QAD Adaptive Applications

Technical Reference

SOAP Integration Configuration

70-3510-2025

QAD QMS Applications version 2025

March 2025

Copyright

QAD Inc.

100 Innovation Place

Santa Barbara, CA 93108

Phone: +1 (805) 566-6100

http://www.qad.com

SOAP Integration Configuration Change Summary

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

Date/Version	Description	Reference	Changed By
SEPT 2022/v2022.1	Initial upload	--	BHH
MAR 2023/v2023	Updated versioning	--	BHH
MAR 2024/v2024	Updated versioning	--	BHH
SEPT 2024/v2024.1	Updated versioning	--	BHH
MAR 2025/v2025	Updated versioning	--	BHH

Chapter 1

Introduction

Introduction

This document is meant to serve as a guide for setting up and configuring custom QAD EQMS <=> QAD Adaptive ERP integration. It is assumed the user is already familiar with level 1 Out of the Box (OTB) integration configuration from the QAD EQMS Integration Configuration technical reference.

About This Guide

This user guide focuses on:

- Inbound SOAP integration
- Outbound SOAP integration

Chapter 2

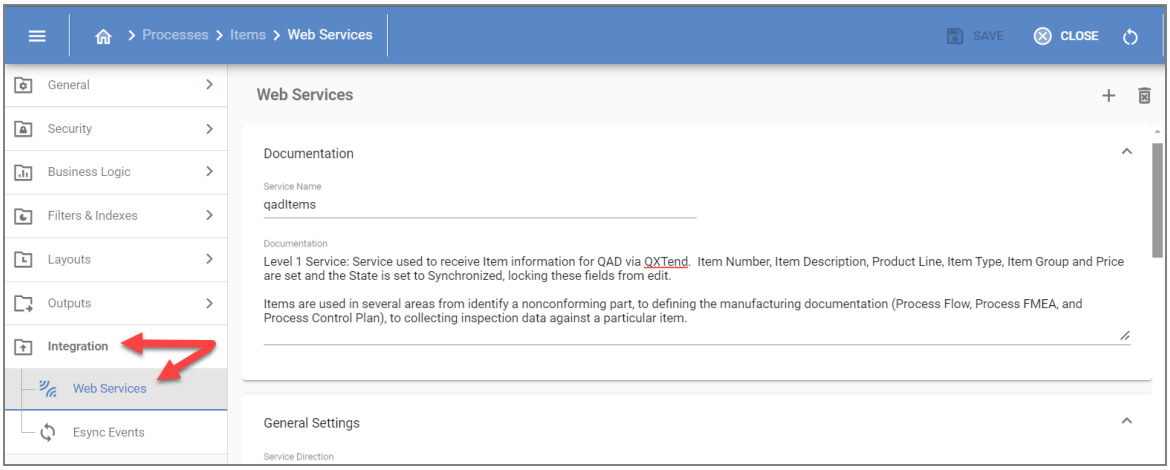
Configuring a SOAP Service

Instructions

SOAP Integration Between QAD EQMS and QAD Adaptive ERP

Both inbound and outbound SOAP services are managed in AppXtender for a process under the Integration > Web Services menu. When creating a new web service, it is important to assign a name and description that will help future developers or engineers understand the intent and goal of the service you created.

Fig. 1: AppXtender, Web Services screen



Service Name

This is the name of the service as it appears throughout EQMS, including the endpoint URL if the service is inbound.

Documentation

Use this field to describe exactly what the integration service does, including the names of the systems sending and receiving the data, the content involved, and the business use case. Additionally, indicate to what extent the integrated data is a prerequisite for additional integration service, or is dependent on another service as its prerequisite.

Configuring a Create Service

Service Direction

Select the Create option to configure an inbound service–that is, one that will create data in EQMS. You are presented with different field choices in what follows depending on your choice between *Create* and *Call*.

The endpoint URL for the inbound service uses the following format:

https://<hostname>/WebApi/Integration/<ServiceName>.asmx

Example: https://mysite.qad.com/WebApi/Integration/qadItems.asmx

Placing the endpoint URL into a web browser will reveal a page from which you can select the Service Description link to retrieve the WSDL, and the operation/method links (such as upsert) to view the SOAP 1.1 and 1.2 definitions.

Note: The Service Description can be saved as a .WSDL file.

