



QAD Adaptive Applications

Technical Reference **QAD EQMS Integration**

70-3474-2020.1
QAD EQMS 2020.1
September 2020

Copyright

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

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

This document contains trademarks owned by QAD Inc. and other companies.

Copyright © 2020 by QAD Inc.

QAD Inc.

27555 Executive Drive

Suite 225

Farmington Hills, MI 48331

Phone: + 1 (810) 534-2222

<http://www.qad.com>

QAD EQMS	1
Copyright	2
QAD EQMS Integration 2020.1 Technical Reference	5
Change Summary	5
Chapter 1: Introduction to QAD EQMS Integration	6
Introduction	7
About This Guide	7
Prerequisites for QAD EQMS Integration	7
User Roles	7
Access and Prerequisite Information Needed	7
Integration Workflows	7
Compatibility Constraints	8
Capabilities of QAD EQMS Integration	8
Web Services	8
Call Service	8
Create Service	9
RESTful APIs	9
Esync	9
Encryption	10
Limitations of QAD EQMS Integration	10
Chapter 2: Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration	11
Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration	12
Step 1	12
Step 2	13
Step 3 (EE Only)	16
Step 4	16
Step 5	20
Step 6	21

Creating Custom Integrations	25
Troubleshooting Integrations / Common Problems	25
Force Publishing Problems	25
Chapter 3: Appendix	29
Appendix	30

Change Summary

[illegible]

Chapter 1

Introduction to QAD EQMS Integration

Introduction...7

About This Guide...7

Prerequisites for QAD EQMS Integration...7

Capabilities of QAD EQMS Integration...8

Limitations of QAD EQMS Integration...10

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 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
- Admin login credentials (to access the QXtend related screens, event screens, and all regular data screens)
- 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 30 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 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 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

Step 1...12

Step 2...13

Step 3 (EE Only)...16

Step 4...16

Step 5...20

Step 6...21

Creating Custom Integrations...25

Troubleshooting Integrations / Common Problems...25

Setting Up Out-of-the-Box QAD EQMS <=> QAD Adaptive ERP Integration

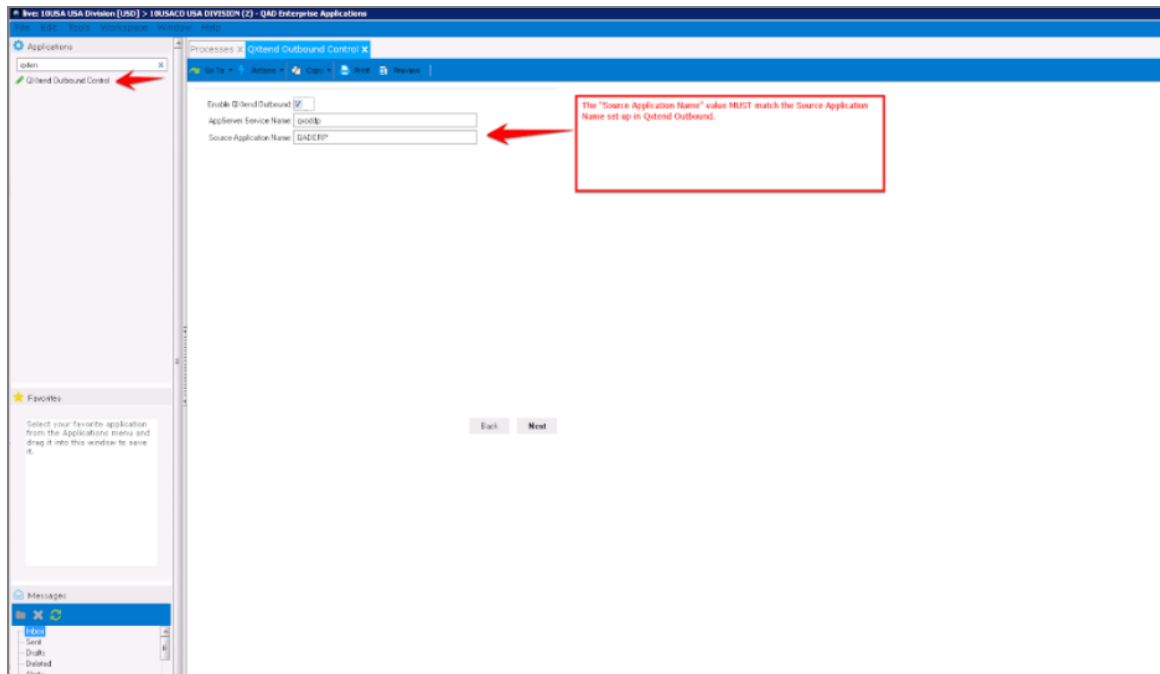
Step 1

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

To begin, ensure the QXtend Outbound source app name in QAD Net UI client matches the information in QXtend Outbound. The source app name can be anything, as long as it matches in these two locations.

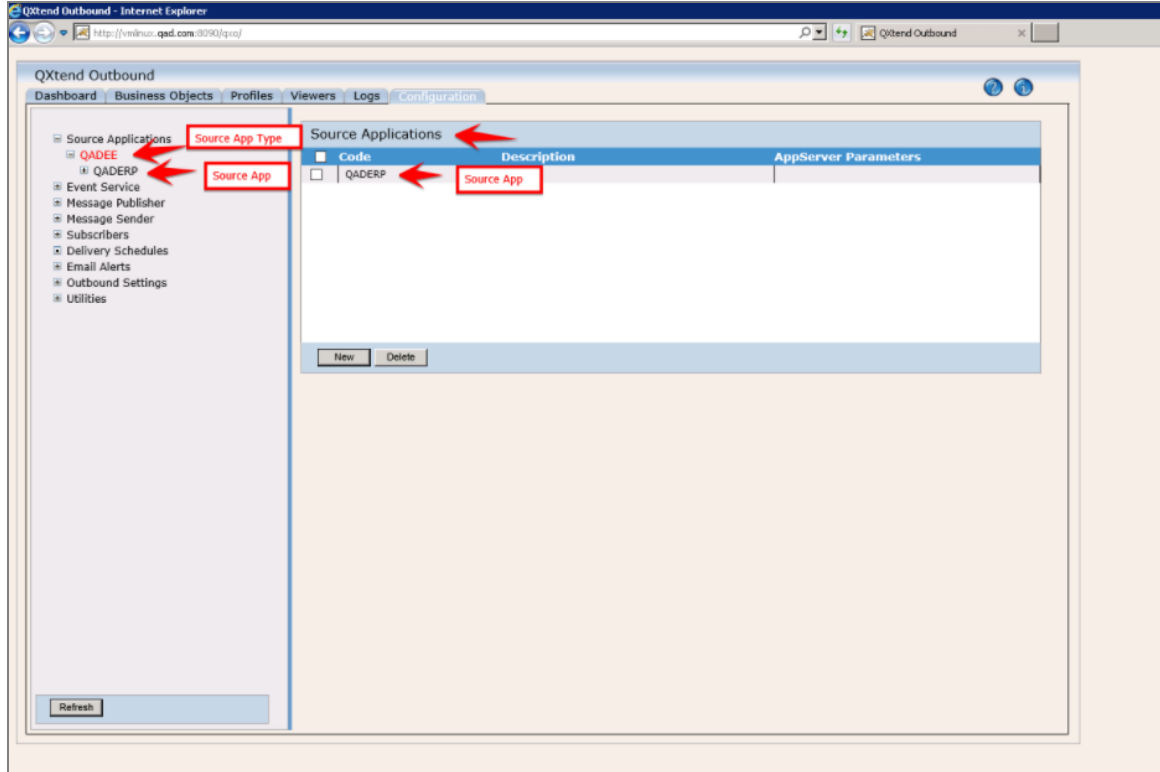
See *Fig. 1: QAD Net UI Client, Source App Name location* below to find the source app name in the QAD Net UI client.

