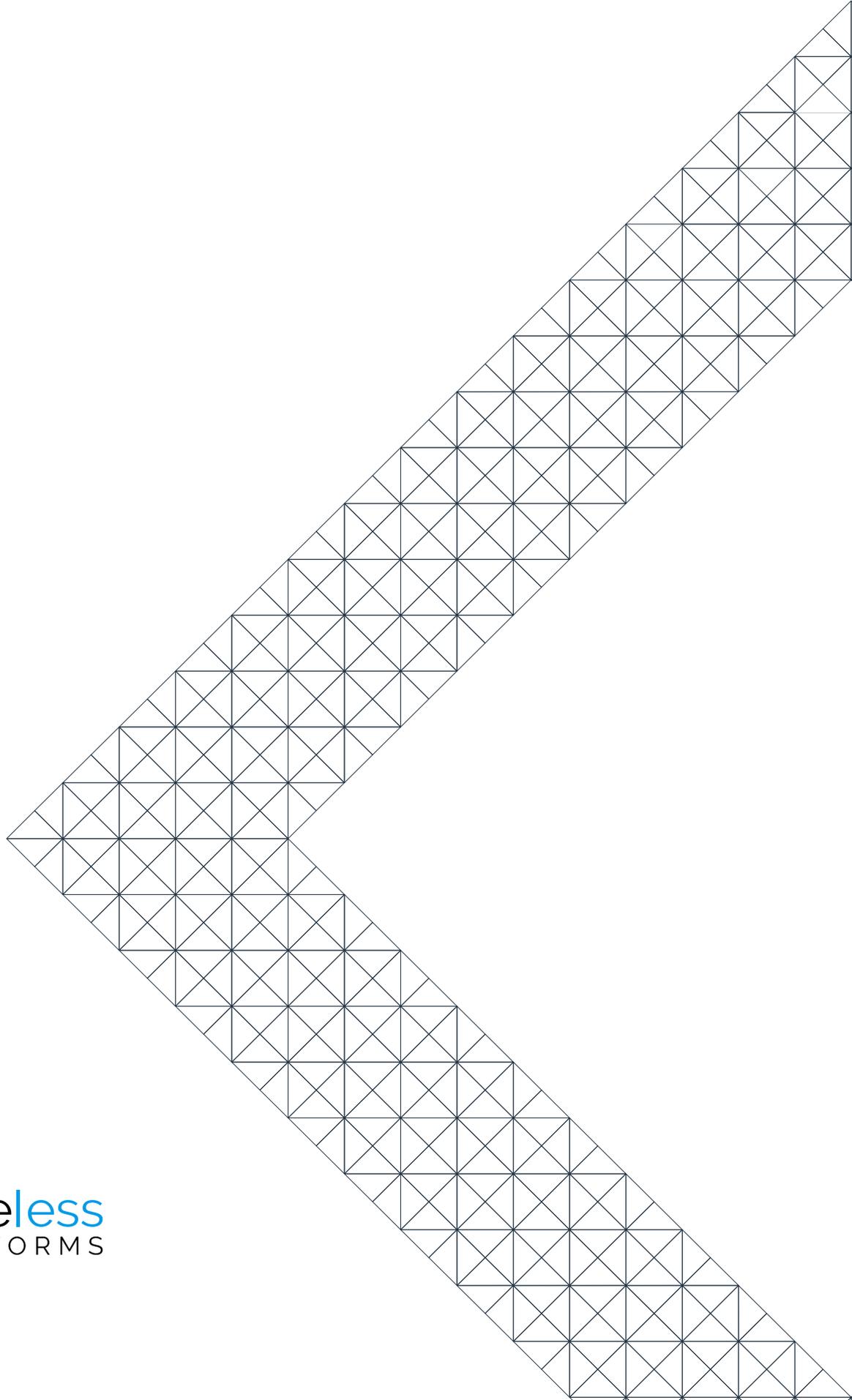


# TaskCentre

## Call Task Tool

### White Paper



# Copyright

The copyright in this document is owned by Orbis Software T/A Codeless Platforms 2017. 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 T/A Codeless Platforms.

## **Head Office:**

Codeless Platforms  
Suite 1 & 2 Bourne Gate  
25 Bourne Valley Road  
Poole  
BH12 1DY  
United Kingdom  
Tel: +44 (0) 330 99 88 700  
Email: [enquiries@codelessplatforms.com](mailto:enquiries@codelessplatforms.com)

# Trademarks

Orbis Software T/A Codeless Platforms owns the registered Trademark 'TaskCentre'.