Fig. 2: Service Description, link

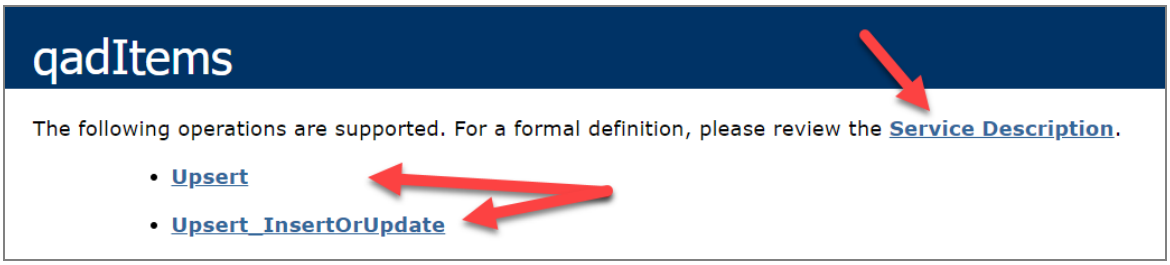
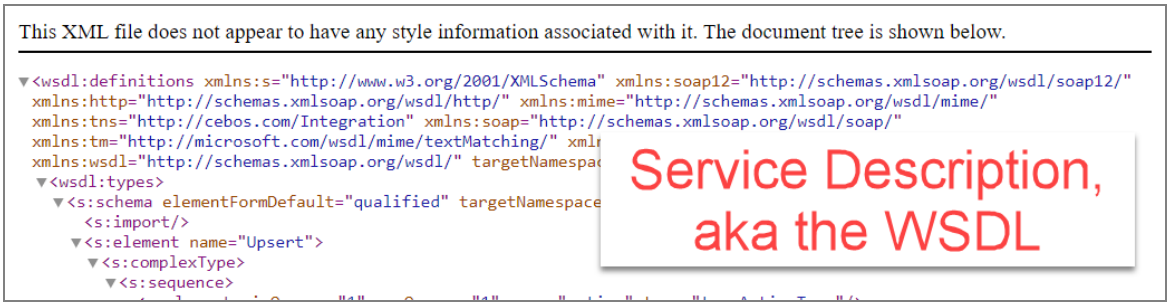


Fig. 3: Service Description, WSDL



EQMS supports SOAP 1.1 and 1.2. Review the definitions to determine what specifically EQMS requires in order for it to accept integration through that particular service (such as qadItems) using that particular method (such as Upsert). As you proceed through this guide, you will see comparisons where EQMS setup corresponds to the XML found in the SOAP definition.

Fig. 4: SOAP 1.1

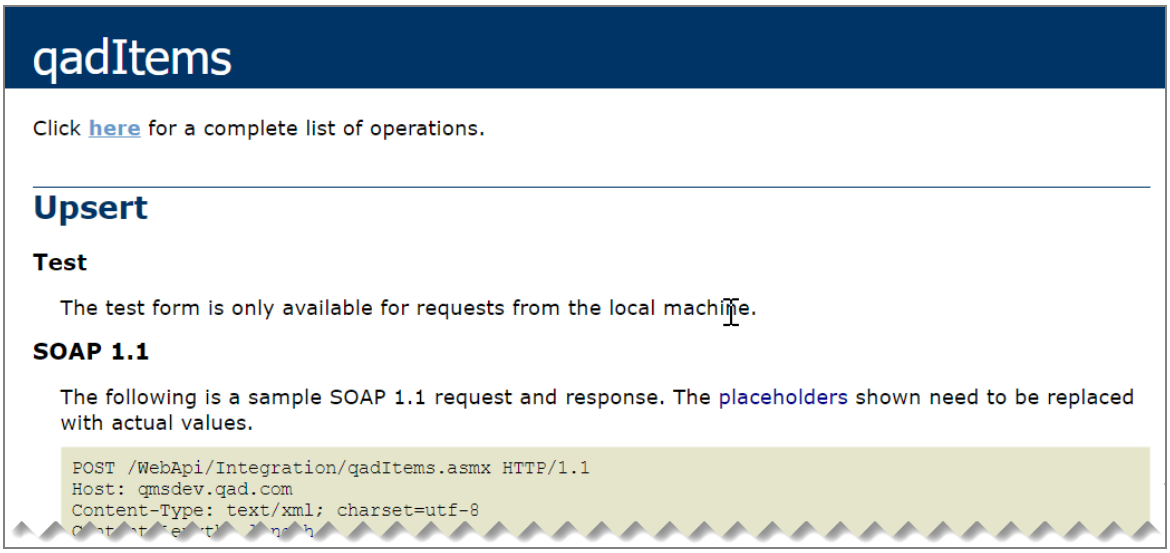
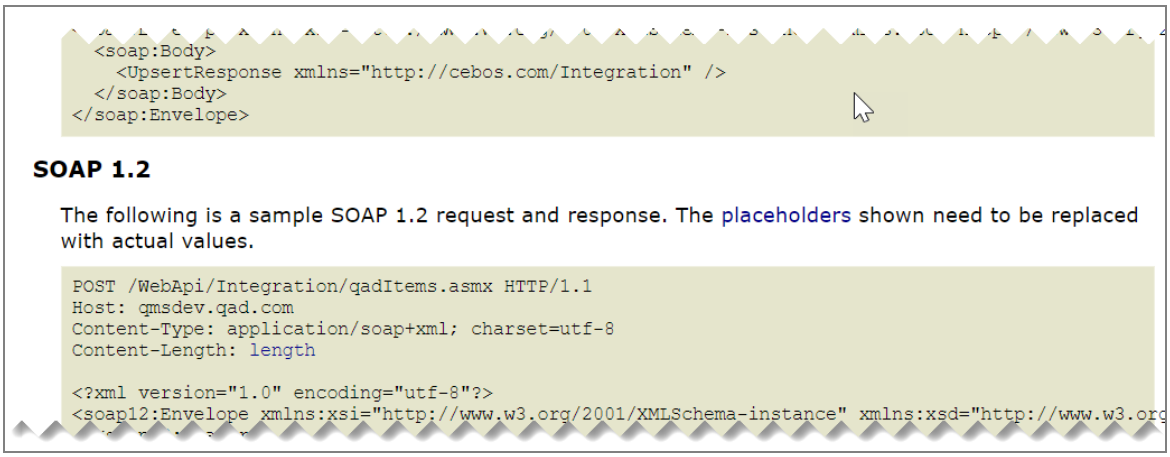