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



See *Fig. 1: QAD Net UI Client, Source App Name location* below. In the Source Applications field within the Configuration tab, the value in the Code column is the source app name that must match.

Fig. 2: QXtend outbound, Source App Name location



Step 2

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

For EE only, the Business Events must be configured for financial data in order to work; you can check for this by navigating to the "BEEventConfig. View" screen in QAD .Net UI. These BEEvent records must be added separately in each QAD 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, you will log into that specific workspace.

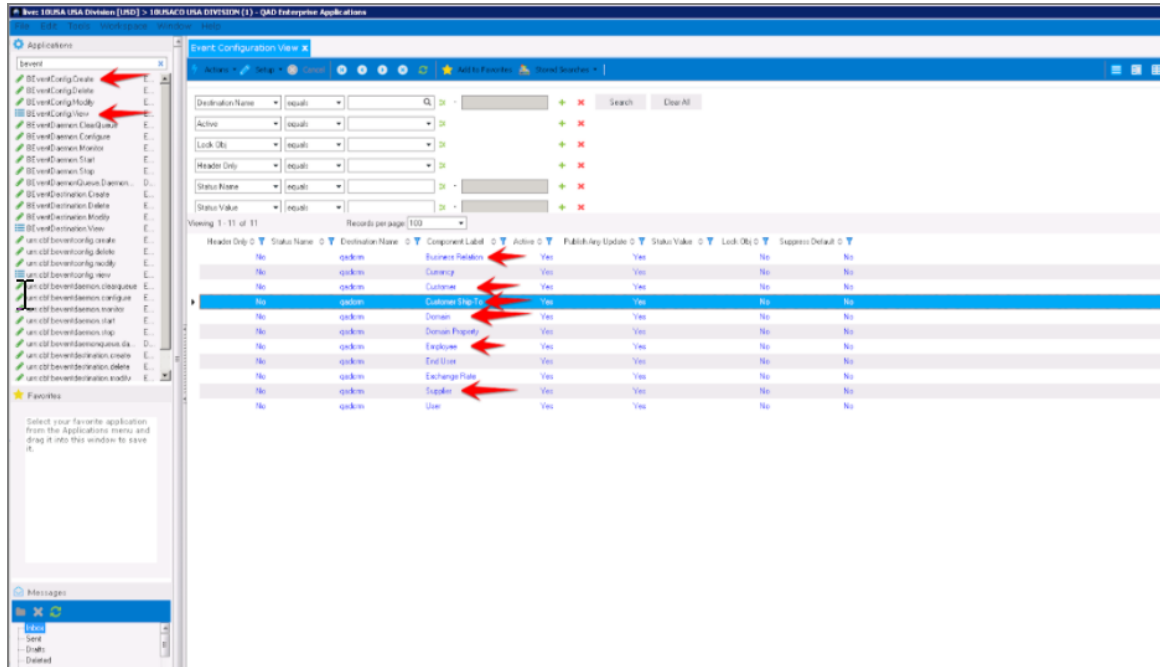
Several interfaces must be performed separately in each workspace from which QXtend should send data. This includes the following interfaces:

- Domains (bdomains)
- Customers (bdebtors)
- Customer Ship-To (bdebtorShip-To)
- Suppliers (bcreditors)
- Business Relations (bbusinessrelation)
- Entities (bCompany)
 - Entities is the latest standard interface

Most of these interfaces should be set up when QAD installs QXtend, but this does not include Entities, 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. 3: QAD .Net UI Client, Event Configuration View tab* below, which shows which records should be configured.

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



After clicking the Create button, you can find the component for Entities by searching for the component code "BCompany".

Fig. 4: QAD .Net UI Client, Event Configuration Modify tab

Header Only	Status Name	Destination Name	Suppress Default	Suppress Empty	Component Label	Active	Publish Any Update	Status Value	Lock
No	qadom		No		No Business Relation	Yes	Yes		
No	qadom		No		No Currency	Yes	Yes		
No	qadom		No		No Customer	Yes	Yes		
No	qadom		No		No Customer Ship-To	Yes	Yes		
No	qadom		No		No Domain	Yes	Yes		
No	qadom		No		No Domain Property	Yes	Yes		
No	qadom		No		No End User	Yes	Yes		
No	qadom		No		No Entity	Yes	Yes		
No	qadom		No		No Exchange Rate	Yes	Yes		
No	qadom		No		No Supplier	Yes	Yes		
No	qadom		No		No User	Yes	Yes		

If you are missing events in the "BEventConfig. View" screen, you can add them by navigating to the "BEventConfig. Create" screen and entering the information provided in Fig. 5: QAD .Net UI Client, Event Configuration Create tab below (once per event).