All other Trademarks used within this documentation 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The information provided in this publication may contain errors, omissions, or typographical errors or may be out of date.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 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 T/A Codeless Platforms in any way except to the extent that it is specifically indicated to be so.

# Contents

<b>Introduction .....</b>	<b>2</b>
Features.....	2
Working with other steps.....	2
Consuming from other steps .....	2
Exposing to other steps .....	3
<b>Global Settings .....</b>	<b>4</b>
Connection Properties .....	5
Connection Tab .....	5
Folder tab.....	6
<b>Step Configuration .....</b>	<b>6</b>
General Tab .....	7
Connection Tab .....	8
Mapping Tab.....	9
Options Tab .....	10
<b>Objects Exposed .....</b>	<b>11</b>
<b>User Security .....</b>	<b>12</b>
<b>Summary.....</b>	<b>12</b>

# Introduction

**Call Task** is an **Execute** tool which is used when one task is required to trigger another task to run. TaskCentre data and variables can be passed between the two tasks at run time if required.

You may wish to implement this tool for design reasons; for example, a large task consisting of multiple steps can be broken down into smaller groups of tasks with the 'master' task triggering the others. This has the benefit of optimising the load on the server due to the server only having to initiate the tasks and steps required by the business process. By comparison, a single large task by default initiates all steps in the task whether they are required or not which can result in extended completion times.

A Call Task step can consume XML data which can then be passed to the task being called at run time. This is achieved by mapping to parameters via a simple drag and drop interface when configuring the tool. This data can be the source of an imported XML file, recordset data converted to XML or data obtained from a web service call.

Additionally, depending on the operation type the output of the called task can also be passed back to the original task.

## Features

- Configure the Call Task API via an easy to use user interface.
- Run another task and optionally wait for it to complete before continuing.
- Call tasks that are present on another TaskCentre server.
- Pass variables and XML data into a Call Task step for use in the called task.
- Retrieve output data back from the called task when run in sync.
- Create a record of a task call for reference purposes.

## Working with other steps

The other steps that a Call Task step will directly interact with to consume information from are listed below. Data from these steps can be passed to the called task.

### ***Consuming from other steps***

Step Icon	Name	Step Type
	<b>Recordset to XML</b>	Format
	<b>XML Document Import</b>	Input
	<b>Flat File Import</b>	Input

Step Icon	Name	Step Type
	<b>Call Task</b>	Execute
	<b>Web Service Connector</b>	Execute

### ***Exposing to other steps***

A **Call Task** step can also directly expose its output for use by the following steps listed below. This is useful when the tool retrieves an output variable back from the called task. The variable data can then be used within the original task.

Step Icon	Name	Step Type
	<b>XML to Recordset</b>	Format
	<b>Save as File</b>	Output
	<b>Text Parser</b> Using the output XML string of the Call Task tool as the source for the Text Parser when dragged from the browser.	General
	<b>Run Microsoft Reporting Services Report</b>	Format
	<b>Web Service Connector Tool</b>	Execute

# Global Settings

The **Call Task** tool provides a global configuration where a connection to a folder in TaskCentre must be defined. Once a connection has been created this will allow a task from the folder to be selected at step level for running.

The global configuration for the tool can be accessed via the toolbar: *Manage > Tools > Execute > Call Task*.

Or, via the TaskCentre system tree within the **Execute** node:

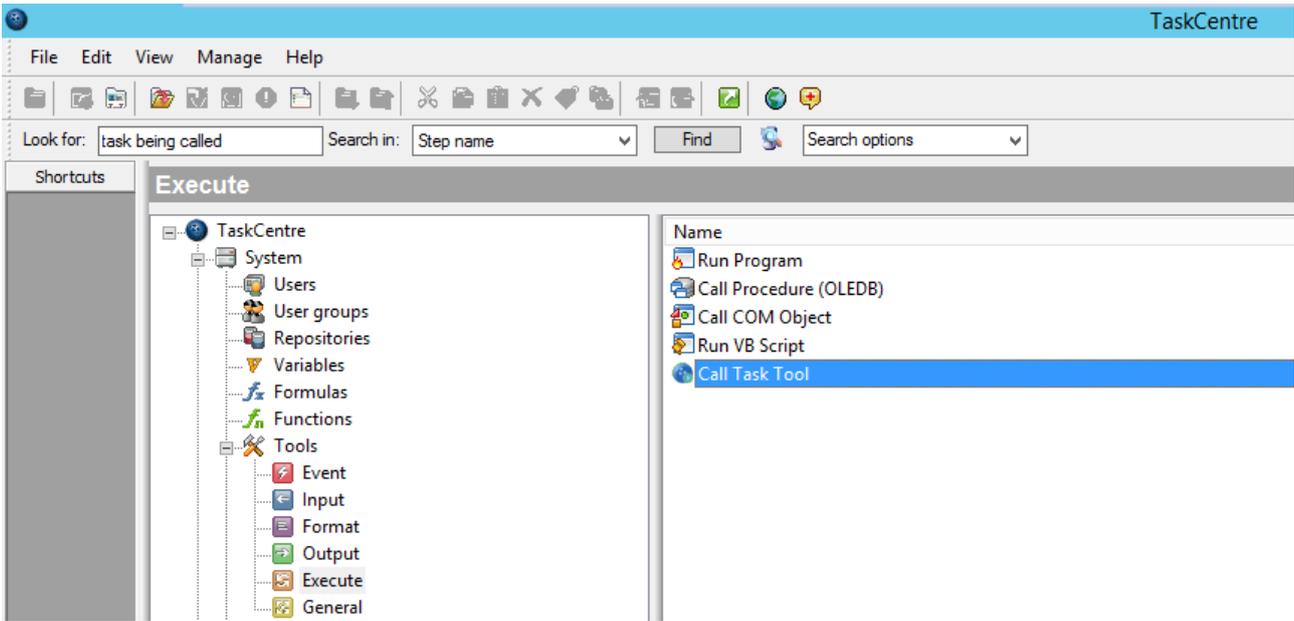


Figure 1. Locating the Call Task Global Settings

The global settings dialog for the tool is used to manage, that is, create, edit, and delete connections to one or more folders containing tasks.