Fig. 5: SOAP 1.2



Namespace

The default value here should not be changed.

Service Type

For a *Create* SOAP integration, use Web Method as your service type.

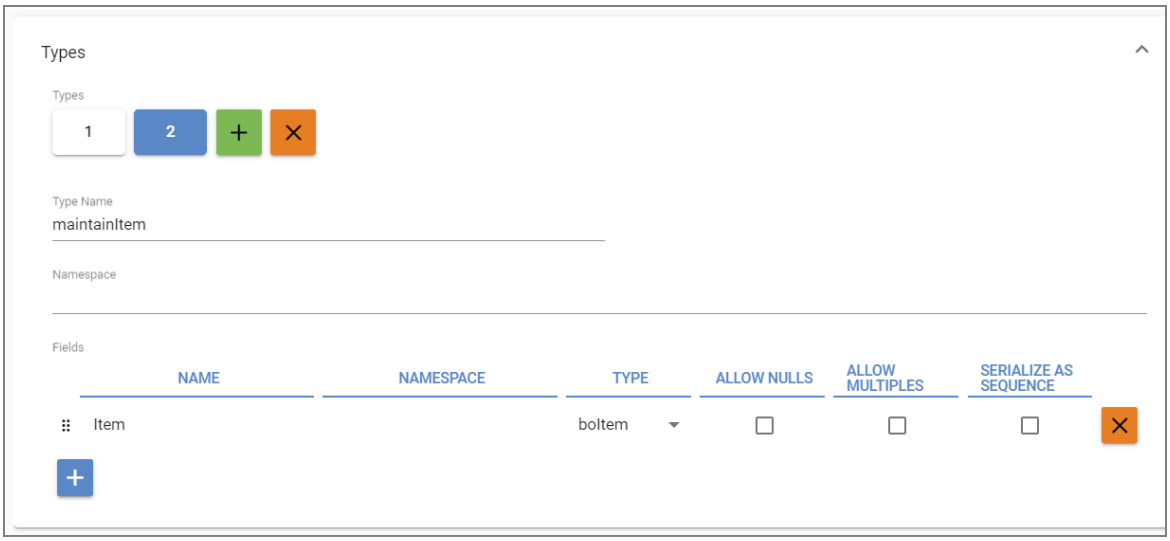
Log Integration Events

Typically, this check box should be selected. It causes all EQMS-side integration activity associated with the particular service to be recorded in the EQMS AppXtender > System Admin > Integration Events Manager. This feature can be disabled if logging the service activity would overly burden system load or be redundant; for example, if QXtend already logs the service actions, it might not be necessary to also keep logs in EQMS.

Configure a Type

The Types control is located at the bottom of the Web Service screen. It is often necessary to set up one or more types in order to completely set up a Method (described later).

Fig. 6: Web Service screen, Types control



Type Name

In the example above, the Type Name is "maintainItem". This name will not correspond to anything in the SOAP definition. Instead, it is a grouping name under which further fields or nodes can be stacked.

Namespace

Ordinarily this field should be left blank when integrating with QAD ERP. However, some third party products may require it.When adding a new Type, this field has a default value; remove it when not needed.

Fields

The following list defines specific fields or nodes used with the web service.

- **Name:** This field corresponds to a node in the SOAP definition. The example above uses the name "Item", which corresponds to the following:

Fig. 7: SOAP request, Item name

- **Namespace:** The namespace typically should not be used in any of the fields or nodes when integrating with QXtend, but other applications may require it at the node level. Be sure to remove the value from the input parameter's namespace field if it is not needed.
- **Type:** Standard options are Boolean, Byte, DateTime, Decimal, Double, Int, and String. Further custom types can be set up as needed to facilitate additional nodes in the SOAP definition hierarchy. For example, in the "maintainItem" example above, the custom type named "boItem" is assigned to the "Item" field. In the SOAP definition, this creates a node and field list under the top level node "dsItem" (see the example below). Note the *Type Name* "boItem" does not appear in the SOAP definition, but the fields defined under the "boItem" group appear in the SOAP definition instead.

