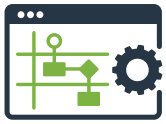


BPA
Platform

Technical Overview

Microsoft Exchange Tool Pack v4.4

codeless
PLATFORMS

Copyright

The copyright in this document is owned by Orbis Software Ltd T/A Codeless Platforms 2021. All rights reserved.

This publication may not, in whole or part, be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form or by any means without the prior written consent of Orbis Software Ltd T/A Codeless Platforms.

Head Office:

Codeless Platforms

Suite 2 Bourne Gate

25 Bourne Valley Road

Poole

BH12 1DY

United Kingdom

Tel: +44 (0) 330 99 88 700

Email: *enquiries@codelessplatforms.com*

Trademarks

Orbis Software Ltd T/A Codeless Platforms owns the registered trademark "TaskCentre®".

Microsoft®, Windows®, Windows Server®, SQL Server®, Internet Explorer®, and Outlook® are all trademarks or registered trademarks of the Microsoft Corporation in the United States and other countries.

All other Trademarks used are acknowledged as the property of their respective owners.

The information provided in this publication may contain errors, omissions, or typographical errors or may be out of date. Orbis Software Ltd T/A Codeless Platforms may change, delete, or update any published information at any time and without prior notice. The information published in this document is provided for informational purposes only and is not binding on Orbis Software Ltd T/A Codeless Platforms in any way except to the extent that it is specifically indicated to be so.

Contents

Introduction ... **1**

The Microsoft Exchange Tool Pack ... 1

System Requirements ... 2

Minimum BPA Platform Software ... 2

Minimum Microsoft Exchange Version ... 2

Architecture ... **3**

Working with Other Tools ... **4**

Consuming from Other Tools ... 4

Objects Consumed ... 5

Exposing to Other Tools ... 5

Objects Exposed ... 6

Where Can the XML Output be Used? ... 7

Error Handling ... 7

About the Microsoft Exchange Event Agent ... **8**

Using the Microsoft Exchange Event Agent Configuration ... 9

Adding the SQL Server Connection ... 10

Adding the Windows Service User Account ... 11

Adding the Push Endpoint ... 12

Adding the Exchange Web Service Connection ... 12

Adding the BPA Platform Connection ... 13

About the Microsoft Exchange Event Agent Instance Manager ... 13

About the Microsoft Exchange Connector Agent ... **15**

Adding a Microsoft Exchange Connection ... **15**

Setting the Time Out ... 16

Using Extended Logging ... 16

About the Agent Connection Tab ... 17

About the Exchange Connection Tab ... 18

Processing Certificate Errors ... 19

Updating Objects and Operations ... 19

Step Configuration ... **20**

About the General Tab 21
About the Connection Tab 23
About the Mapping Tab 24
Creating Mappings 24
Refreshing the Exchange Schema 25
Supported Objects 25
About the Options Tab 27
Processing Incomplete Inputs 27

Introduction

Use components in the **Microsoft Exchange Tool Pack** to send and retrieve data from Microsoft Exchange, including:

- Email messages
- Calendar meetings and appointments
- Contacts
- Task items

You can also use the tool pack to create user-defined folder structures.

All this allows for data to be synchronised between Exchange and third-party systems that it does not normally connect with.

The Microsoft Exchange Tool Pack

The tool pack consists of:

- **Microsoft Exchange Event Agent** — This **Agent** subscribes to events occurring in Exchange. When an event occurs, the **Event Agent** is notified allowing BPA Platform to act accordingly. The **Event Agent** can be installed on any computer that has access to both the BPA Platform server and Exchange, local to the BPA Platform server, or if available, on the on-premise server hosting Exchange. For more information, see *About the Microsoft Exchange Event Agent*.
- **Microsoft Exchange Connector Agent** — This **Agent** communicates directly with the Exchange Web Service (EWS). It can be installed on any computer that has access to both the BPA Platform server and Exchange, local to the BPA Platform server, or if available, on the on-premise server hosting Exchange. For more information, see *About the Microsoft Exchange Connector Agent*.
- **Microsoft Exchange Connector** — The **Connector** communicates directly with the **Connector Agent** instead of directly interacting with Exchange. It must be installed on the BPA Platform server and any remote BPA Platform client computers which run **Microsoft Exchange Connector** tasks.

System Requirements

The following prerequisite software must be in place before installing the **Microsoft Exchange Tool Pack**:

Minimum BPA Platform Software

The **Microsoft Exchange Tool Pack** requires BPA Platform (formerly known as "TaskCentre") version 2020 or above.

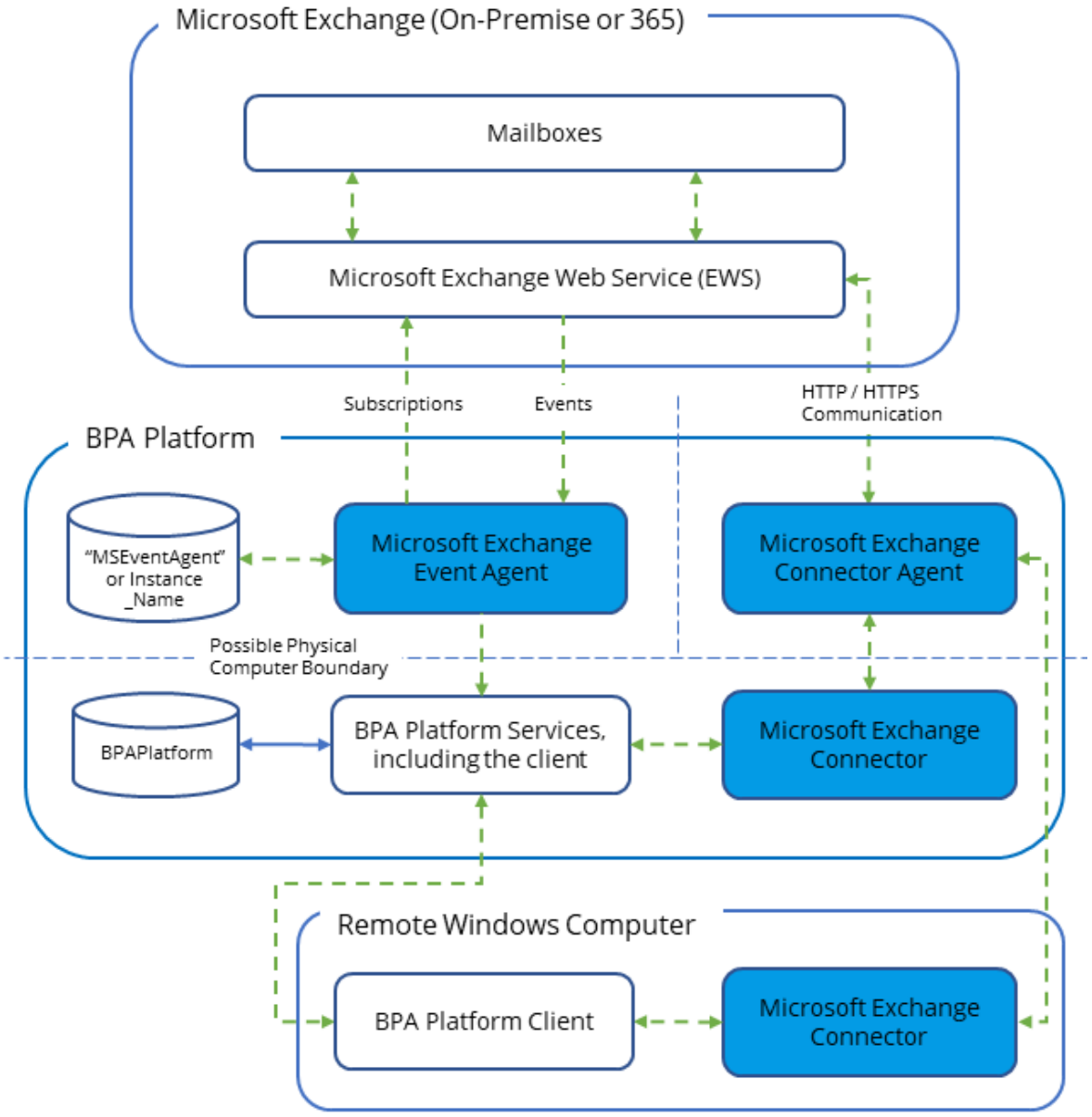
Minimum Microsoft Exchange Version

This tool pack is compatible with:

- Exchange 2013 — on-premise or hosted (both local and remote are supported)
- Exchange 2016 — on-premise or hosted (both local and remote are supported)
- Exchange 2019 — on-premise or hosted (both local and remote are supported)
- Exchange 365 / Exchange Online — referred to as Exchange 365 in this document

Architecture

The diagram below provides a high-level system architecture overview of the **Microsoft Exchange Tool Pack** with BPA Platform and the EWS.



Whether Exchange is locally or remotely hosted, or supplied as part of Office 365, the connection between Exchange and BPA Platform is via the EWS.

The **Microsoft Exchange Event Agent** must be registered with the BPA Platform server. The only other communication the **Event Agent** has with BPA Platform is to log error messages in the Event Log (***Tasks*** toolbar **>** ***Event Log***) — hence the one-way arrow in the above diagram; all configuration for the **Event Agent** must be done through the **Microsoft Exchange Event Agent Configuration** wizard. To access the events in the **Event Agent's** database, you must use the **Database Query (ODBC)**, **Database Query (OLEDB)**, or **Microsoft SQL Server Trigger** tools, refer to the product help. The **Event Agent** requires access to a SQL Server instance to store and

manage the mailbox events — this can be local to the **Event Agent** installation or on a remote computer that the **Event Agent** can access. Should your organisation make use of multiple Exchange instances, for example, a local on-premise installation as well as using Exchange 365, only a single installation of the **Event Agent** is required but additional **Event Agent** services must be configured — see *About the Microsoft Exchange Event Agent Instance Manager*.