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

Component Label: *Use the look up and select the needed component. This would be things like "Customer", "Supplier", "Domain" or "Business Relation".*

Publishing Update: ☒ *Make sure this is checked*

Destination Name: *Use the look up and select the Destination Name. Typically there is only 1 and this should be set up by default. If there is no destination to choose, you will need someone from QAD to set that up.*

Active: ☒ *Always set to true*

Lock Obj: ☐

Header Only: ☐

Suppress Empty Fields: ☐

Suppress Fields with Default Value: ☐

Step 3 (EE Only)

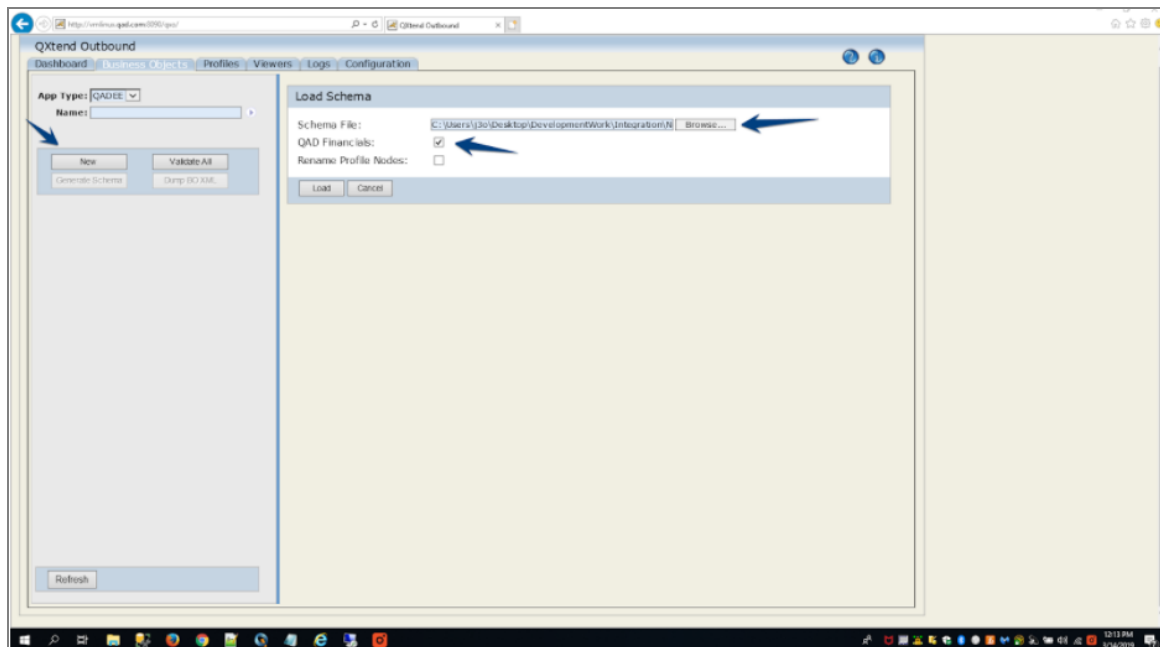
Manually import the BCompany (Entities) business object into QXtend Outbound. Download the new BCompany (Entities) schema file, located [here](#).

Note: The import should be performed prior to running any of the commands below.

In the QXtend outbound URL (QXO), navigate to the Business Objects tab and click New. QXO will open a prompt to ask if you want this to be a Direct Data Publish BO. Click Yes.

Next, in the Schema File field, enter or browse to the path in which you downloaded the BCompany schema. Select the check box for QAD Financials, then click Load to finish creating the custom direct data publish business object for Entities.

Fig. 6: QXtend Outbound, Business Objects tab



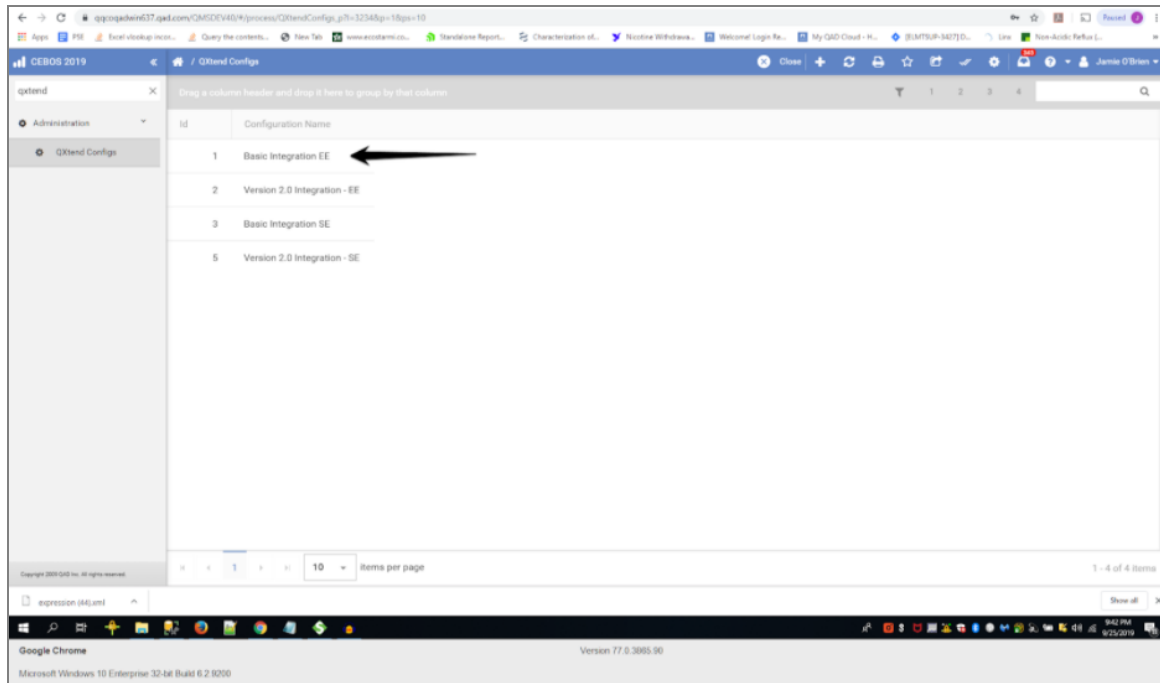
From here, the QXtend and QAD .Net UI Client should be ready to receive the SOAP messages from QAD EQMS, which will automatically configure the integration in QXO.

Step 4

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:

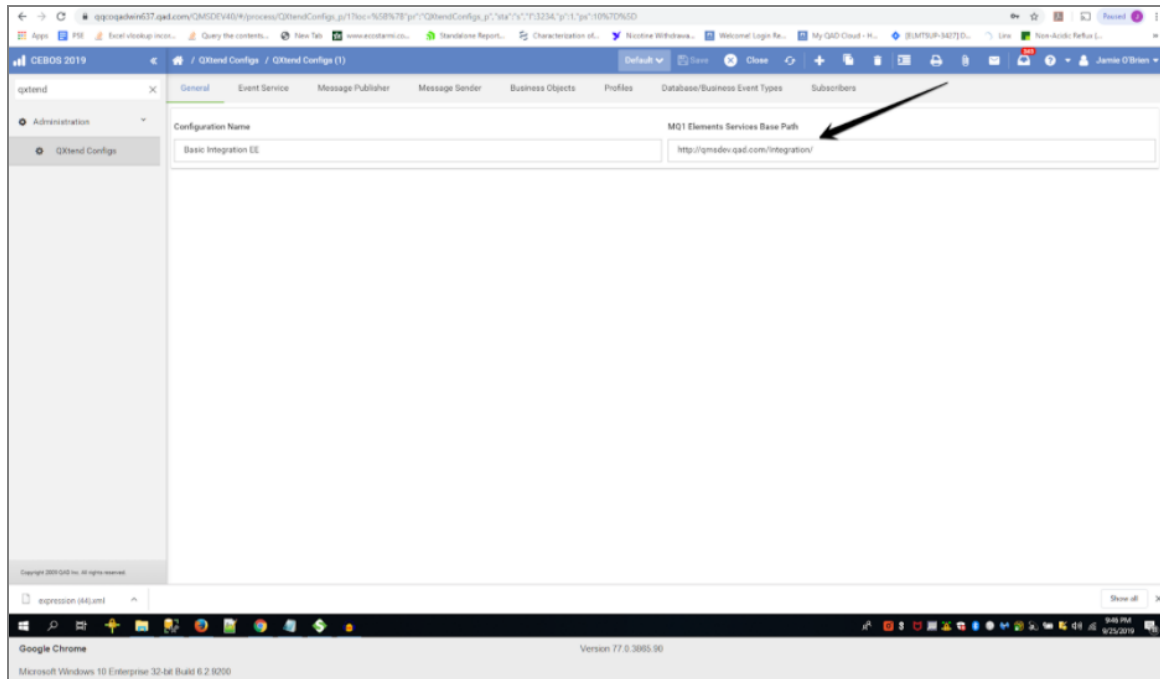
1. Open the QXtend Configs process search screen in QAD EQMS. Four records appear: two labeled for EE, two labeled for SE.
2. Choose between "Basic Integration EE" or "Basic Integration SE", depending on which you are configuring.

Fig. 7: QAD EQMS, QXtend Configs search screen



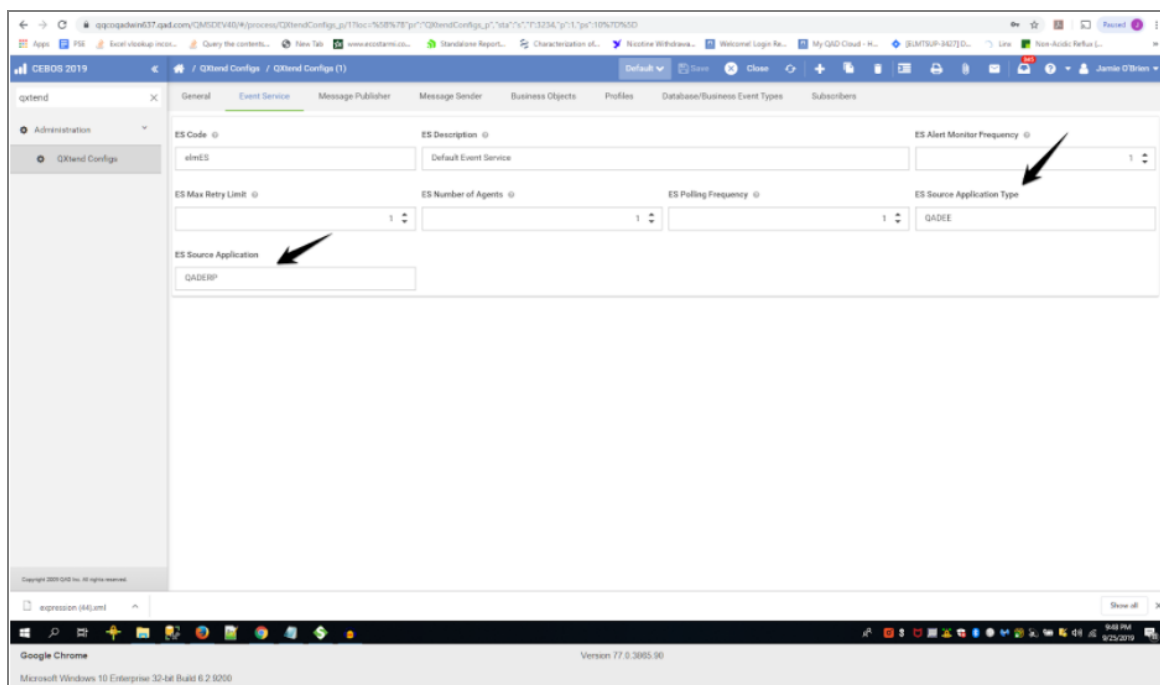
3. Open your selected record. In the General tab, ensure that the MQ1 Elements Services Base Path field is set to the current QAD EQMS sites integration URL.
 - a. 2019+: The URL consists of the base site URL, the webapi name from IIS, and "/Integration/". Here is an example of a 2019 integration URL:
`https://qmsdev.qad.com/webapi/Integration/`
 - b. 2018: The URL consists of the base site URL followed directly by "/Integration/". Here is an example of a 2018 integration URL:
`https://qmsdev.qad.com/Integration/`
 - c. You MUST include the slash at the end of integration or this will not work. You can test this URL by adding "qadItems.asmx?wsdl" to the end of the URL (for example: `https://qmsdev.qad.com/webapi/Integration/qadItems.asmx?wsdl`). This URL should take you to an XML document, due to the wsdl. If it does, then the URL is correct and you may proceed.