Fig. 8: SOAP request, node and field list

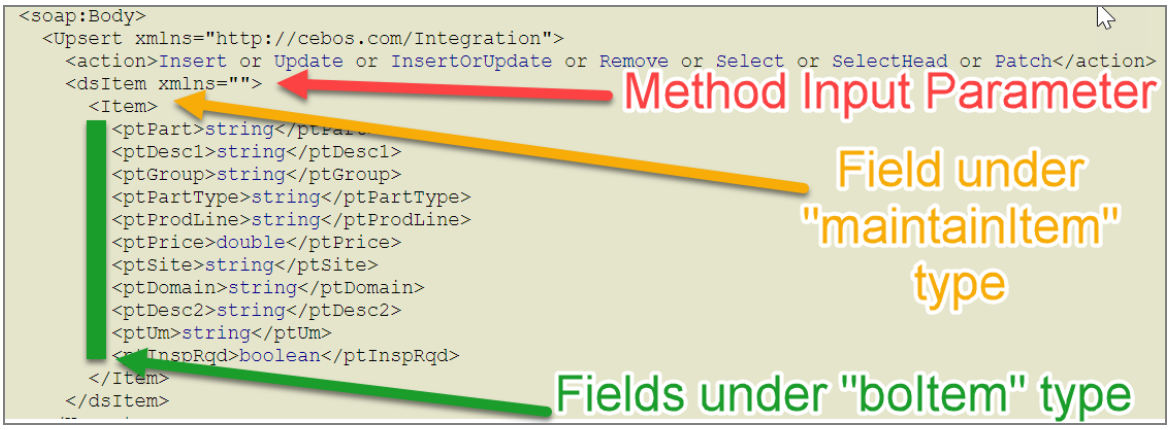
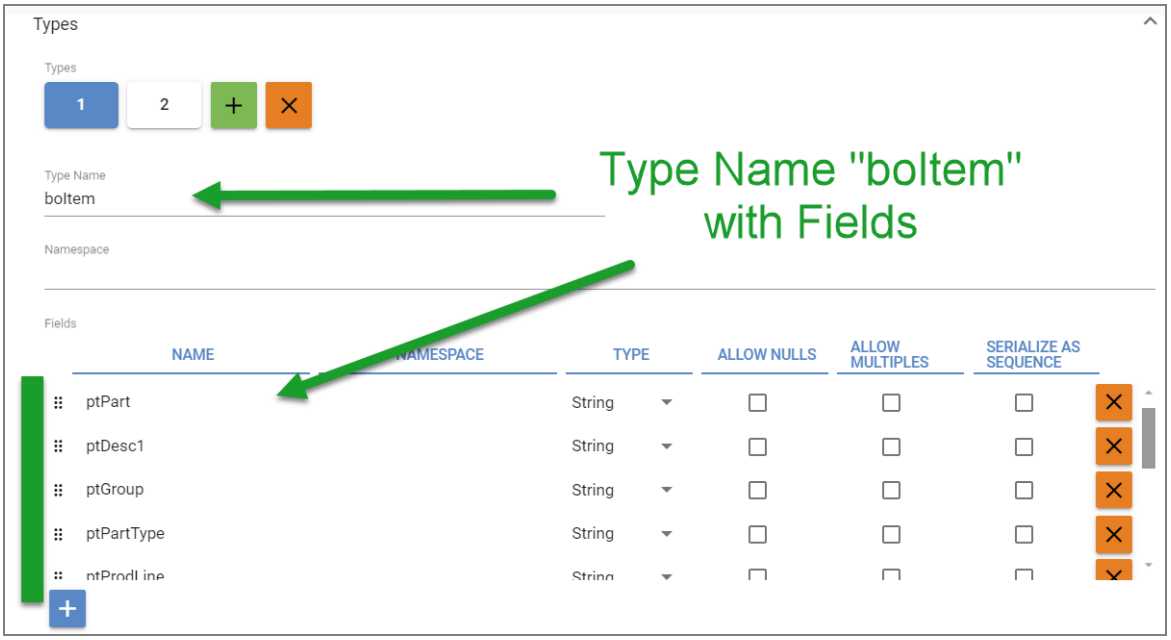


Fig. 9: Types control, boItem type name with fields



- **Allow Nulls:** Permits reception of a null value from the integration source. Be careful not to accept nulls if it will conflict with a required field setting in the EQMS destination process.
- **Allow Multiples:** Converts the field or node into an array.
- **Serialize as Sequence:** This is an option for list type fields (MCR, Checklist, etc.). The default format in the XML of the WSDL file is an array, where the sender will need to send the data as an array to all the values. But if "Serialize as Sequence" is selected, it converts this into a comma-separated string value instead. If you publish and view the WSDL after setting each option (selected or deselected) you should see the difference.

Configure a Method

Method Name

Give the new Method a name that describes what it does. It should be as brief as possible. It will be incorporated into the endpoint mapping URL used by the service.

Namespace

For integration with QXtend, the Namespace field should match the one from the General section.