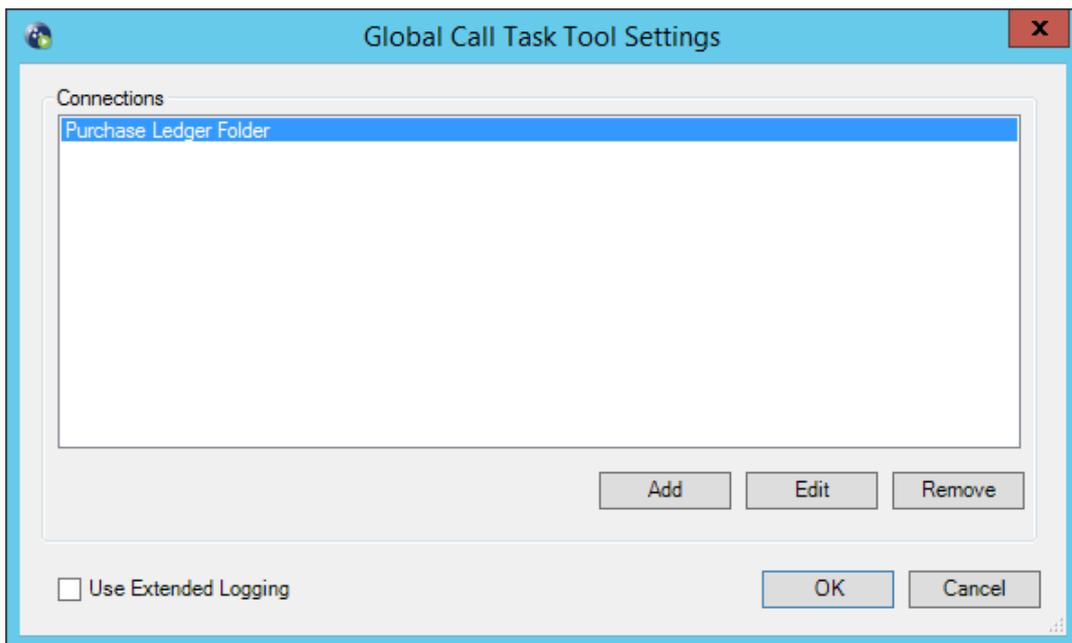
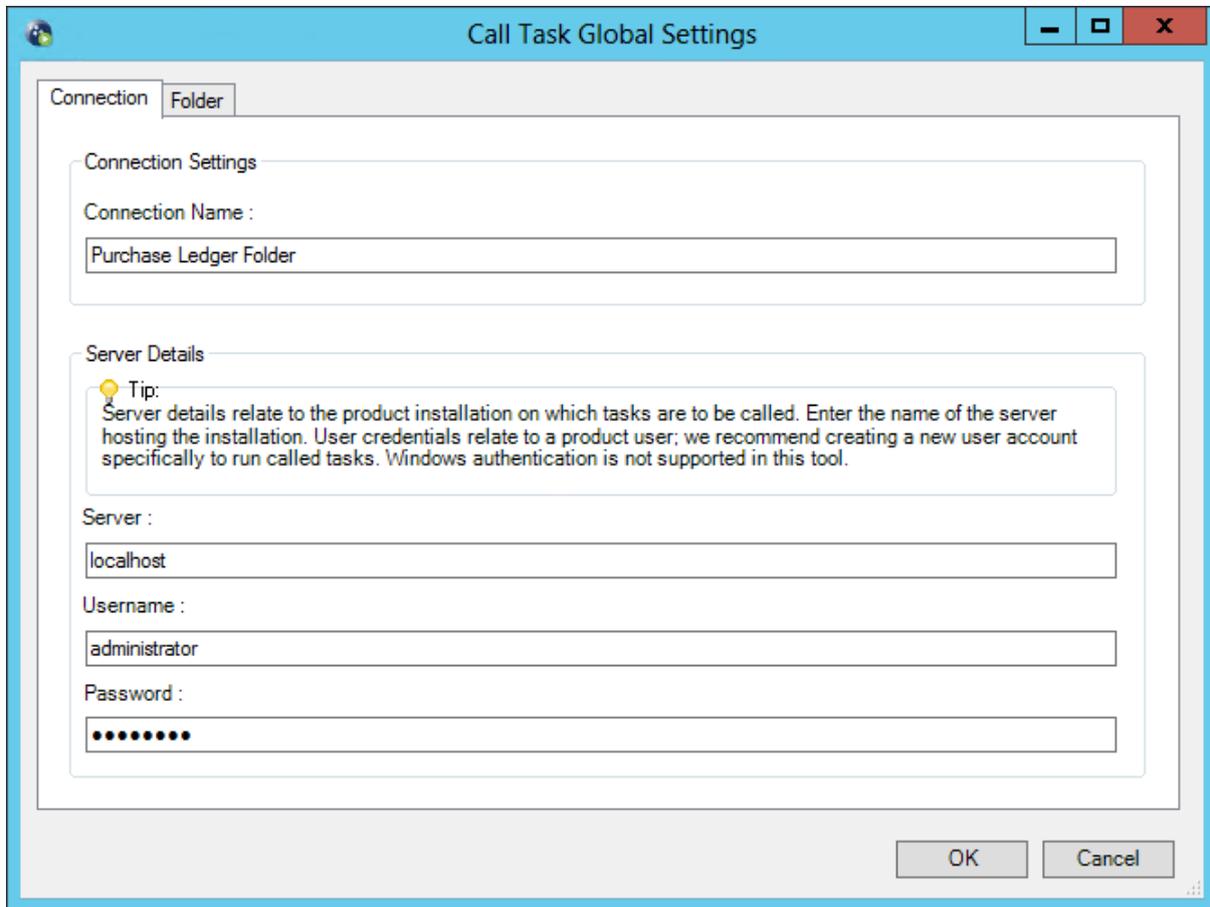


Figure 2. Global Settings dialog — Displaying an example connection

## Connection Properties

The connection properties are displayed when you create or modify an existing connection. Here you specify the folder containing the task that requires calling.

### **Connection Tab**



The screenshot shows a window titled "Call Task Global Settings" with a blue header bar. Inside, there are two tabs: "Connection" (selected) and "Folder". The "Connection Settings" section contains a "Connection Name" field with the text "Purchase Ledger Folder". The "Server Details" section includes a tip box with a lightbulb icon and text: "Tip: Server details relate to the product installation on which tasks are to be called. Enter the name of the server hosting the installation. User credentials relate to a product user; we recommend creating a new user account specifically to run called tasks. Windows authentication is not supported in this tool." Below the tip are three input fields: "Server" with "localhost", "Username" with "administrator", and "Password" with a masked field of ten dots. At the bottom right are "OK" and "Cancel" buttons.

*Figure 3. Call Task Global Settings — Connection tab*

Within the **Connection** tab you provide a name for the connection. The **Server Details** section requires the location of the TaskCentre Server machine to be entered into the **Server** field; this can be either the machine name, IP address or simply localhost.