The **Microsoft Exchange Connector Agent** is the **Connector's** interface to the EWS. It only requires access to Exchange and the computer hosting the **Microsoft Exchange Connector** (if not installed locally).

Both **Agents** can be installed local to the BPA Platform server or together on a remote Windows computer that has access to Exchange, BPA Platform, and SQL Server. Separate remote installation of the **Agents** is also supported so long as access to necessary components is available.

The **Microsoft Exchange Connector** must be installed on the BPA Platform server and any remote installations of the BPA Platform client where **Microsoft Exchange Connector** tasks are required.

Working with Other Tools

The **Microsoft Exchange Connector** can directly interact with the following tools:

Consuming from Other Tools

The **Microsoft Exchange Connector** can consume output from the following tools:

Icon	Tool Name	Tool Category
	Import XML Document	Input
	Retrieve Text Message	Input
	Convert Recordset to XML	Format
	Transform Data	Format
	Call Task	Execute
	Applications Platform Connector	Data Connectors
	Microsoft Exchange Connector	Data Connectors

In addition, the **Microsoft Exchange Connector** can consume the output from other Data Connector tools that provide connectivity to an external application, such as an ERP or CRM system.

Objects Consumed

The **Microsoft Exchange Connector** consumes the following objects exposed by other steps:

- **XML** — XML data from any BPA Platform tool capable of exposing such data (see above)

Exposing to Other Tools

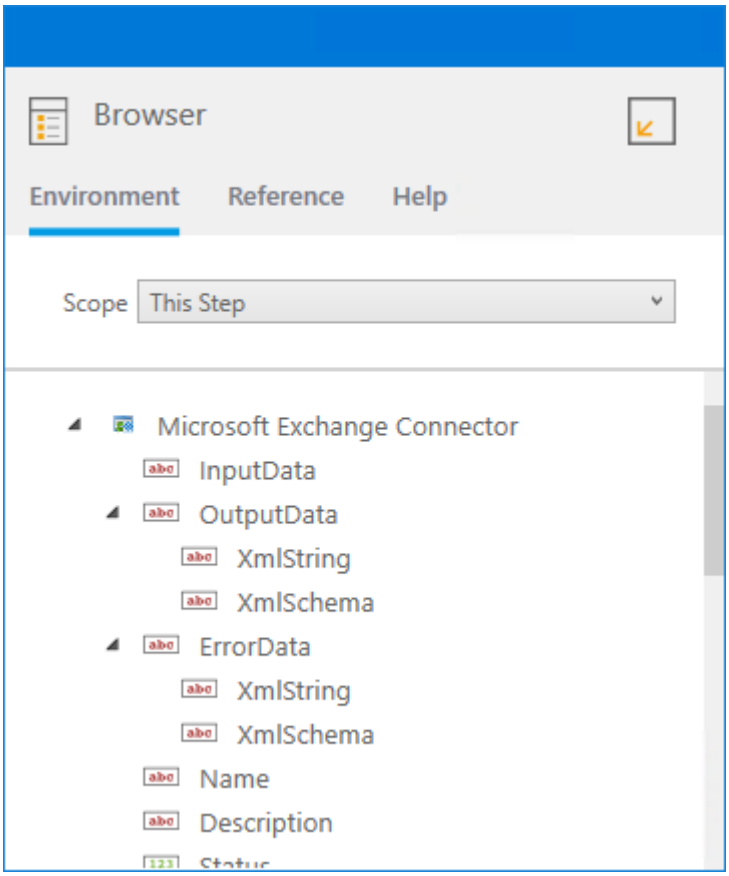
The following tools can directly consume output form the **Microsoft Exchange Connector**:

Icon	Tool Name	Tool Category
	Retrieve Text Message	Input
	Convert XML to Recordset	Format
	Run Microsoft Reporting Services	Format
	Transform Data	Format
	Save File	Output
	Call Task	Execute
	Applications Platform Connector	Data Connectors
	Web Service Connector	Data Connectors
	Microsoft Exchange Connector	Data Connectors

In addition, other Data Connector tools that provide connectivity to an external application, such as an ERP or CRM system, can also consume the output from the **Microsoft Exchange Connector**.

Objects Exposed

The **Microsoft Exchange Connector** tool outputs the following objects which can be consumed by other tools:



- **InputData** — This document contains the input XML received by the **Microsoft Exchange Connector** tool. It is only available if a task step has been selected as the **Data Source** (see *About the General Tab*).
- **OutputData** — The **OutputData** object contains two sub-objects:
  - **XmlString** — This is the XML document produced by the tool, containing data returned from Exchange for all operations. Also included are the key fields for the mapped elements affected by the used operation and a `SupplementaryReference` field for task auditing purposes.

    The mapped fields in the **Mapping** tab (see *About the Mapping Tab*) define the structure of this XML document.
  - **XmlSchema** — This contains the output schema in XSD format.
- **ErrorData** — The **ErrorData** object also contains two sub-objects:
  - **XmlString** — This contains any error data reported by Exchange

    `<ErrorData>`

    `<Exchange_ObjectItem />` — The mapped object, for example, **ContactItem**

`<mapped_fields />` — All the fields that were mapped with their values that were sent to Exchange, plus all data contained in `SupplementaryReference`

`<ResponseCode />` — The error code received back from Exchange

`<ErrorDetails />` — The error text received back from Exchange

`</ErrorData>`

For more information about the errors received, see *Error Handling*.

- **XmlSchema** — This contains the output schema in XSD format.

- **Step Properties** — Standard step properties are also available allowing you to use statistical data of the **Microsoft Exchange Connector** step.

Where Can the XML Output be Used?

The incoming XML is translated into the XML format for the object and operation selected in the configuration. The data for the linked fields is brought across into the output XML — only those fields that are linked are brought across. The XML is passed to the **Connector**, which then:

- Processes the data
- Performs the operation requested
- Receives an XML document containing the response

Both the `OutputData` and `ErrorData` documents can be directly used by succeeding task steps that can consume XML data, as part of an application integration or synchronisation process. To use the documents in a non-XML consuming tool, use a **Convert XML to Recordset** or **Transform Data** step first to create a recordset copy of the XML data.

The XML documents are also available as consumable objects from the Task Browser (`XmlString`). When used in a task step, such as **Format as Text** or **Save File**, this exposes the actual XML string.

Error Handling

Errors are written to the BPA Platform Event Log (***Tasks*** toolbar **> *Event Log***). You define how errors are handled in the **Options** tab of the tool (see *About the Options Tab*).

Reasons for the errors could include:

- Web service connection errors
- User privilege errors
- Errors, messages, and warnings from the Microsoft Exchange API

- Any reported task runtime errors, including **Connector or Event Agent** errors, such as, loss of connection

About the Microsoft Exchange Event Agent

Use the **Microsoft Exchange Event Agent** to create tasks based on activity in specified mailboxes in Exchange. This activity includes:

- An object or folder being copied in the mailbox
- An object or folder being created in the mailbox
- An object or folder being deleted in the mailbox
- An object or folder being modified in the mailbox
- An object or folder being moved in the mailbox

where *object* relates to the part of the mailbox being monitored, for example, the inbox or calendar, and *folder* is an Exchange folder ID number for a custom mailbox folder.

When installed, the **Microsoft Exchange Event Agent** creates its own SQL Server database (named `MSEventAgent` by default, unless additional instances are created) where it records events that Exchange has notified it about. Stored procedures are available that allow you to:

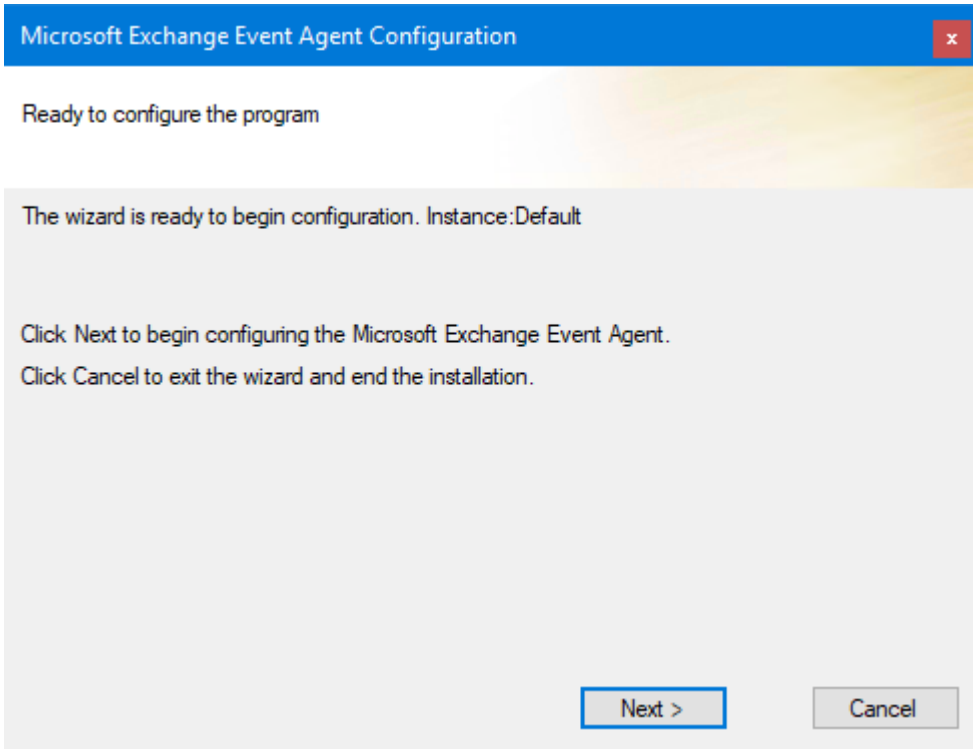
- Specify the mailbox to monitor (`Subscribe`)
- Update the processed flag for record (`UpdateNotifications`)
- Remove processed records from the database (`RemoveProcessedRows`)
- No longer monitor the mailbox (`Unsubscribe`)

You can configure the **Event Agent** to monitor more than one instance of Exchange without the need to install multiple **Event Agents**. For a detailed description of how to do this, see *About the Microsoft Exchange Event Agent Instance Manager*.