Username & Password

The credentials required for authentication in the SOAP header. Complete the login credential fields if you want to require authentication using those credentials when integrating with the EQMS endpoint. Leave these fields blank if you do not wish to enforce authentication.

Input Parameters

It may be necessary to configure Type information (see above) to set up input parameters.

- **Name:** Create a concise name for the parameter. The parameter name will appear as a top-level node for this method. For example, you can associate the "qadItems" service with the "dsItem" input parameter. This corresponds to the <dsItem> node in the SOAP definition.

Fig. 10: SOAP request, dsItem input parameter

- **Namespace:** The namespace typically should not be used in any of the input parameters or nodes when integrating with QXtend, but other applications may require it at the node level. Be sure to remove the value from the input parameter's namespace field if it is not needed.
- **Type:** Standard options are Boolean, Byte, DateTime, Decimal, Double, Int, and String. Custom types can be set up as needed to facilitate further nodes in the SOAP definition hierarchy.
- **Allow Nulls:** Permits reception of a null value from the integration source.
- **Allow Multiples:** Converts the parameter into an array.
- **Serialize as Sequence:** This is an option for list type fields (MCR, Checklist, and so on). The default format in the XML of the WSDL file is an array, where the sender will need to send the data as an array to all the values. But if "Serialize as Sequence" is selected, it converts this into a comma-separated string value instead. If you publish and view the WSDL after setting each option (selected or deselected) you should see the difference.

Endpoints

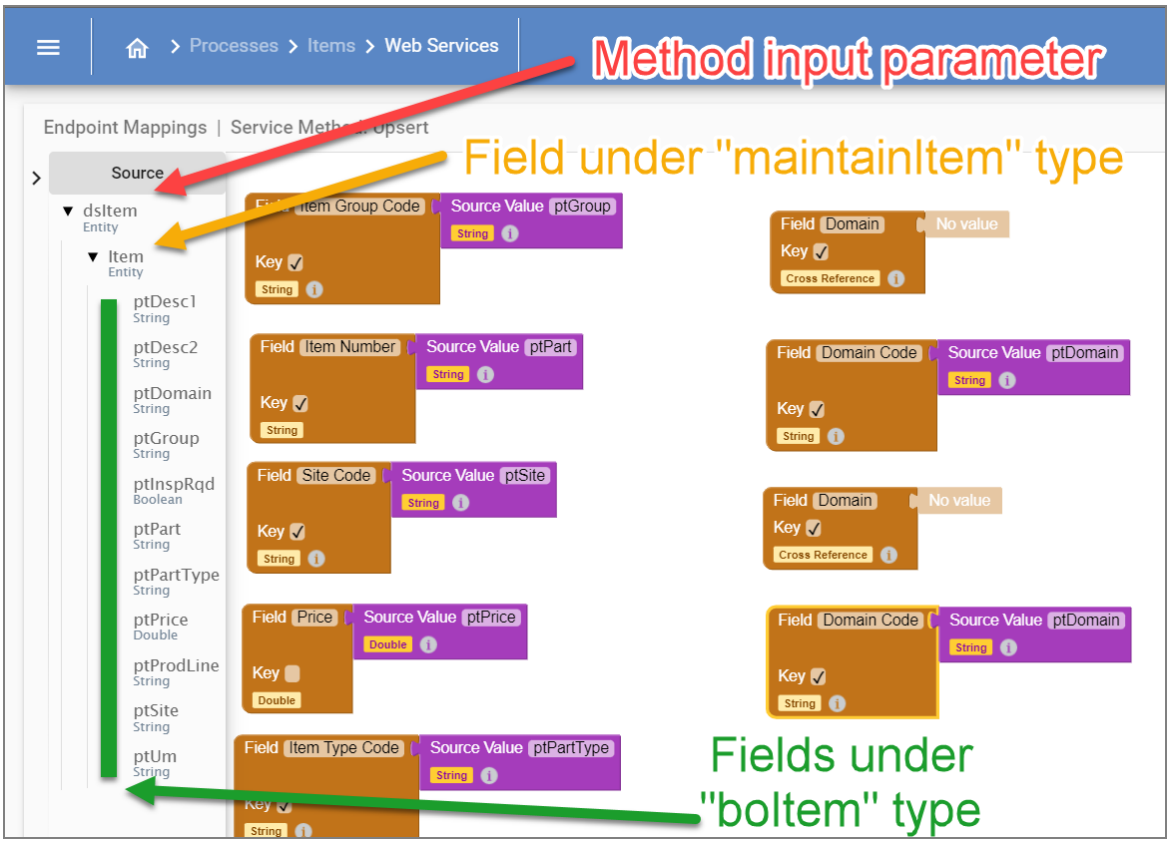
Select as many of the endpoint actions as you require for the service. The source application integrating with EQMS will use whichever of them is needed to carry out its task.

Fig. 11: Endpoint Actions

Endpoint Mapping

Select the Endpoint Mapping button to reveal the structure set up by the Method configuration.