The **Username** and **Password** fields require the credentials of a TaskCentre account that has sufficient permissions to:

- Use the Call Task tool
- Access the folder and required task within

It is recommended that a new TaskCentre user account is created to specifically run called tasks; this will prevent any login conflicts. Windows accounts are not supported.

## Folder tab

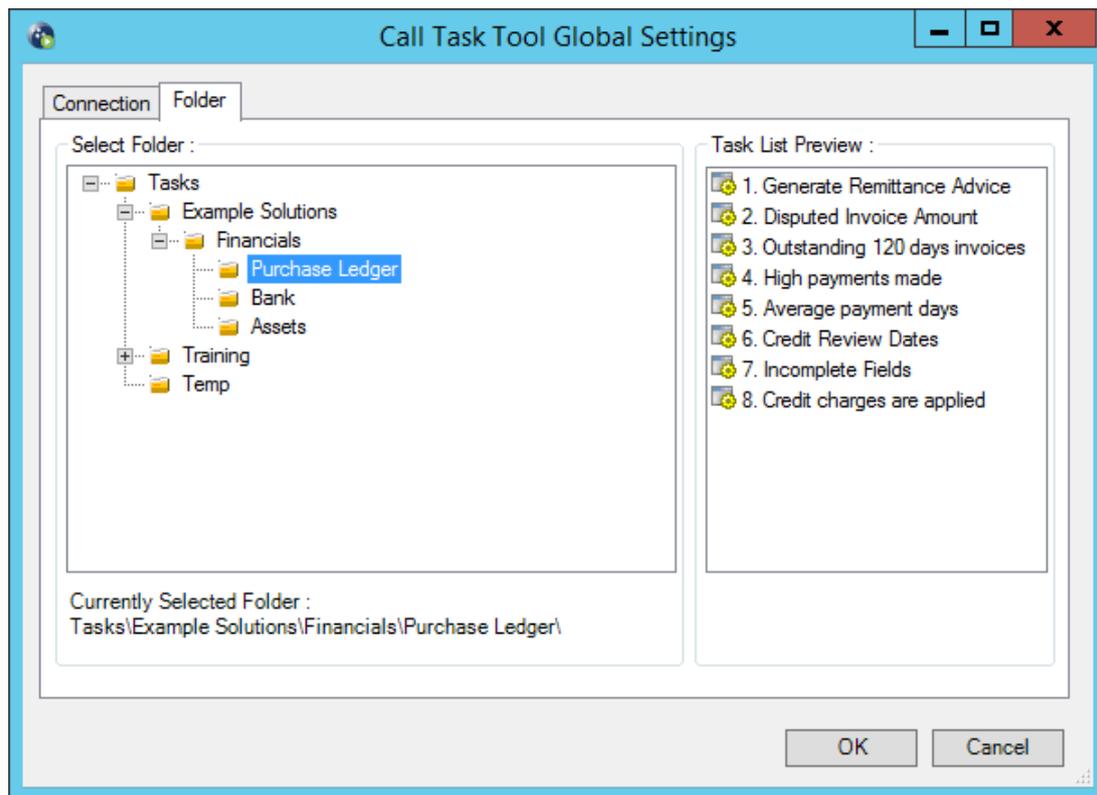


Figure 4. Call Task Global Settings — Folder tab

The **Folder** tab is where you select the folder required by the connection. Only the tasks contained in the selected folder will be exposed at step level. If tasks are required from sub-folders then further global connections will be required.

The right-hand pane displays a preview of the tasks within the folder to ensure the correct one has been selected. Underneath, the path of the folder is also displayed.

## Step Configuration

The **Call Task** tool is dragged from the **Task Browser** under **Execute** tools and into the task design area. Alternatively, you can right-click anywhere in the design area and use the context menu to select the tool.