Fig. 8: QAD EQMS, QXtend Configs process screen, General tab



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



5. Set the QAD EQMS side references to the QXtend Inbound URL to the correct URL.

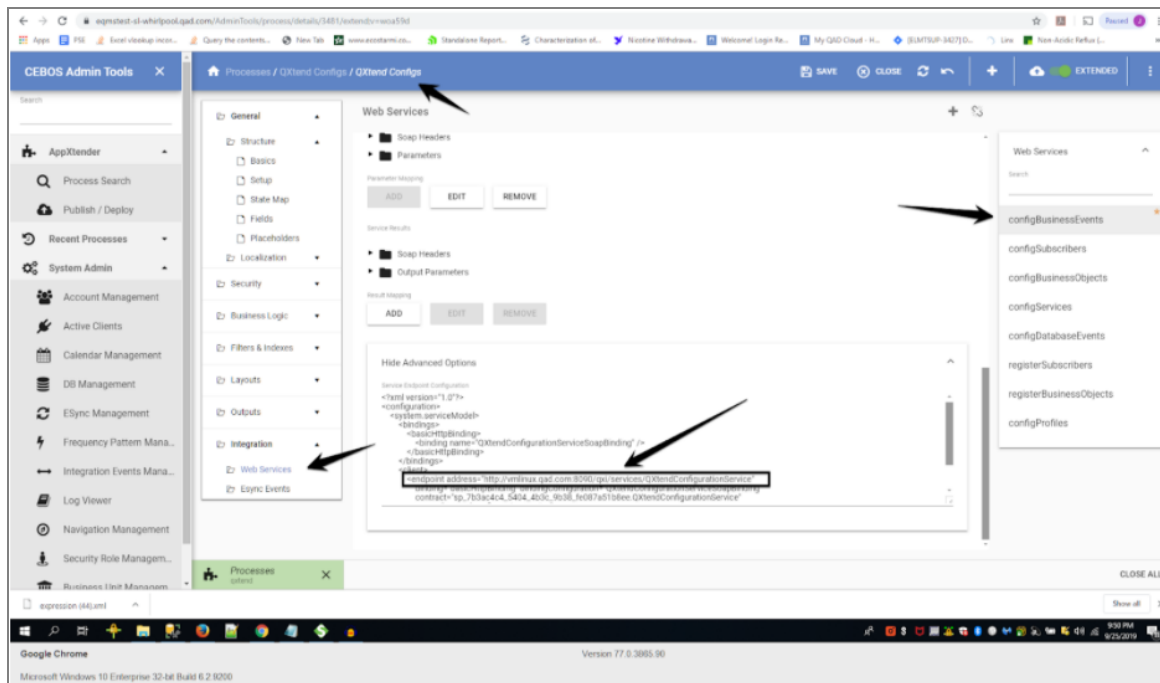
To do this, follow the instructions below:

- Open the QXtend Configs process in the QAD EQMS AppXtender.
- Navigate to the Integration tab. Several pre-configured services are visible.
- (If version 2019 or later)** Click Extend to change the process. Then click the paperclip icon for each service to extended services, making them editable.
- (If prior to version 2019)** For each service, click Show Advanced Settings. A text box appears labeled Service Endpoint Configuration.
- Within the text box, reset the existing place holder QXtend inbound URL to the correct QXtend inbound URL, provided by the customer or QAD.

Note: Only change up to "qxi" ("http://vmlinux.qad.com:8090/qxi/"). Do not change any other URLs on the Integration tab or take any other action. You only need to change the value under Service Endpoint Configuration.

- Perform this change 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
 - QXtend Database Event Configs.
- Soft publish each process until you reach the last one.
- When you reach the last process, publish and deploy, as seen in the screen shot below.

Fig. 10: EQMS AppXtender, QXtend Configs process, Web Services section



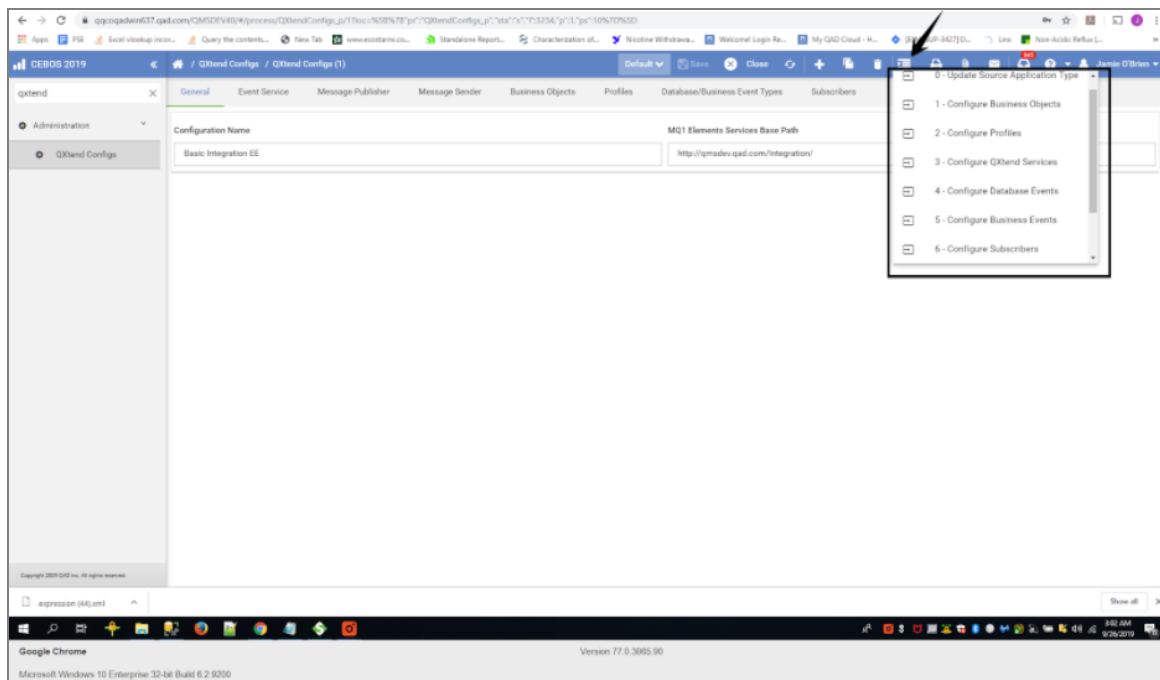
At this point, both QXtend and QAD EQMS are ready to auto-configure the integration interfaces.

Step 5

Return to the Basic Integration EE/SE record from the QXtend Configs process search screen in QAD EQMS (see "Step 3 (EE Only)" 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 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. 11: QAD EQMS, QXtend Configs process screen, Commands list



The best way to confirm success is to check the Configuration tab in QXO. 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

Step 6

The final step is to force publish the data. 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"
- Status = "PEND"

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.

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.

For EE, the order is as follows:

- Domains (bdomain)
- Entities (bcompany)
- Sites
- Departments
- Inventory Status Codes
- Item Group
- Item Type
- Product Lines
- UofMs
- Inventory Locations
- Work Centers
- Processes
- Customers (bdebtor)
- Customer Ship-To (bdebtorShipTo)
- Suppliers (bcreditor)
- Business Relation (bbusinessrelation)
- Items
- Customer Items
- Supplier Items

For SE, the order is as follows:

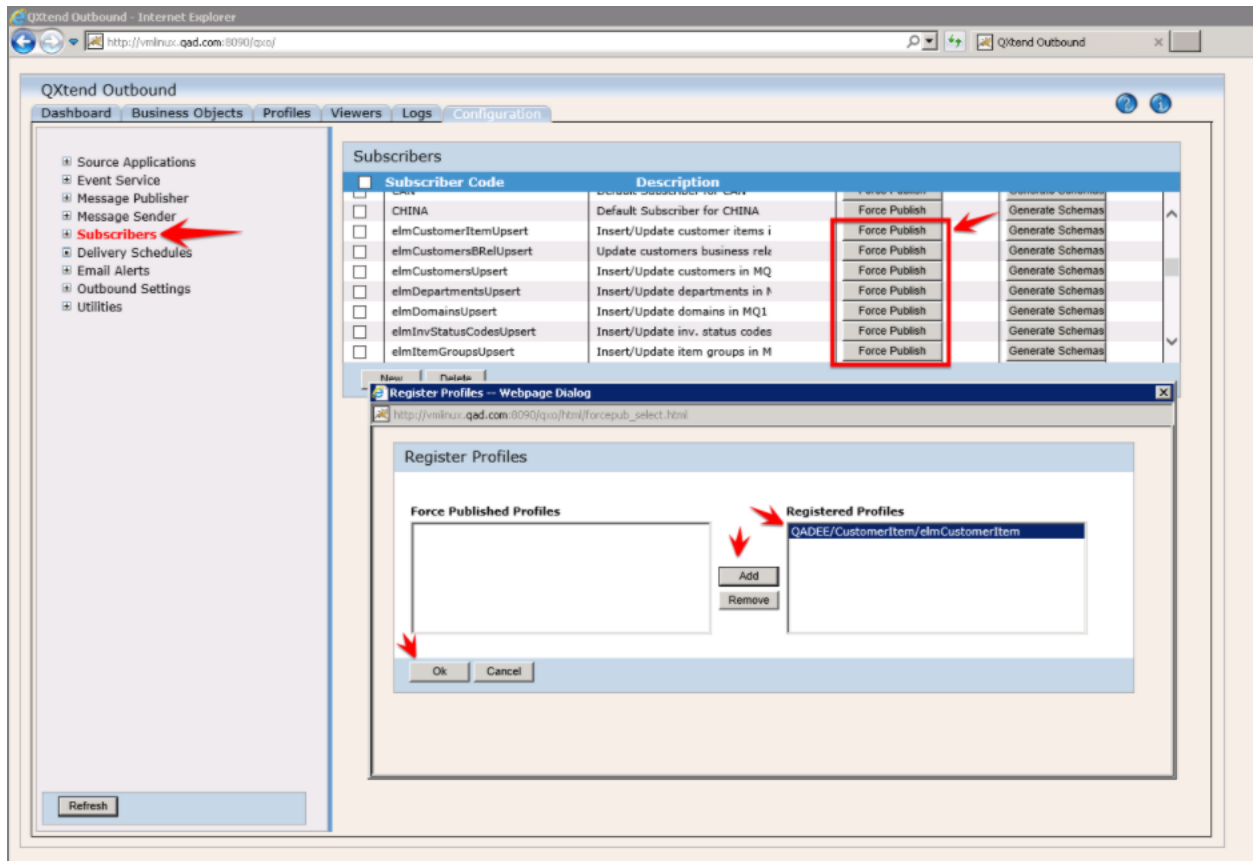
- Domains
- Entities
- Sites
- Departments
- Inventory Status Codes
- Item Group
- Item Type
- Product Lines
- UofMs
- Inventory Locations
- Work Centers

- Processes
- Customers
- Suppliers
- Ship To
- Items
- Customer Items
- Supplier Items

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

1. Navigate to the Configuration tab in QXO.
2. Click Subscribers.
3. Find the correct interfaces in the list.
4. Click Force Publish.
5. Complete the wizard shown in *Fig. 12: QXtend Outbound, Configuration tab, Register Profiles wizard* below.

Fig. 12: QXtend Outbound, Configuration tab, Register Profiles wizard



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

- Domains (bdomains)
- Customers (bdebtors)

- Customer Ship-Tos (bdeborShipTo)
- Suppliers (bcreditors)
- Business Relations (bbusinessreleation)

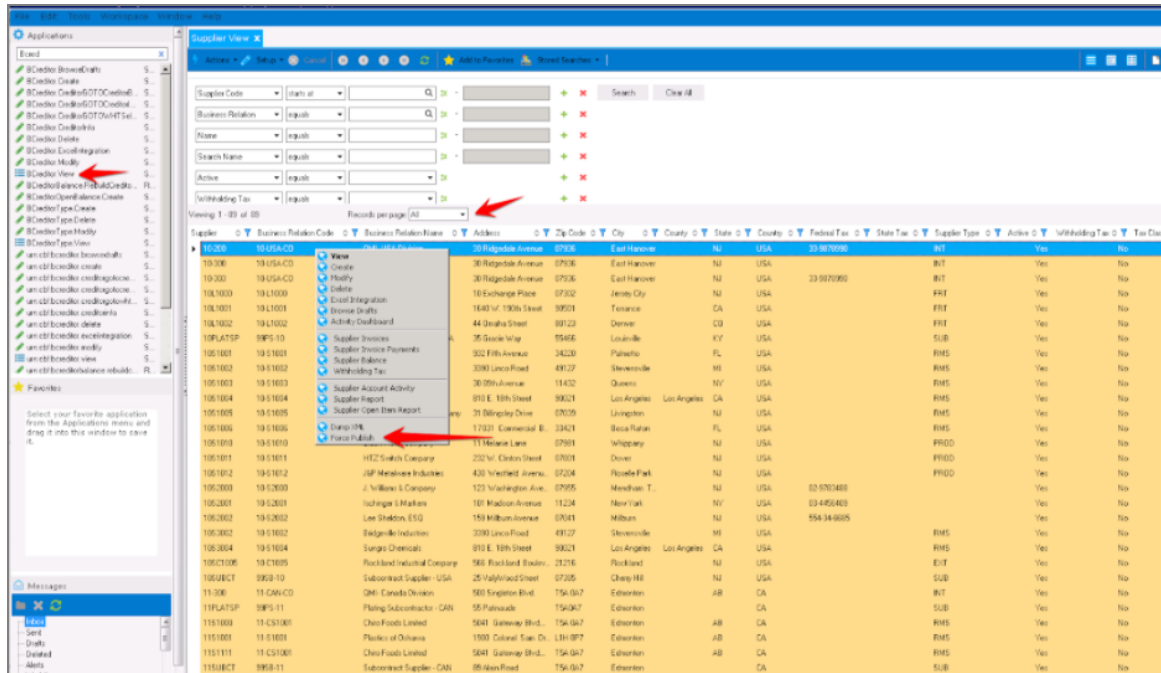
For these interfaces, the following steps must be performed to publish financial data from EE:

- 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. 13: QAD .Net UI Client, force publishing suppliers* below for an example.