Fig. 12: Web Services screen, Endpoint Mappings

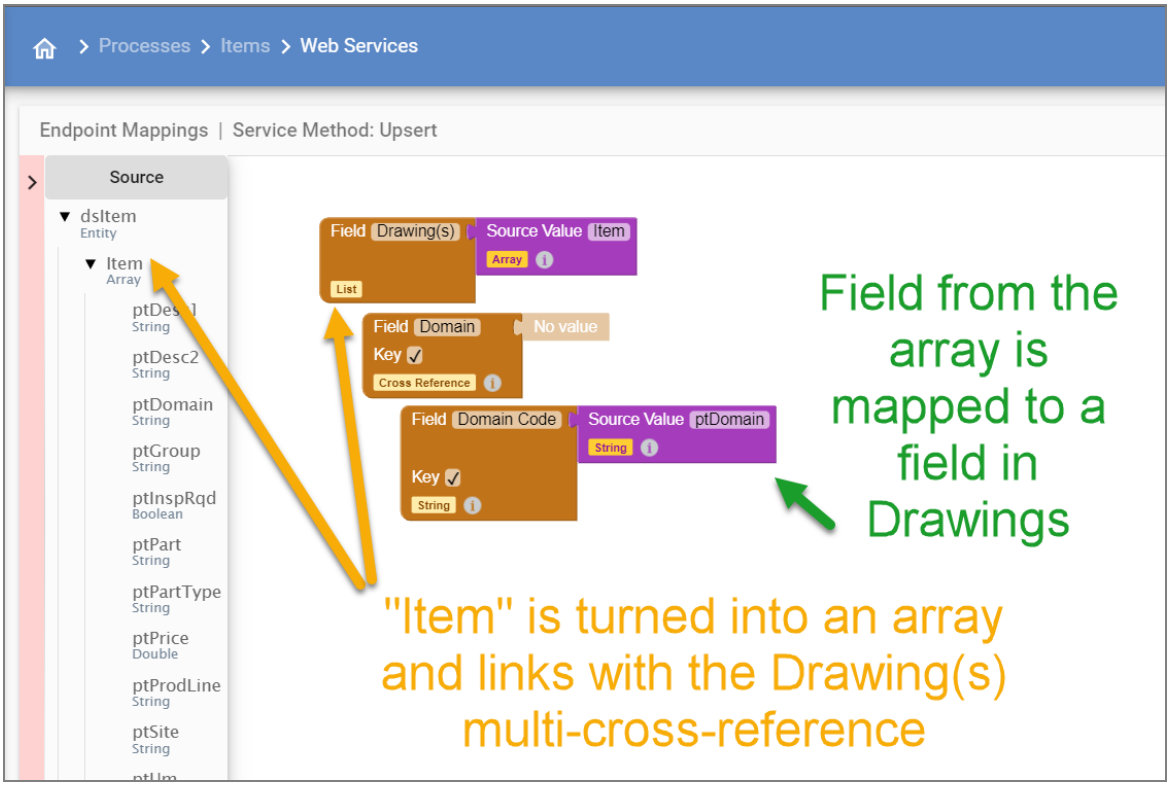


From here, field matching between Source and the endpoint process (in this example, the Items process) works very much like configuring a transaction in EQMS. Fields are represented as blocks, with Source field blocks in purple and endpoint process blocks in brown. Key fields are flagged using the "Key" check box on the endpoint process block.

In the example above, note that there are two instances where Domain and Domain Code are flagged as key, but the Domain field has no source field mapped to it. Click the (i) bubble to reveal that these are Domain fields belonging to cross-referenced records, not the Items process itself. The Item record has a cross-reference to Item Group and another to Item Type, and in each case the linked Item Group/Type record's > linked Domain record's > Domain Code field is key. This means that the Domain Code is not key for the entire Item record; rather it is key just to find the right Domain to link. **The overall record needs at least one key field flagged, but also each cross-reference on the overall record needs at least one key flagged.**

An array in EQMS Expression Builder is compatible with a cross-reference List type. To map an array to a LIst type in EQMS, such as to Drawings on the Items process, the top level of the array should link to the List field. Then one or more fields in the array should be linked to field matches in the cross-reference list.

Fig. 13: Web Services screen, Endpoint Mappings, mapping an array



An important distinction between building expressions for transactions vs integration is that if a cross-referenced record cannot be found by integration, then a Method with insert ability will attempt to create the record. For example, if a Domain with the key Domain Code field does not exist in EQMS, then the system will try to create a new Domain record that has that Domain Code and link it to the record. If, though, there are required fields o the Domain process that are not being populated from integration, then the service will be unable to finish creating the new Domain record and will return an error instead. The error can be found in the EQMS AdminTools > System Admin > Integration Events Management screen.

Result Type and Mapping

Configure the Result Type with mapping option to send a custom response back to the application originating the integration event. Like with Method input parameters and Type fields, QAD uses a structure created from a Type and maps the response values to it. The structure from that type with the mapped values is sent as a response to the integration originator. Note that this feature is not required and is often not used.