For a detailed description of the stored procedures and runtime processing for the **Microsoft Exchange Event Agent**, refer to the *Microsoft Exchange Tool Pack Data Dictionary*.

Using the Microsoft Exchange Event Agent Configuration

Immediately after installation, the Microsoft Exchange Event Agent Configuration wizard is launched. (For a detailed description of how to install the **Microsoft Exchange Event Agent**, refer to the *Microsoft Exchange Tool Pack Quick Start Guide*.

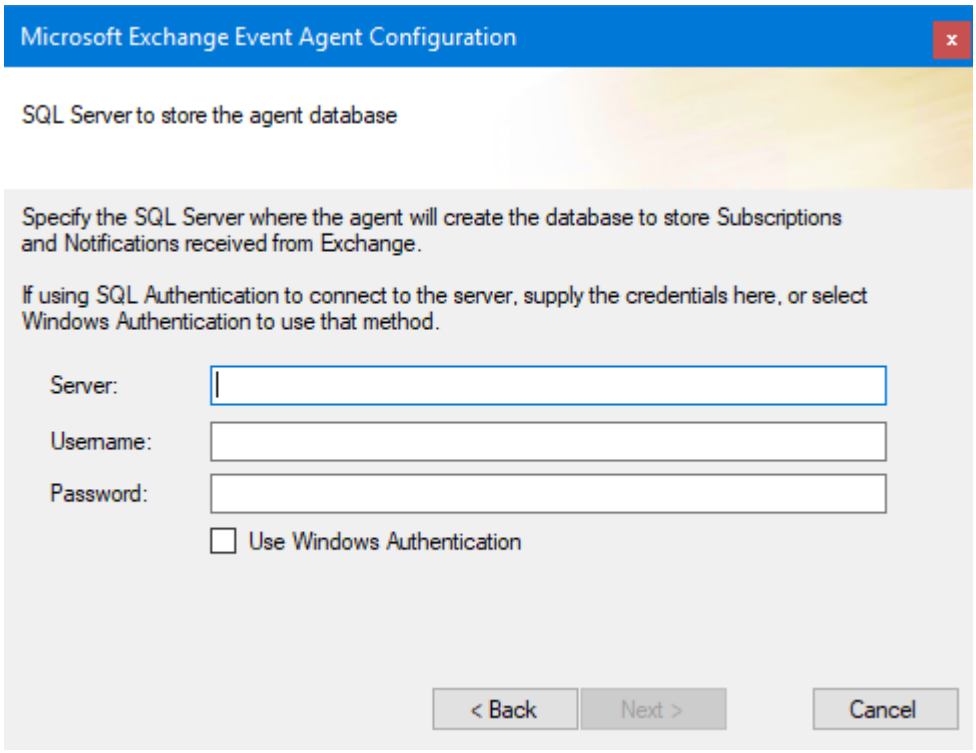


This wizard steps through creating the connection from the **Microsoft Exchange Event Agent** to the Exchange Web Service (EWS) and the BPA Platform server.

> **TIP:** If you want to cancel out of the wizard and return to it at a later time, you can re-launch the wizard from the Microsoft Exchange Event Agent Instance Manager — see *About the Microsoft Exchange Event Agent Instance Manager*.

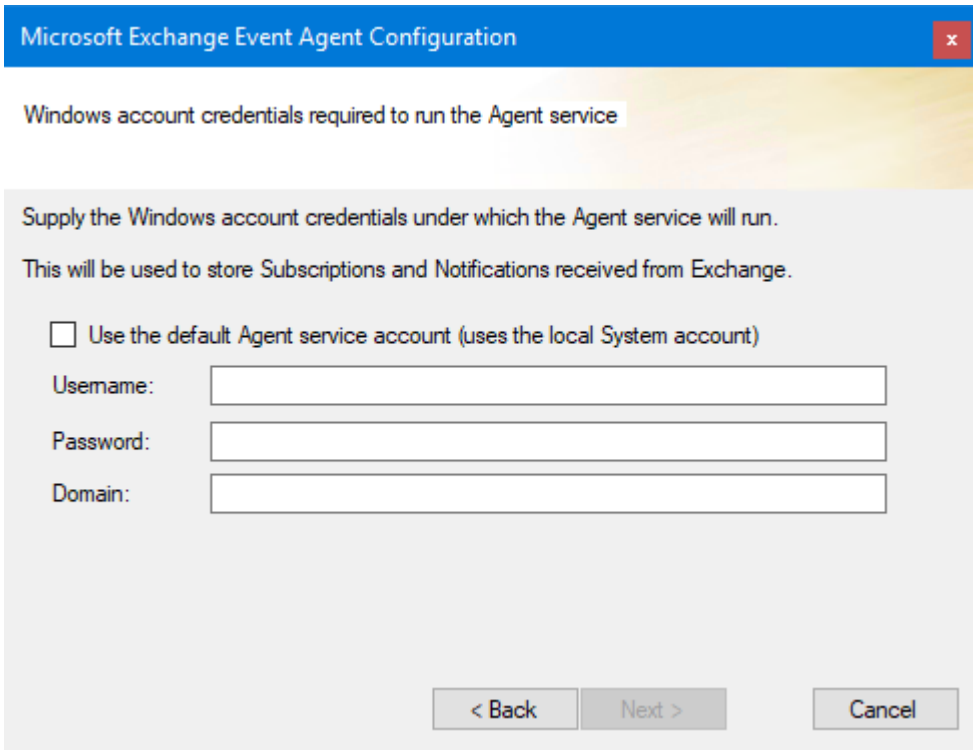
Adding the SQL Server Connection

The **Microsoft Exchange Event Agent** requires access to an SQL Server instance to store event notifications from Exchange.



Adding the Windows Service User Account

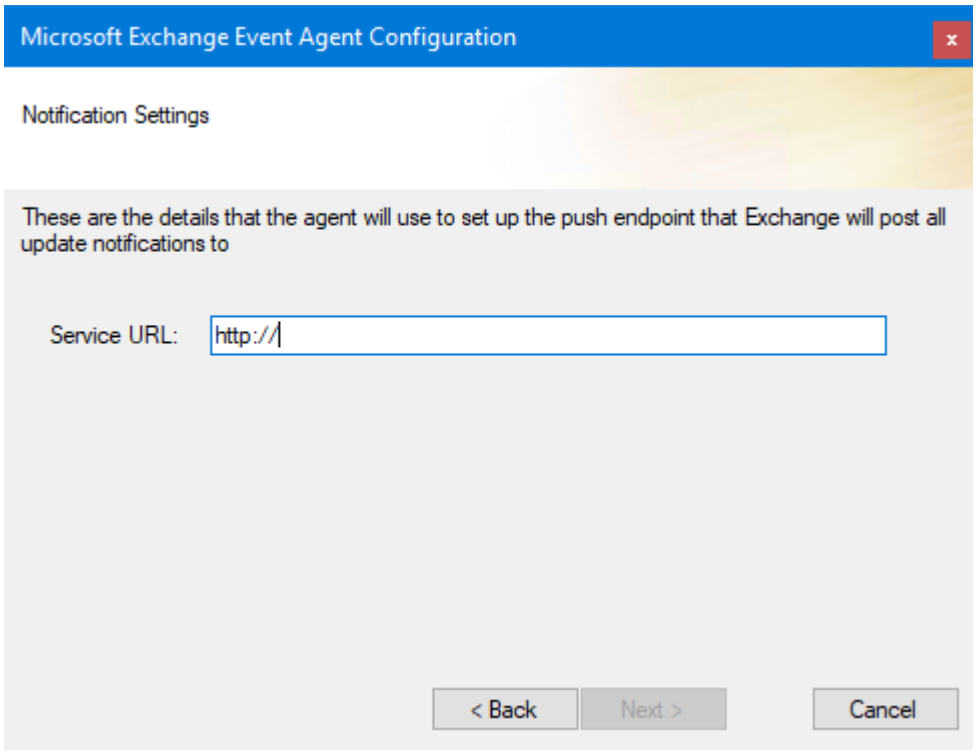
The **Microsoft Exchange Event Agent**, as a Windows service, requires a Windows user account to run under. This user account must have permissions to add, update, and remove records from the **Event Agent** database.



Typically, the Windows Local System account which the **Event Agent** services runs as has the relevant permissions to read and write to the **Event Agent** database (**Use the default Agent service account**).

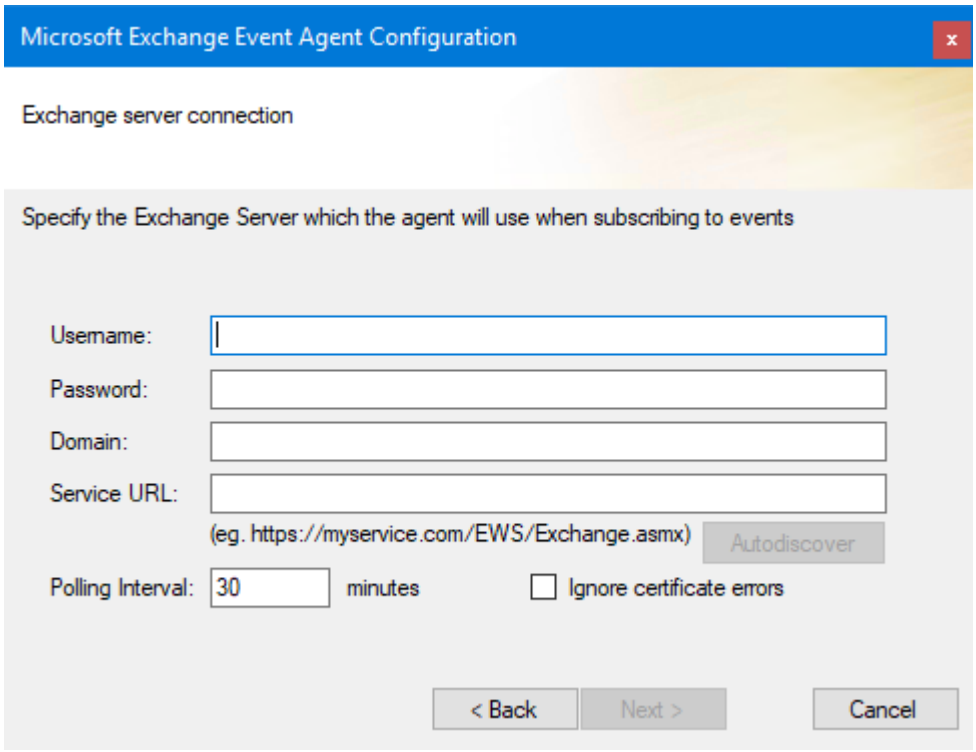
Adding the Push Endpoint

When Exchange has accepted the subscription requests, it pushes all event notifications to a nominated endpoint.