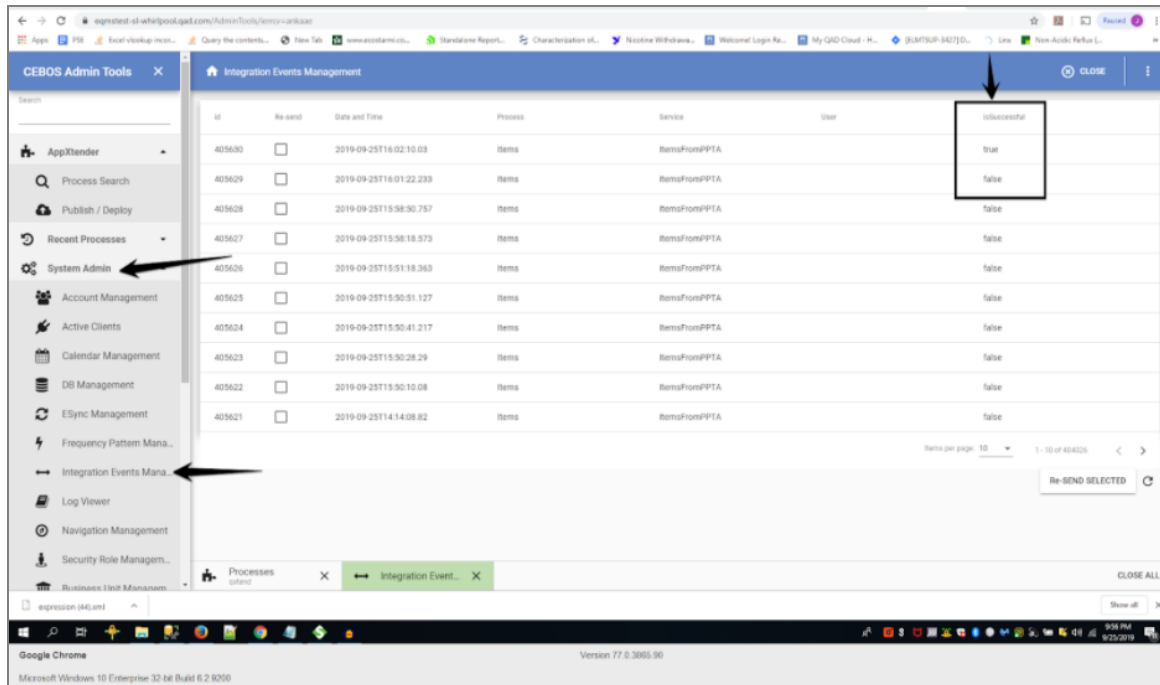
Reminder: To access a given workspace, you must select it from the Workspace drop-down menu in the top screen toolbar while logged into the QAD .Net UI Client.

Fig. 13: 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. 14: QAD EQMS, Integration Events Management process search screen* below). Filter results for "False". Only records whose "issuccessful" column reads "False" should be returned.

Fig. 14: QAD EQMS, Integration Events Management process search screen



Creating Custom Integrations

- Training video covering QAD EQMS side SOAP integration configuration:
<https://drive.google.com/open?id=1bhWC1PJpKBZVKsm3UJiZGVvpb2GAGeEcc>
- Training video covering QAD EQMS side REST integration configuration:
<https://drive.google.com/open?id=1jcOpx3vYz-dWJNPWKOGtKoM-5wWdr7lt>
- Training video covering QXtend QXO components:
https://drive.google.com/open?id=1dW_PHxyRV6Q9dX1rU-VEJCePzaoHIYyZ

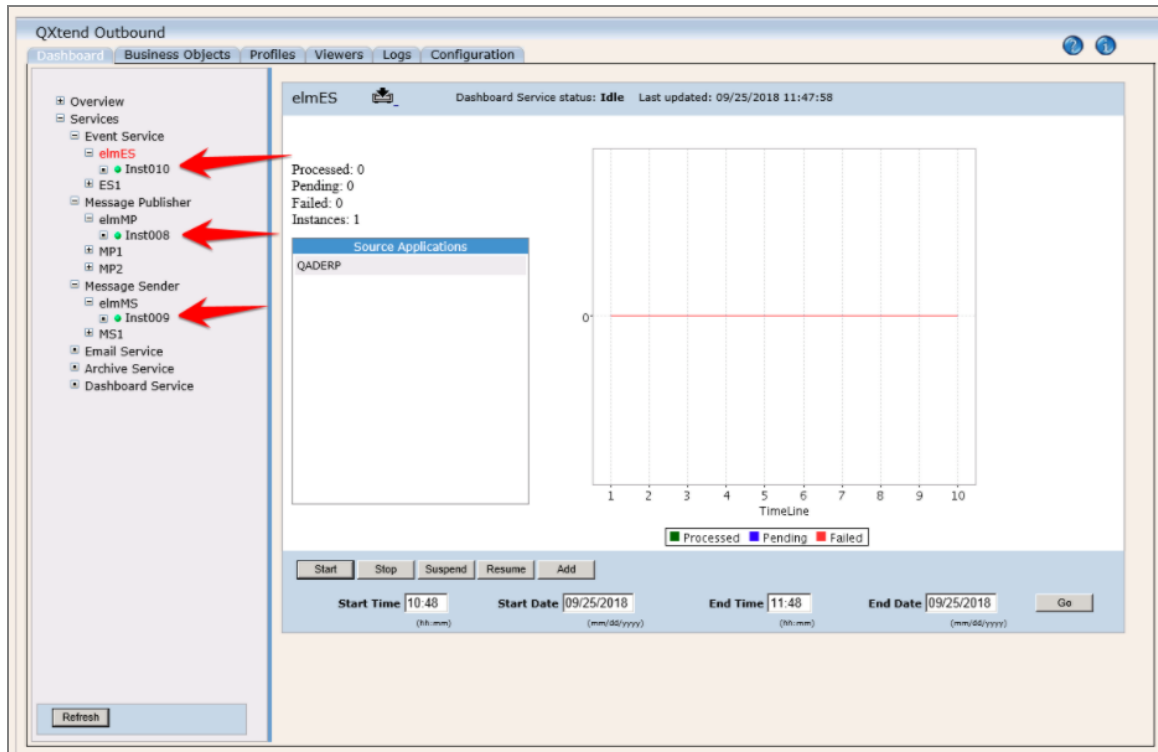
Troubleshooting Integrations / Common Problems

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 down is shown below the service (see *Fig. 15: QXtend Outbound, Dashboard tab, running services* below). If the services are not running, then select the service name and click Start.