Configuring a Call Service

Service Direction

Select the *Call* option to configure an outbound service, that is, one that will send data from EQMS. You are presented with different field choices in what follows depending on your choice between *Create* and *Call*. It is assumed from here that you are familiar with how to set up a *Create* service as described above.

Service Type

Ordinarily for a *Call* SOAP integration use Web Method as your service type.

The URL type just involves mapping a file location or URL to the Result field in the mapping. The service would then be tied to a transaction or a command in EQMS.

The REST type is used for configuration RESTful integration, and is outside the scope of this guide.

Execute Only Once on Save

Select this check box to prevent an event from firing multiple times when the process' execution plan fires. Depending on the complexity of the execution plan and the number of re-saves triggered by it, a service may end up sending its data many times, even though the end user saved a record only once. This box is used to prevent this situation, and it should be marked most of the time.

Address

This field holds the endpoint URL for obtaining the WSDL from the QXtend or third party endpoint application.

Username & Password

If credentials are required to retrieve the WSDL, enter them here.

Get Description

Select this button to retrieve the WSDL. This will automatically build most of what needs to be configured to complete this service.

Note: Retrieving the WSDL will reset the entire outbound configuration, and you will have to configure it anew. If using this button to refresh an existing configuration because the WSDL at the endpoint was updated, then be sure to first back up your mapping via export from Expression Builder. To avoid this problem. it is best practice to configure an outbound integration only once the WSDL at the endpoint is stable.

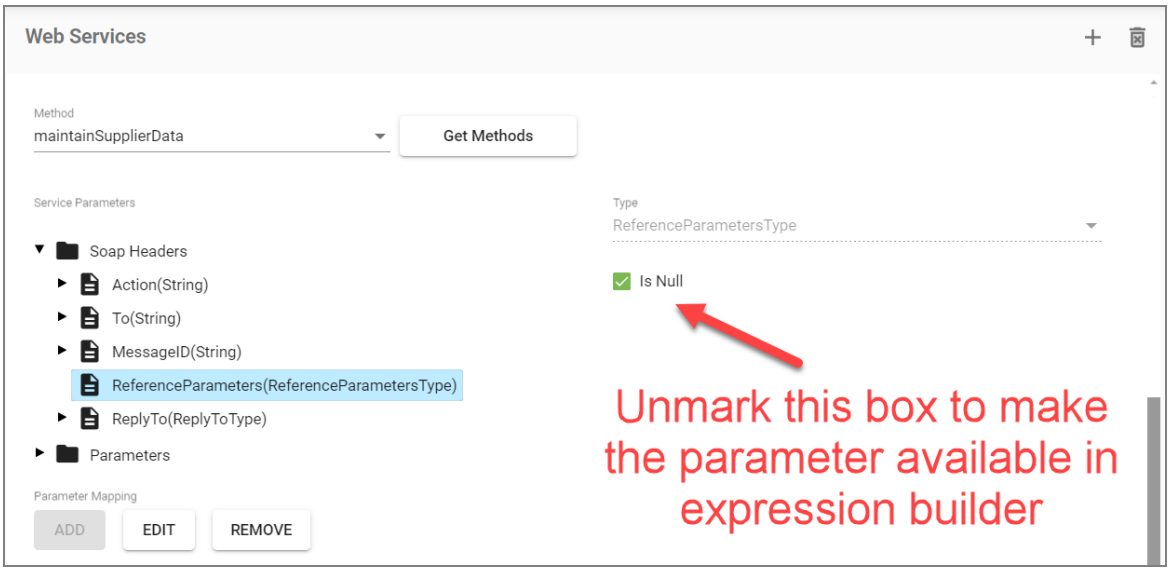
Method & Get Method

Use the "Get Method" button to retrieve the available methods as defined in the WSDL. Use the "Method" selection to choose which method this service will use. This will populate all the service parameters.