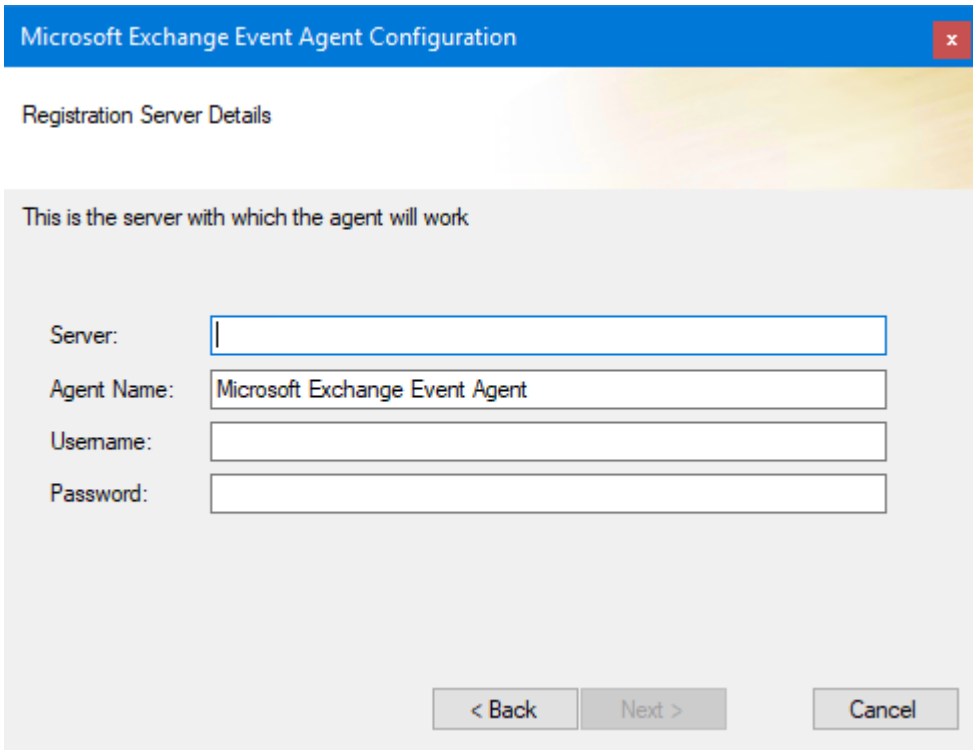
Adding the Exchange Web Service Connection

You must supply the Exchange server details this **Microsoft Exchange Event Agent** connects to.



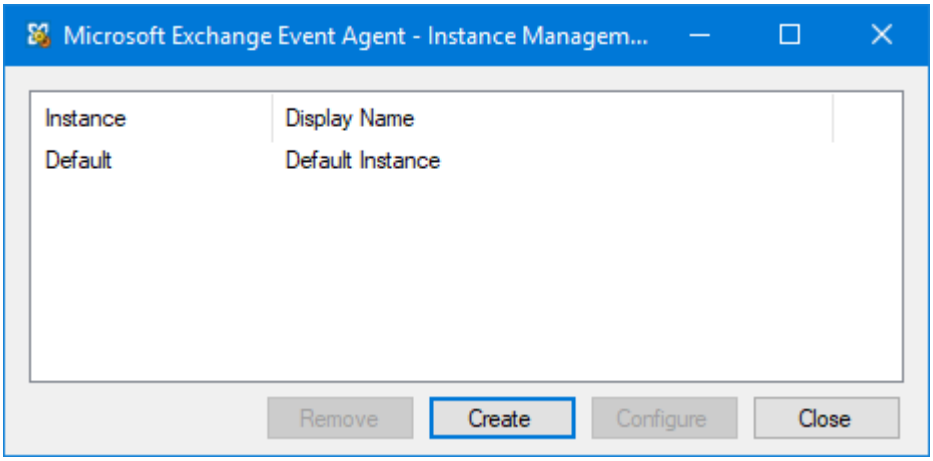
Adding the BPA Platform Connection

The final stage of the Microsoft Exchange Event Agent Configuration wizard is to register the **Event Agent** with the BPA Platform server.



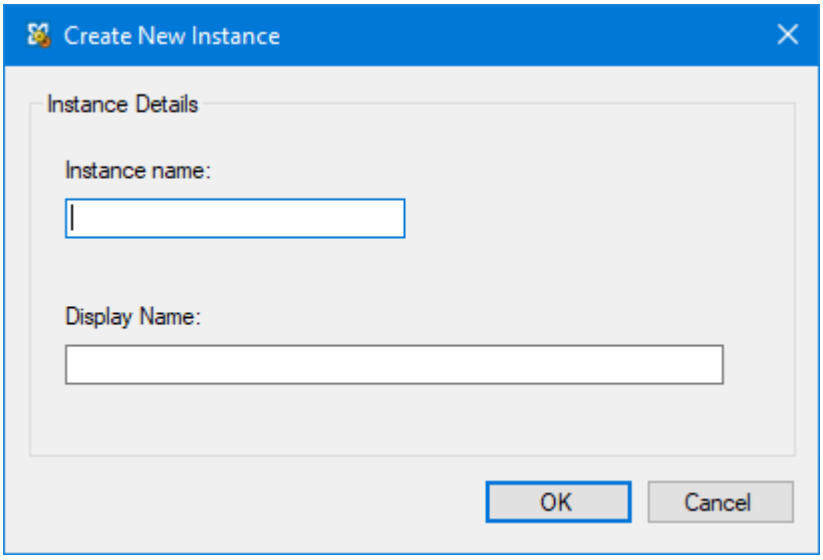
About the Microsoft Exchange Event Agent Instance Manager

If you previously cancelled out of the Microsoft Exchange Event Agent Configuration wizard without completing the configuration, you can re-launch it from the Microsoft Exchange Event Agent Instance Manager. This is located in the BPA Platform menu group in the Windows Start menu.

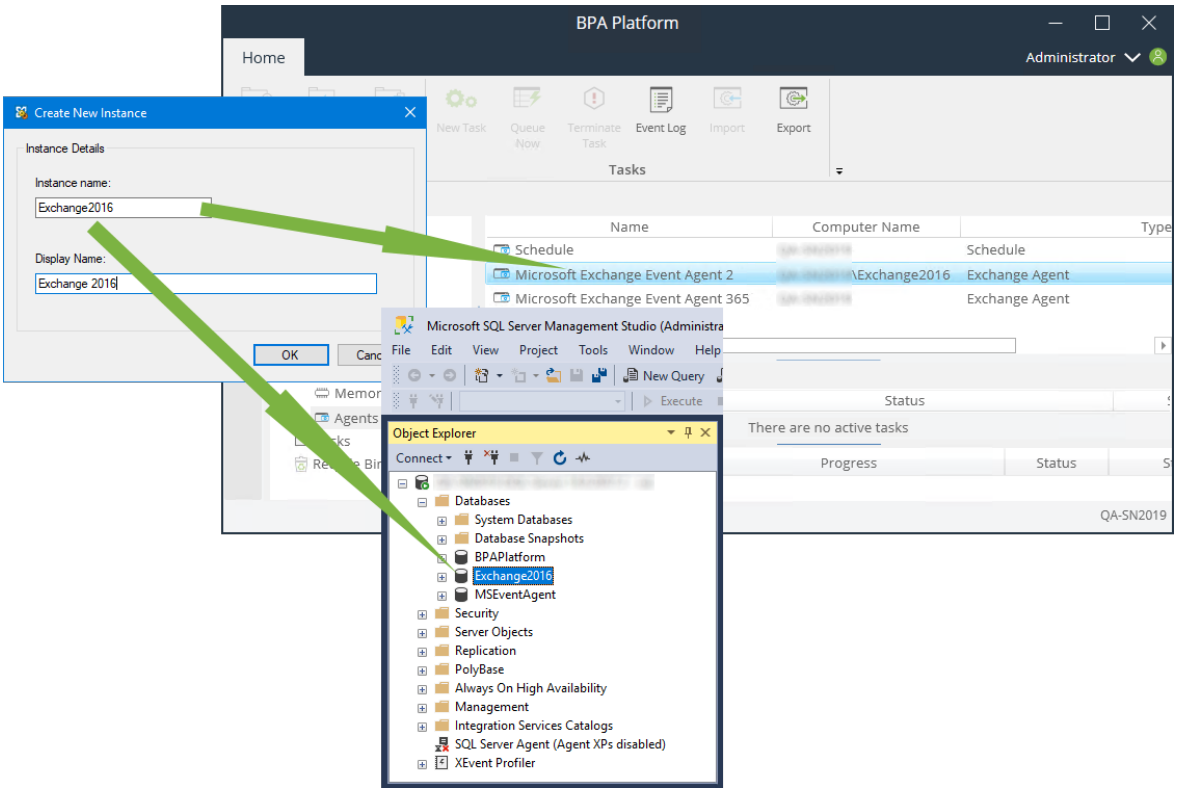


Unlike the **Connector Agent** (see *About the Microsoft Exchange Connector Agent*) where it can only connect to a single Exchange instance, you can also have the same **Event Agent** subscribe to events from different Exchange instances — for example, if your organisation makes use of an on-premise Exchange server as well as Exchange 365.