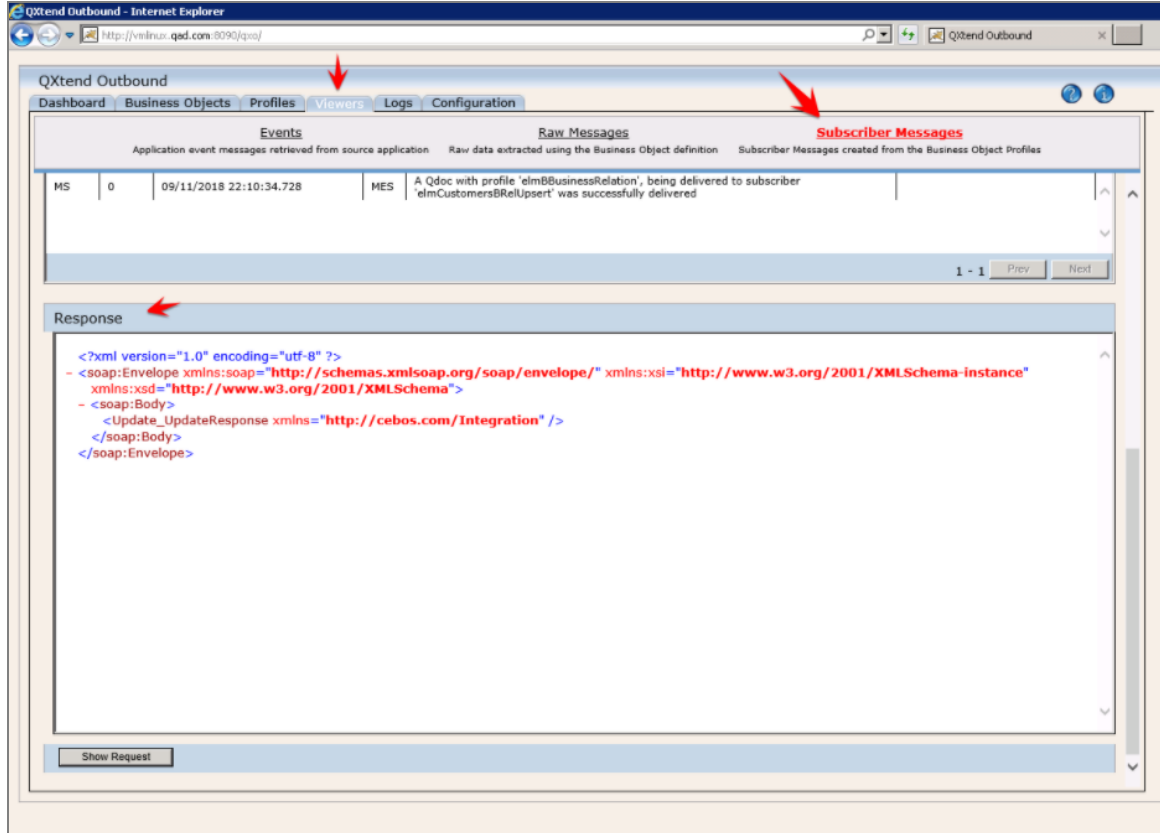
Fig. 15: QXtend Outbound, Dashboard tab, running services



- 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.

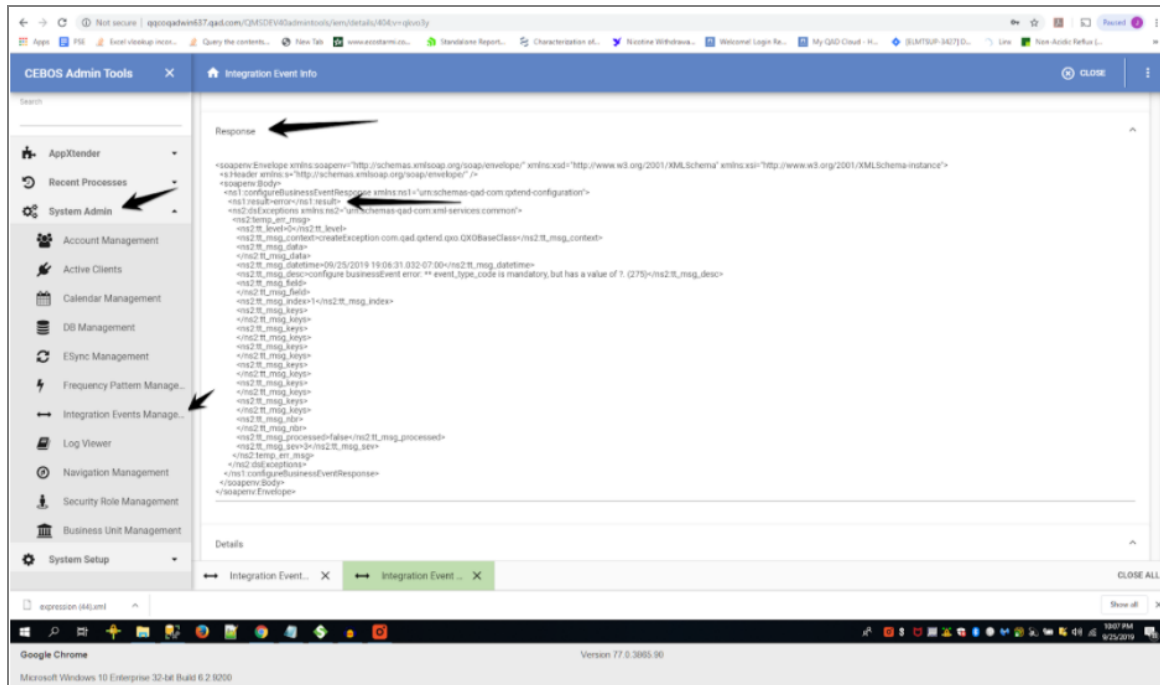
If no messages appear in the Subscriber log, then there may be a QXtend configuration issue. If messages are appearing in the log, then you can view the error messages by highlighting the message and clicking Response, as shown in Fig. 16: QXtend Outbound, Viewers tab, Subscriber Messages below.

Fig. 16: 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. 17: QAD EQMS, Integration Events Manager, Integration Even Info



If QXtender 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.

Chapter 3

Appendix

Appendix...30

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>
- 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 2019.1:
<https://drive.google.com/open?id=1ffvgGt8EzJYJZCw5guaSOhdR4D1cpexb>
- 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/MQ1/Setting+Up+REST+Integration>
- Training video covering QAD EQMS side SOAP integration configuration:
<https://drive.google.com/open?id=1bhWC1PJpKBZVKsm3UJiZGVvpb2GAgEcc>
- Training video covering QAD EQMS side REST integration configuration:
<https://drive.google.com/open?id=1jcOpx3vYz-dWJNPWKOgtKoM-5wWdr7lt>
- Training video covering QXtend QXO components:
https://drive.google.com/open?id=1dW_PHxyRV6Q9dX1rU-VEJCePzaoHIYyZ
- List of QXtend QXO components, what they are for, and who is responsible for them:
<https://drive.google.com/open?id=16CfdHX0uRcwBX9j1ZXZQsQbsROIluOoAWHrUbbDGltY>