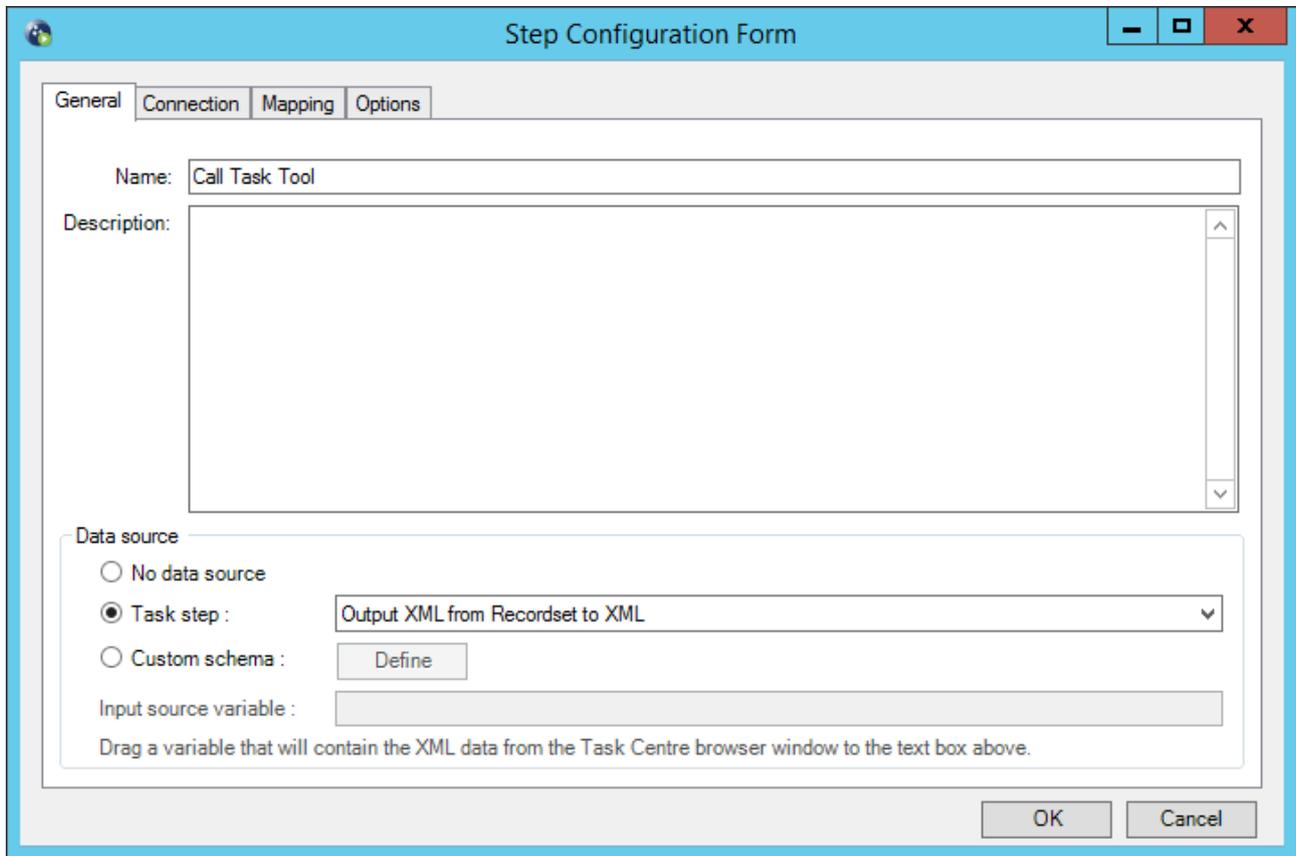
This will automatically open the step for configuration where you will be presented with the following four tabs:

- **GENERAL TAB**
- **CONNECTION TAB**
- **MAPPING TAB**
- **OPTIONS TAB**

**NOTE:** It is not possible to add the tool onto the task designer until at least one connection has been defined in the global configuration.

## General Tab

The **General** tab allows the step to be named and described. XML can also be consumed by the step as an input data source allowing the XML data to be mapped to variables of the task being called.



The screenshot shows a window titled "Step Configuration Form" with a blue header bar. Inside, there are four tabs: "General", "Connection", "Mapping", and "Options". The "General" tab is active. It contains a "Name:" field with the text "Call Task Tool" and a "Description:" field which is empty. Below these is a "Data source" section with three radio button options: "No data source", "Task step:", and "Custom schema:". The "Task step:" option is selected, and a dropdown menu next to it shows "Output XML from Recordset to XML". Below the radio buttons is an "Input source variable:" field which is empty. A small instruction below the field reads: "Drag a variable that will contain the XML data from the Task Centre browser window to the text box above." At the bottom right of the window are "OK" and "Cancel" buttons.