The **Default Instance** shown above is the one you create immediately after the **Event Agent** has finished installing. To create additional instances, click **Create**.



Enter a unique and meaningful **Instance Name**; note that this is the name given to the database used to store Exchange events and so is used to form the **Computer Name** when the **Event Agent** is registered with the BPA Platform server:



The Microsoft Exchange Event Agent Configuration wizard is launched after this window — see *Using the Microsoft Exchange Event Agent Configuration*.

Use **Configure** to change any of the **Event Agent** connection parameters — note that you cannot change the **Instance name** once the database has been created.

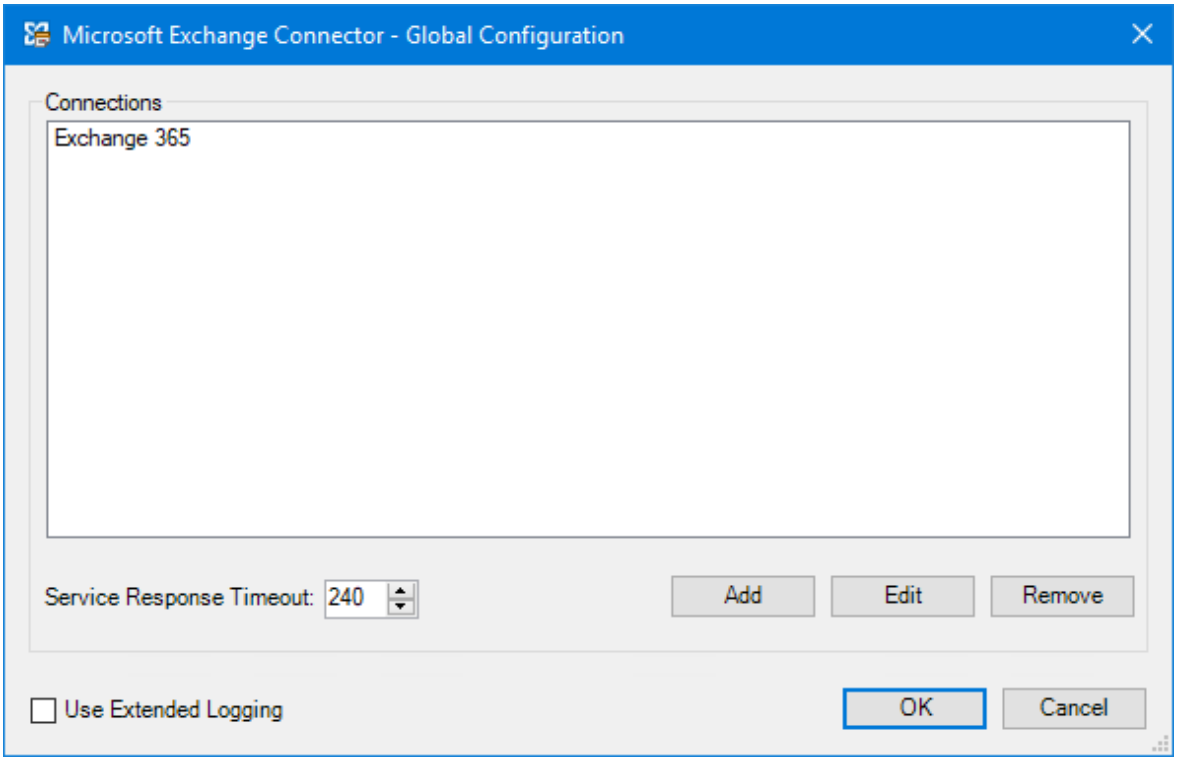
About the Microsoft Exchange Connector Agent

To communicate with the Microsoft Exchange Web Service (EWS), you must install the **Microsoft Exchange Connector Agent**. This acts as a "go-between", updating and retrieving data for use in **Microsoft Exchange Connector** tasks. The **Agent** can be installed local to the BPA Platform server, on a remote computer that has access to both the BPA Platform server and your Exchange instance, or if available, local to the on-premise Exchange server.

Adding a Microsoft Exchange Connection

The global configuration for the **Microsoft Exchange Connector** is used to create connections to the **Microsoft Exchange Connector Agent** and the Exchange web service.

You open this interface from the resources tree — expand ***System > Tools > Data Connectors*** and double-click ***Microsoft Exchange Connector*** in the items list.



When a connection is created, the available objects and fields of your Exchange installation are retrieved.

Click **Add** to create a connection to Exchange.

Setting the Time Out

You can control how long the **Microsoft Exchange Connector** waits for a response from Exchange before setting the connection to an "unconnected" status.

The default time out is `240` seconds. Typically, this a sufficient length of time required for a response from the Exchange Web Service, but you can adjust the **Service Response Timeout** parameter to suit your organisational needs.

Using Extended Logging

Selecting this option exposes the full XML parsed between the **Microsoft Exchange Connector** and Microsoft Exchange.

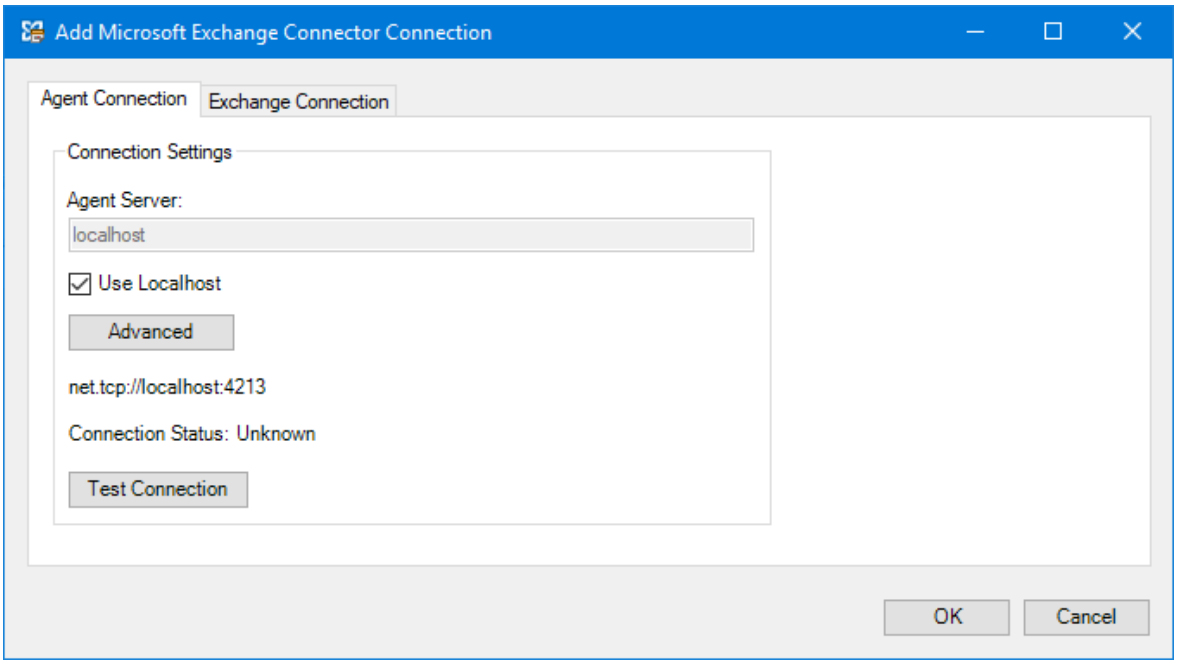
Without extended logging, the Event Log only contains start and end of transaction messages, plus any error messages encountered at runtime.

You can view the extended log in the BPA Platform Event Log (***Tasks*** toolbar **> *Event Log***).

> **TIP:** The Event Log database table may grow substantially large as extended logging adds additional rows for each call made to and from the **Connector**. To manage this, adjust the maintenance routine for the Event Log — refer to the BPA Platform product help.

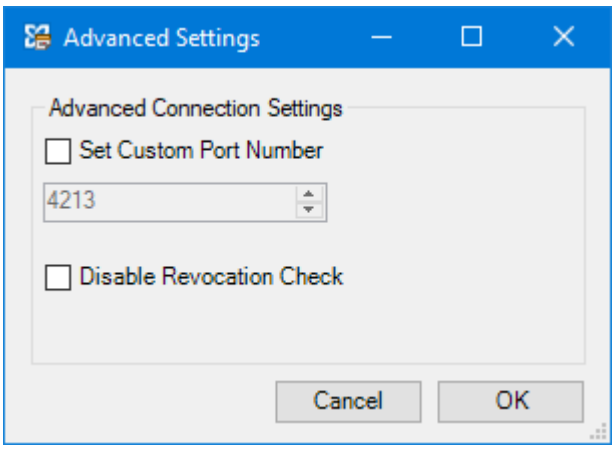
About the Agent Connection Tab

Use this tab to configure the connection to the **Microsoft Exchange Connector Agent** (the **Microsoft Exchange Event Agent** is configured separately — see *Using the Microsoft Exchange Event Agent Configuration*).



Enter the **Agent Server** (hostname or IP address) hosting the **Connector Agent** installation. If it is installed local to the BPA Platform server, click **Use Localhost** instead.

By default, the **Connector Agent** to **Connector** communication is over port `4213`. If the **Connector Agent** has been configured to use a different port, click **Advanced** and select **Set Custom Port Number**. Enter the new port in the box provided.



