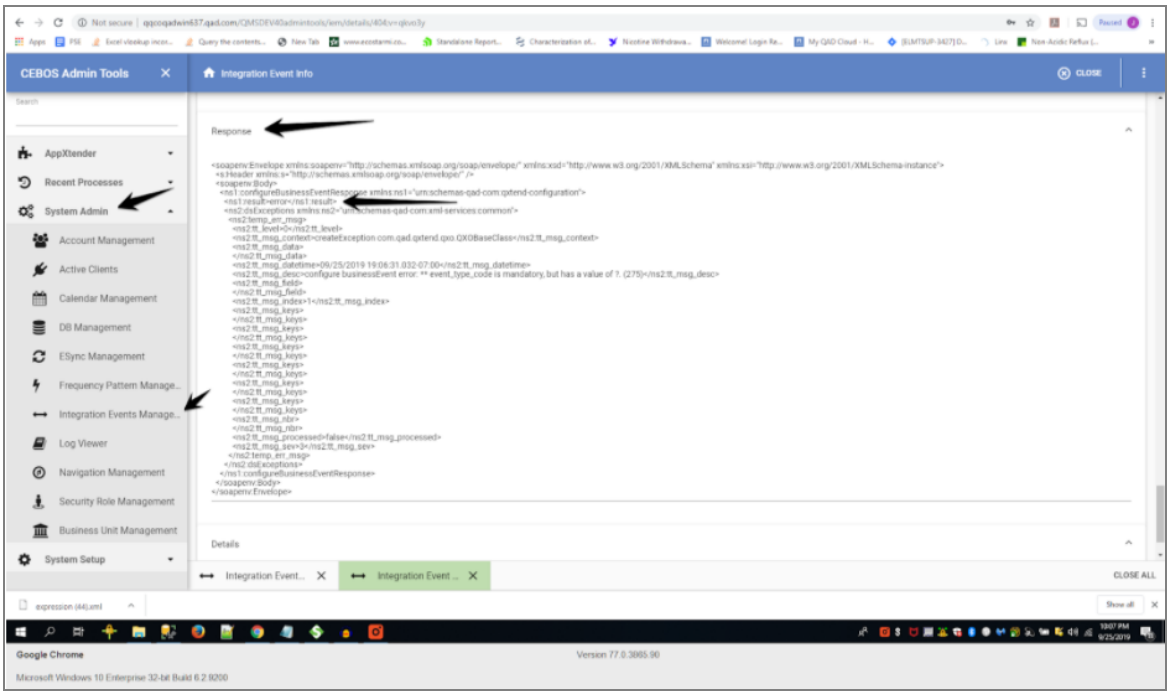
   If no messages appear in the Subscriber log, then there may be a QXtend configuration issue. If messages are appearing in the log, then you can view the error messages by highlighting the message and clicking Response, as shown in *Fig. 32: QXtend Outbound, Viewers tab, Subscriber Messages* below.

Fig. 32: QXtend Outbound, Viewers tab, Subscriber Messages



3. You may find that QXtend is not being configured even though the QAD EQMS side commands have been run. In this case, it is likely that one or more outbound messages sent from the commands to configure QXtend from QAD EQMS failed. Troubleshoot this by checking the QAD EQMS Integration Log for errors. This can be slightly deceiving because the failed logs may show the "IsSuccessful" column with a True value. Note that "IsSuccessful" only indicates whether the message arrived to its target; it does not indicate that the target accepted the message successfully.
4. If QXtend was not configured in some way, drill into the outbound messages from the QAD EQMS Integration Log and check the responses for errors (even if "IsSuccessful" is true). The screenshot below displays a positive response. If the message had failed, you would see more information, including a full error message.

Fig. 33: QAD EQMS, Integration Events Manager, Integration Even Info



If QXtend appears to be configured correctly, the services are running, but there are no new messages appearing in the subscriber log, or the subscriber log is adding new messages but they are stuck in Pending status, the issue may be that the QXtender subscriber log is overloaded with too many pending messages. The only option to make messages flow again may be to create a ticket requesting that QAD Support clear out the subscriber log of all pending messages.

Appendix

- Complex Integration 2.0/3.0 Workflows:
  - (PE Integration) Employee Skill Check Inbound and Manual Documents Push Services: *https://docs.google.com/document/d/1vERY1IZwfHX9EVsLu1tSw7ZDU8bRRF49mdtLTPrtpdE/edit?usp=sharing*
  - OneLogin, QAD EQMS and Supplier Portal Supplier Account Integration Related Services: *https://docs.google.com/document/d/17BoKP7sSyocGSiLjU23DbOljNZP7xEhq8xm55C-jS6Y/edit?usp=sharing*
  - AS Skip Lot Integration: *https://docs.google.com/document/d/1KFSzB2_EECjd3baSHsknLsh46uxbP7YvVFtp-r2fUm4/edit?usp=sharing*
- How to set up a custom QXtend Subscriber and how its settings correspond to QAD EQMS Integration configuration: *https://drive.google.com/open?id=1SfXy_RFKAgG6mmnBYa1i5CPMyHqZohGu*
- Integration Specifications 2025: *https://drive.google.com/file/d/1E4wGdHDb_muIPOUrAyA24khpgc0BocM8/view?usp=drive_link*
- Using Postman to get REST token from QAD EQMS: *https://team.qad.com/pages/viewpage.action?pageId=148933790*

- Guide to setting up an example REST services in QAD EQMS: *https://team.qad.com/display/EQMS/Outbound+REST+Integration+Services*
- SOAP Integration Configuration Guide 2025: *https://documentlibrary.qad.com/documents/2370665/2520006/SOAPIntConfig_TR_v2025.pdf*
- REST Integration Configuration Guide 2025: *https://documentlibrary.qad.com/documents/2370665/2520006/OutboundRESTIntConfig_TR_v2025.pdf*
- Training video covering QXtend QXO components: *https://drive.google.com/open?id=1dW_PHxyRV6Q9dX1rU-VEJCePzaoHlYyZ*
- List of QXtend QXO components, what they are for, and who is responsible for them: *https://drive.google.com/open?id=16CfdHX0uRcwBX9j1ZXZQsQbsROIIuOoAWHrUbbDGltY*