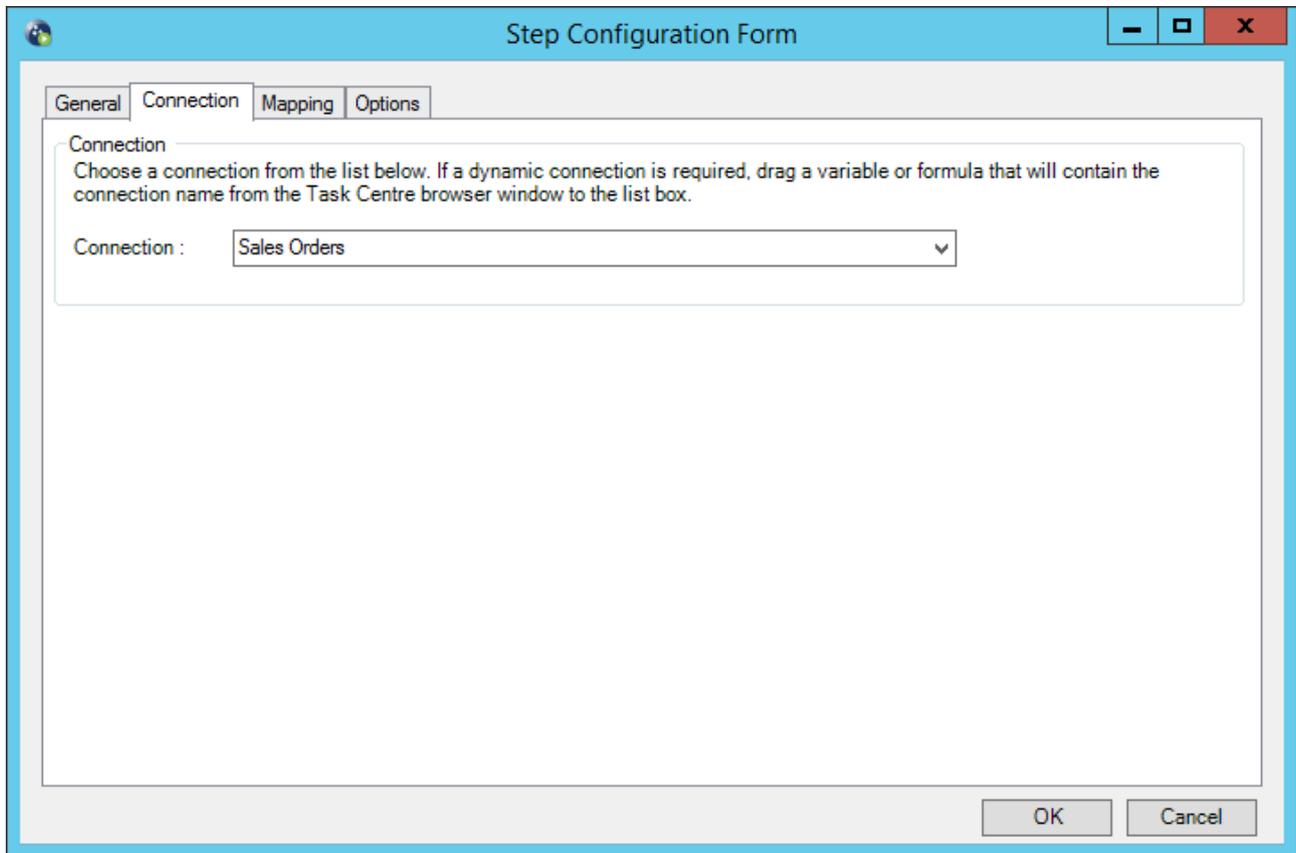
Figure 5. Call Task step — General tab

By selecting the option **Task step**, the XML data can be obtained from another step in the task that outputs XML, such as the **Recordset to XML** tool or **Web Service Connector** tool.

Alternatively, selecting the option **Custom Schema** allows you to import an XML or XSD file from disk via the **Define** button. This is used in conjunction with a task variable that contains XML. The variable would be dragged from the task browser and into the **Input source variable** field.

## Connection Tab

The **Connection** tab allows you to select the folder connection as defined in the tool global configuration. This will allow a task present in the folder to be selected within the **Mapping** tab.



The screenshot shows a window titled "Step Configuration Form" with a blue header bar. Below the header are four tabs: "General", "Connection", "Mapping", and "Options". The "Connection" tab is selected. Inside the tab, there is a section titled "Connection" with the following text: "Choose a connection from the list below. If a dynamic connection is required, drag a variable or formula that will contain the connection name from the Task Centre browser window to the list box." Below this text is a dropdown menu labeled "Connection :" with "Sales Orders" selected. At the bottom right of the window are two buttons: "OK" and "Cancel".