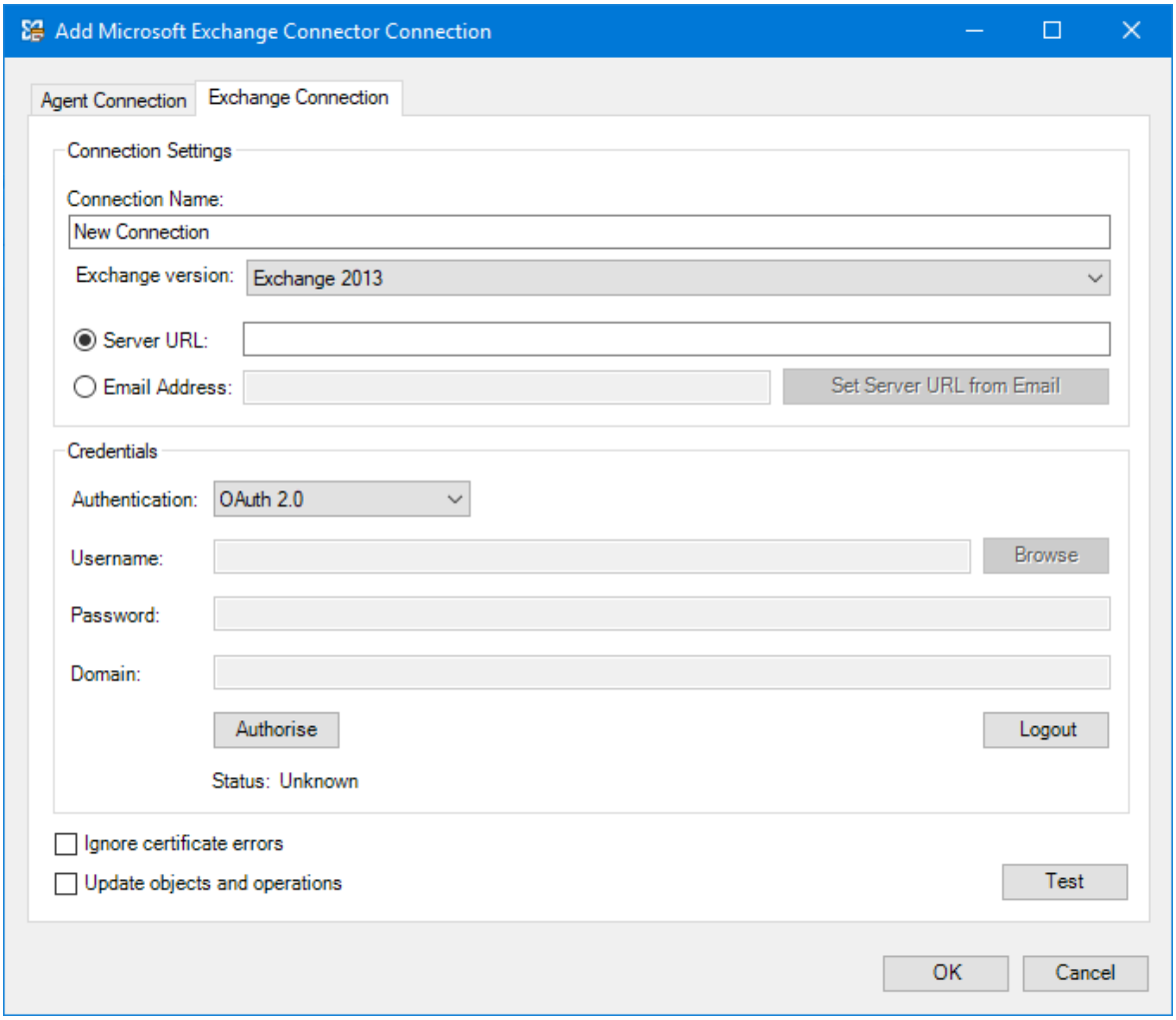
If using self-signed SSL certificates to validate the connection between the **Connector Agent** and the EWS, you can select **Disable Revocation Check** to prevent checking the certificate against the Certificate Authority's revocation list.

Use **Test Connection** to ensure the **Agent** can be reached successfully.

About the Exchange Connection Tab

Use the **Exchange Connection** tab to configure how the **Microsoft Exchange Connector Agent** connects to the Exchange Web Service.

> **NOTE:** The **Microsoft Exchange Connector Agent** uses an Exchange account to access the email server. It is recommended you create a dedicated account assigned to the **Agent** to for traceability purposes and to ensure access does not interfere with other users.



Configure a meaningful **Connection Name** for this **Agent** connection; this is the **Connection** name used when creating **Microsoft Exchange Connector** task steps.

From the **Exchange version** drop-down, choose the relevant version you are running. If connecting to Office 365, select **Exchange Online**.

If the URL to the Exchange service, `exchange.asmx`, is known, enter it into **Server URL**, using the following format:

`https://hostname_or_ip_address_or_url/ews/exchange.asmx`

where *`hostname_or_ip_address_or_url`* is the hostname or IP address of the on-premise Exchange server, or the URL of the Exchange 365 instance.

If the URL is not known, it can be determined from a valid email address. To do this, select **Email Address** and enter the assigned email into the box provided, enter the **OAUTH2.0** account **Credentials** in the pane below, and click **Set Server URL from Email**.

Whether the **Server URL** is known or you're determining the URL from an **Email Address**, the **Agent** requires Exchange credentials. Choose the relevant **Authentication** method from the drop-down:

- **OAUTH2.0** — If connecting to Exchange Online, use this authentication type. Click **Authorise** and enter the account details in the Microsoft Authorisation Console.
- **Transparent Windows** — Use the credentials for the currently logged in Windows account.

  To use this option, the Windows account must also:

  - Have permission access the Exchange server
  - Must be an administrator (granted delegation permissions in Exchange)

  If connecting to an on-premise Exchange server, the BPA Platform server must be in the same Active Directory domain.
- **Basic** — Enter the **Username**, **Password**, and (optional) **Domain** of the **Microsoft Exchange Connector Agent's** account. Use Browse to search for the **Username** in Active Directory.

Processing Certificate Errors

To ignore any untrusted certificate errors that may occur when connecting to Exchange, select **Ignore certificate errors**. To "self-certify", leave this option blank and register the Exchange server certificate on the BPA Platform server — for more information, see our knowledge base article, *How to register a third party server certificate for use on a BPA Platform server*.

Updating Objects and Operations

Enabling **Update Objects and Operations** allows the Microsoft Exchange schema in BPA Platform to be refreshed every time an update is made in Exchange itself.

Should you also upgrade Exchange to a newer, compatible version, all related schemas within BPA Platform must be refreshed. To do this, select this option and save the changes. You must repeat this for each existing connection you have defined.

NOTE: After enabling **Update Objects and Operations**, you must refresh the schema in each **Microsoft Exchange Connector** task step. You do this in the **Mapping** tab (**Schema Refresh**) — see *About the Mapping Tab*.

Step Configuration

When creating new tasks, the **Microsoft Exchange Connector** is located under **Data Connectors** of the Task Browser.

NOTE: It is not possible to add the **Connector** to the task until at least one connection to Exchange has been defined in the global configuration — ensure at least one connection exists in the *global configuration*.

To add a new **Microsoft Exchange Connector** step to an existing task, do the following:

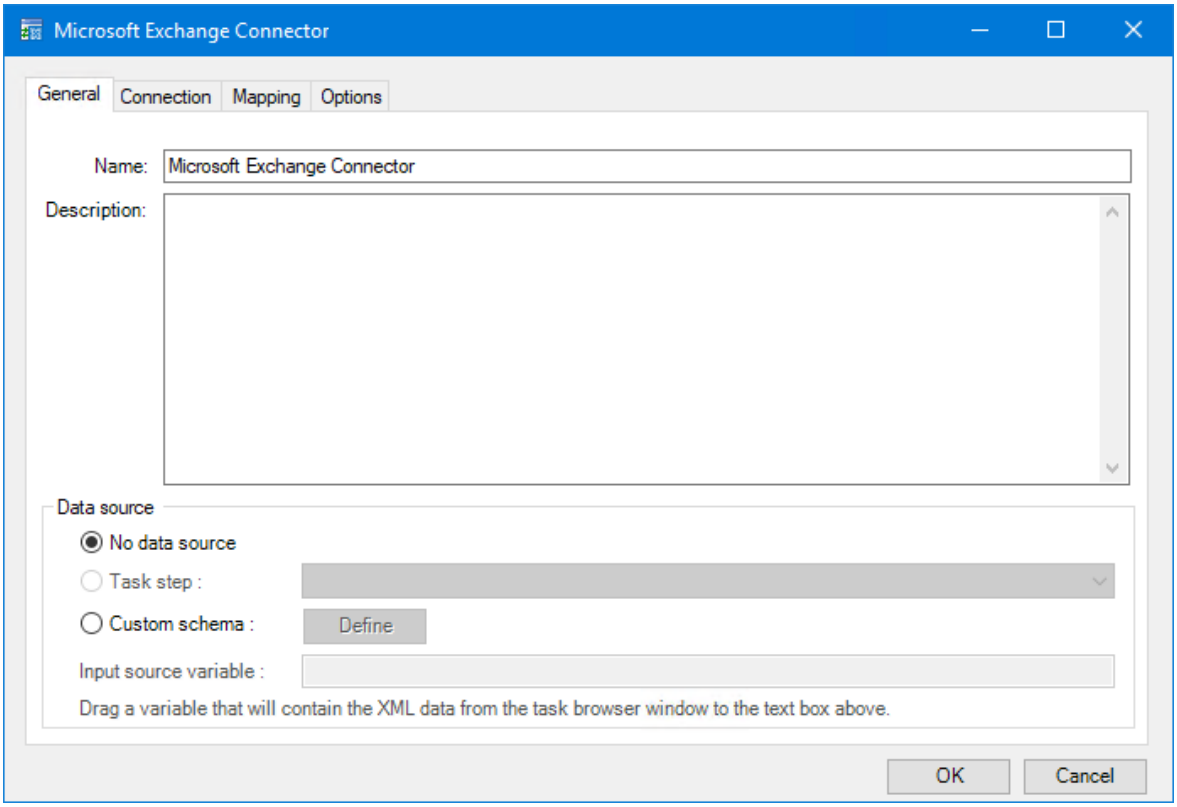
From the relevant task, either:

- Click and drag the **Microsoft Exchange Connector** icon from the **Task Browser** to the task **Design** area.
- From the task's **Design** tab, right-click on empty space and select ***New > Data Connectors > Microsoft Exchange Connector***.