Service Parameters

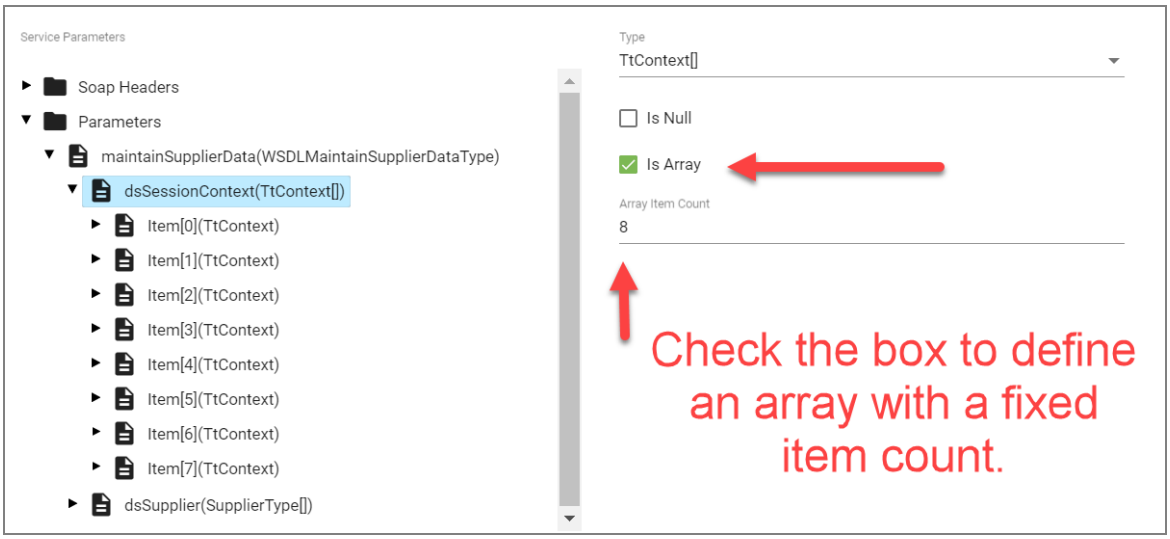
Having populated these automatically by selecting a Method, expand the parameters list and select each parameter you wish to use for Parameter Mapping. Deselect the "IsNull" check box to include it in the XML and in Parameter Mapping (below).

Fig. 14: Web Services screen, Service Parameters



When selecting from the Parameters tree, an array has an option to limit the number of inputs to a fixed amount rather than a dynamic amount.

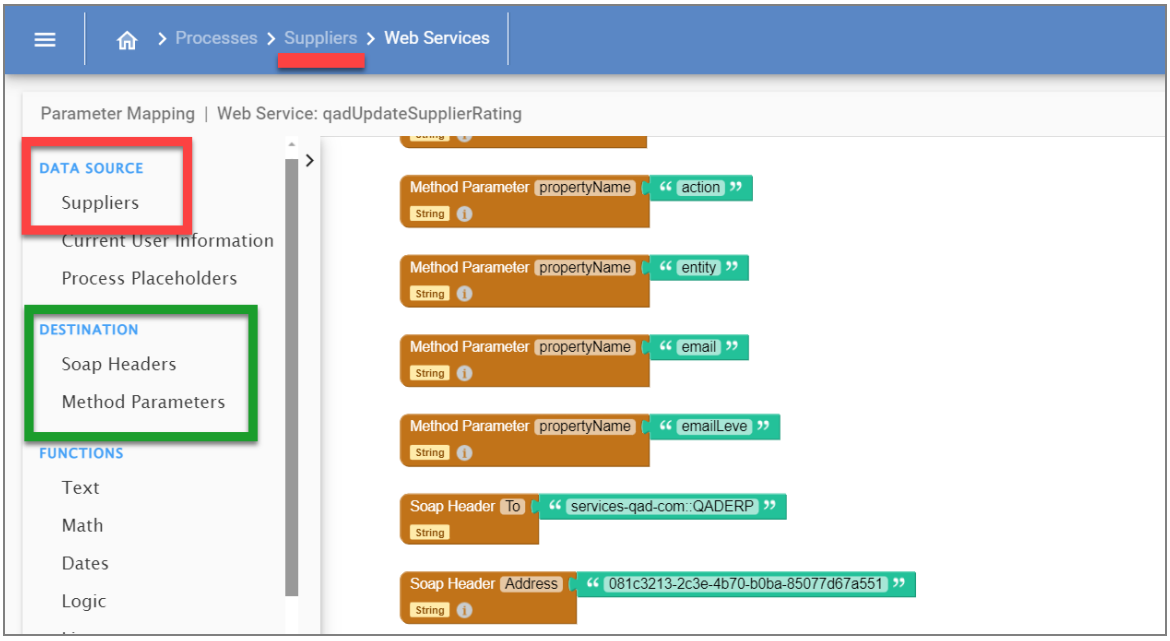
Fig. 15: Web Services screen, Service Parameters



Parameter Mapping

The Service Parameter structure configured above appears in the mapping interface. This works just like above when defining a Create service, but the relationship is reversed and there is no need to handle key matching.

Fig. 16: Web Services screen, Parameter Mapping



When integrating with QAD ERP, the "Address" from the SOAP Header is usually the same. The"To" field is also the same, except for the part that reads "QADERP".

Service Results

The Service Results tree is configured from the WSDL. What is left open for configuration is the mapping to send specific data back in response. Most commonly, this is used to populate a custom check box on an EQMS process, indicating a message has been sent.

Advanced Results

Normally this section does not need to be touched. The only area of real importance when configuring an outbound SOAP integration is the "endpoint address". Level 1 integrations will have this configured as part of the Level 1 setup workflow. The "endpoint address" should point at the endpoint URL of the service receiving the integration request. IF the outbound integration will be promoted with a system refresh from Development to Test, then the "endpoint address" will need to be updated to point at the new test endpoint. Save and publish any changes.

Fig. 17: Service Results tree, Advanced Results