For a detailed description of how to create new tasks, refer to the product help.

About the General Tab

Use the **General** tab to choose the BPA Platform data source to be mapped to the Microsoft Exchange objects. The data source must be in XML form. To map BPA Platform recordsets, configure a **Convert Recordset to XML** or **Transform Data** step in the task before the **Microsoft Exchange Connector** step, then select this as **Task step** (see below).



Provide a meaningful **Name** and **Description** for this step.

> **TIP:** If this task instance makes use of two or more **Microsoft Exchange Connector** steps, ensure the **Name** used is unique for each individual step.

Data source can either be:

- **No data source** — If you don't make use of a dedicated XML input source, select this option to use BPA Platform variables in place of the XML objects' fields.

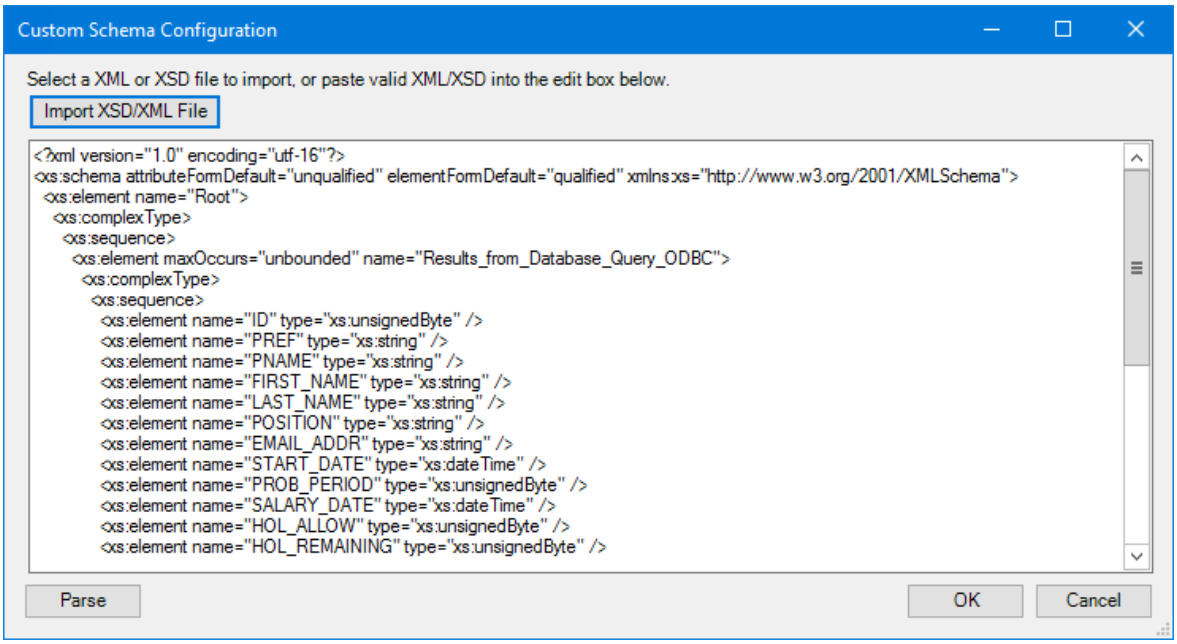
  For example, you can extract various bits of information from an email and store them in BPA Platform variables. The variables are then mapped to Microsoft Exchangeoperational objects and fields — see *About the Mapping Tab*.

- **Task step** — The data source can be set to an available BPA Platform XML data source.

Only those steps that are capable of natively exposing an XML document at runtime are listed. This may be another **Microsoft Exchange Connector** step, or a tool such as **Convert Recordset to XML** or **Transform Data**.

- **Custom schema** — An XML schema defines the structure of the parsed XML: what tags are present, and the nesting of the tags. You **Define** the schema of the XML that is used as the input data source for this step. The **Microsoft Exchange Connector** tool uses the industry standard XSD format. Any XML processed by this step must conform to this schema else an error will be reported.

  If the XSD schema is available, either import it into the **Custom Schema Configuration** (use the **Import XSD/XML File** button), or copy and paste it into the configuration box.

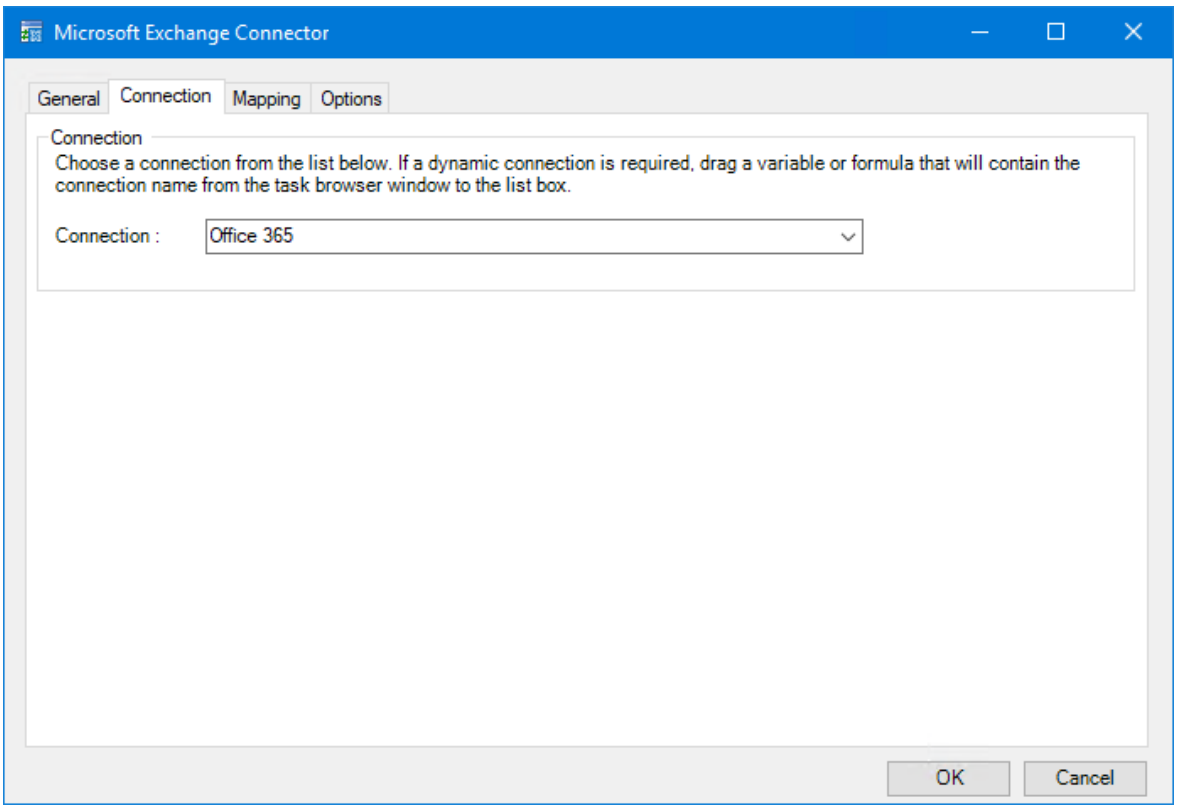


If the XSD schema is not available, you can import an example of the runtime XML (**Import XSD/XML File**), or copy and paste it into the configuration box. Use the **Parse** button to create the schema.

- **Input source variable** — As well as defining the schema, specify the BPA Platform variable that contains the XML data at runtime.

About the Connection Tab

To map task data to Microsoft Exchange, you must specify with Exchange connection to use.

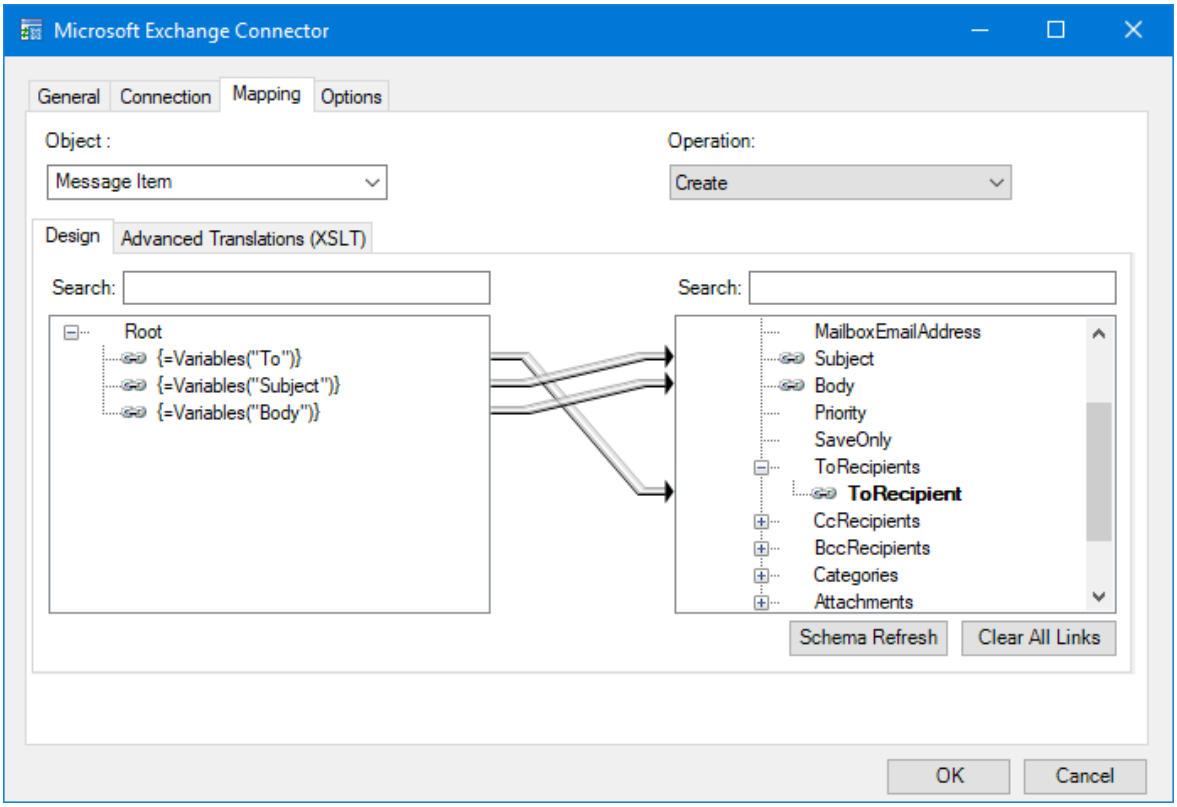


All connections created in *global configuration* are presented here.

Alternatively, you can use a BPA Platform variable or formula to create a dynamic connection, where the connection used is determined by runtime circumstances. At runtime, the contents of the variable must match the name of one of the global connections — this is case-sensitive. To do this, you must install an **Microsoft Exchange Connector Agent** for every Exchange server you want to connect to.

About the Mapping Tab

Use the **Mapping** tab to define links between the data source XML (see *About the General Tab*) and those required by Exchange. This defines how, at runtime, the incoming XML is to be translated into the XML required for the relevant object and operation.



The **Object** drop-down shows the available Microsoft Exchange objects. The **Operations** drop-down shows the operations available for the selected **Object**. For more information about supported objects and operations, see *Supported Objects*.

Further down, the left-hand XML tree structure shows those data source fields available for mapping (see *About the General Tab*). The right-hand XML tree structure displays those input fields for the selected **Object** and **Operation**.

> **TIP:** If you have selected **No data source** in the **General** tab, you must add BPA Platform formulas and variables to the left-hand data source XML tree structure before mapping.

Creating Mappings

Create links by dragging and dropping a data source field (left) onto its corresponding **Microsoft Exchange Connector** input field (right). Only linked fields are used in the output XML.

BPA Platform formulas and variables can be included in the source data even when using a **Task step** or **Custom schema** (see *About the General Tab*) — drag them from the Task Browser to the data source's XML; these can then be

linked to the relevant input fields. BPA Platform recordsets must first be converted to XML using either the **Convert Recordset to XML** or **Transform Data** tool before they can be used here.

Each operation has an additional field, `SupplementaryReference`, which allows for traceability when transferring data from one place to another. When mapped, the data resides locally at runtime. It is added to the output, and creates a record for reference purposes only — you can choose to map any field to `SupplementaryReference` to assist with checking where the data originated from or at what time the data transfer occurred, for example.

The **Microsoft Exchange Connector** tool uses eXtensible Stylesheet Language Transformations (XSLT) to translate the received XML. The **Advanced Translations (XSLT)** tab shows the XSLT generated for the links created for the object and operation. Use **Enable Free Type Mode** to directly edit the XSLT — this is particularly useful when translating a non-standard requirement.

Refreshing the Exchange Schema

If new fields, objects, and operations have been added to your Exchange instance after this task step was originally created, use **Schema Refresh** to make the latest API metadata available for use (ensure **Update Objects and Operations** is selected first in the relevant global connection).

Supported Objects

The **Microsoft Exchange Connector** makes the following objects and operations available for mapping, with fields that must be mapped highlighted:

> **NOTE:** The following information is referenced from Microsoft Exchange 365 and Exchange 2019. If using a different version of Exchange, you may find objects and operations differ slightly.

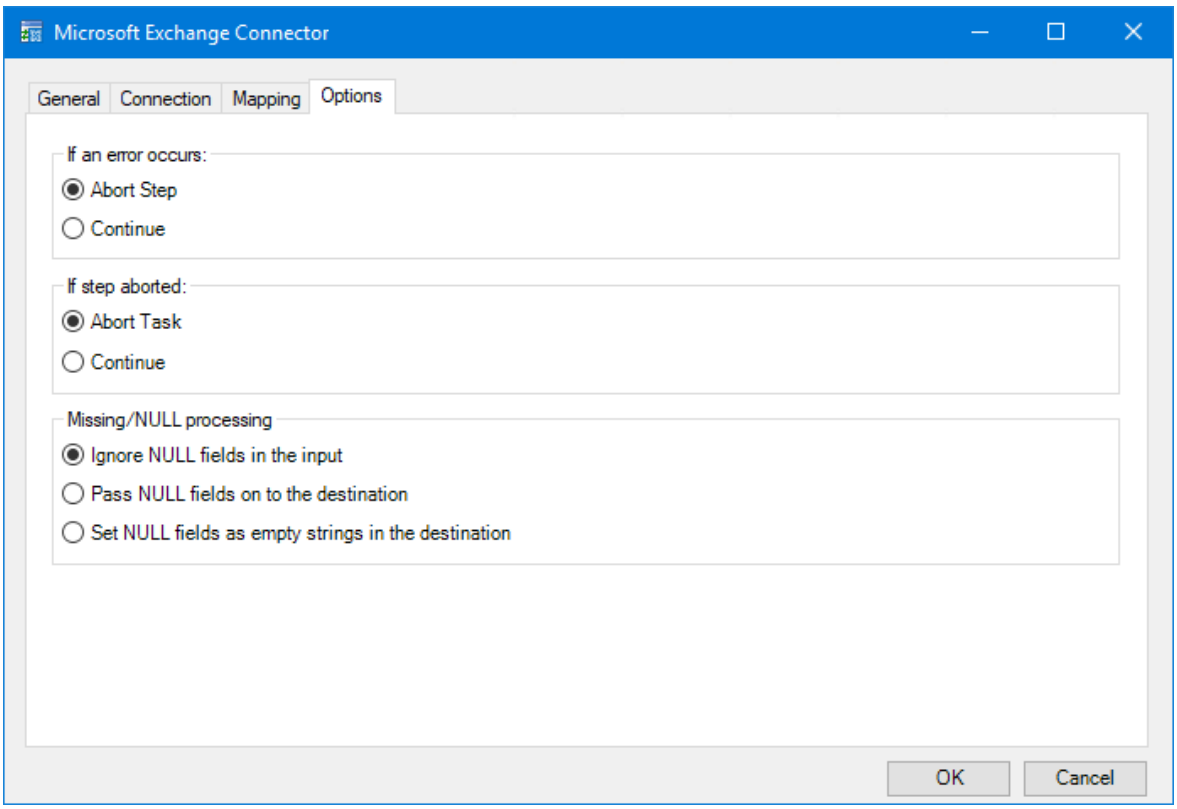
Object	Supported Operations					Notes
	CREATE	DELETE	FIND	GET	UPDATE	
Calendar item	✔ **Subject** **Body**	✔ **Exchange-CalendarId**	✔ **Search-Field** **Search-Expression**	✔ **Exchange-CalendarId**	✔ **Exchange-CalendarId**	Additional supported operations: ▸ `FIND OCCURANCES` **SearchField** **SearchExpression** **StartDate** **EndDate** ▸ `FIND OCCURANCES BY MASTER` **ExchangeCalendarId** **StartDate** **EndDate**

Object	Supported Operations					Notes
	CREATE	DELETE	FIND	GET	UPDATE	
Contact Item	✔ No mandatory fields for the CREATE operation.	✔ **Exchange-ContactId**	✔ **Search-Field** **Search-Expression**	✔ **Exchange-ContactId**	✔ **Exchange-ContactId**	
Task Item	✔ No mandatory fields for the CREATE operation.	✔ **Exchange-TaskId**	✔ **Search-Field** **Search-Expression**	✔ **Exchange-TaskId**	✔ **Exchange-TaskId**	
Message Item	✔ **ToRecipient**	✔ **Exchange-MessageId**	✔ **Search-Field** **Search-Expression**	✔ **Exchange-MessageId**	✔ **Exchange-MessageId**	Additional supported operations: ▸ FORWARD **ExchangeMessageId** **ToRecipient** ▸ MOVE **ExchangeMessageId** ▸ REPLY **ExchangeMessageId**
Folder Item	✔ **Name**	✔ **Exchange-FolderId**	✔ **Search-Field** **Search-Expression**	✔ No mandatory fields for the GET operation.	✔ **Exchange-FolderId**	

For a detailed description of the available fields for each object and operation combination, refer to the *Microsoft Exchange Tool Pack Data Dictionary*.

About the Options Tab

The **Options** tab allows you to define how errors in this step are handled at task runtime.



If an error occurs, you can decide whether the step should **Continue** processing, or terminate the step immediately (**Abort Step**).

If the step is aborted, you can choose to **Continue** processing onto the next step in the task, or terminate the whole task immediately (**Abort Task**). By allowing the task to **Continue**, you can use the error XML received back in a **Save File** step for investigation purposes, for example.

Processing Incomplete Inputs

You can also control how the **Connector** processes missing input or empty input elements for mapped fields — a missing input is where a field is mapped in the **Mapping** tab (see *About the Mapping Tab*) but not received in the input (see *About the General Tab*), whereas an empty element is where ***<element_name></element_name>*** or ***<element/>*** is received in the input. The following behaviours are available:

- **Ignore NULL fields in the input** (default selection) — Missing fields or those received without a value are not passed to Exchange even though they are mapped (and therefore required)
- **Pass NULL fields on to the destination** — This option appends `xsi:nil="true" xmlns:xsi="http://w3.org/2001/XMLSchema-instance` to the element: ***<element_name* xsi:nil="true" xmlns:xsi="http://w3.org/2001/XMLSchema-instance></*element_name*>**

- **Set NULL fields as empty strings in the destination** — Missing fields or those received without a value are passed without values: ***<element_name></element_name>***

Want to learn more?

Discover how Codeless Platforms can help your business by improving performance, boosting efficiency and cutting costs.

+44 (0) 330 99 88 700

enquiries@codelessplatforms.com

www.codelessplatforms.com

codeless PLATFORMS