Figure 6. Call Task step — Connection tab

## Mapping Tab

The **Mapping** tab is used to select the task to be called and is where data can be selected to be passed between the tasks at run time.

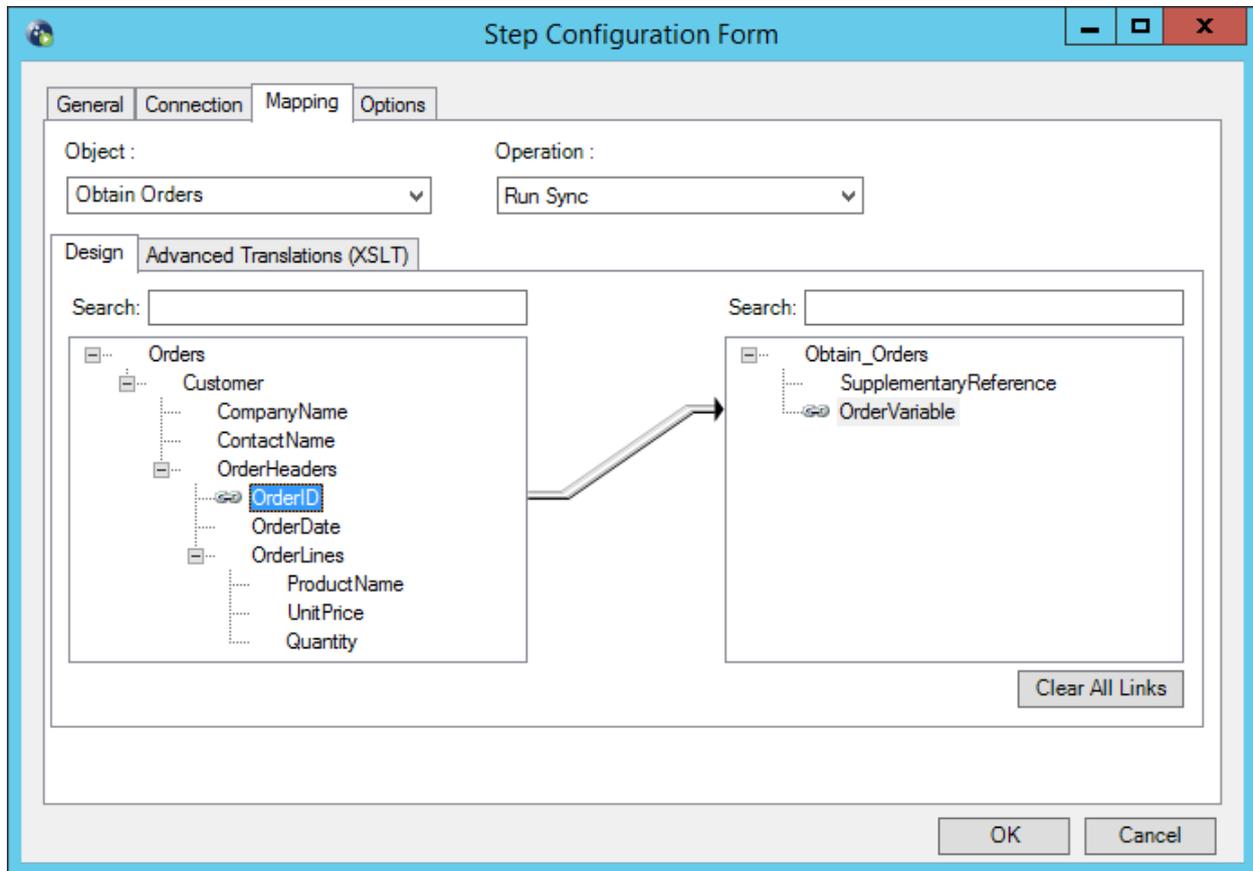


Figure 7. Call Task step — Mapping tab

The **Object** drop-down list is first used to select the task. The **Operation** drop-down list then allows you select how you want the task to run when called, in or out of sync with the original task.

There are two options available as follows:

- **RunSync** — Runs the second task in sync; the original task will wait for the second task to complete. This allows for output variables in the called task to be retrieved.
- **Queued** — Queues the second task and continues to run the original task to completion; both tasks run independently of each other. This option does not allow you to retrieve output variables from the called task.

Below these options is the **Design** tab where data is mapped to variables of the called task. Data is mapped by dragging a field from the left pane on to a field in the right pane to create a link between the two.

The left-hand pane will display, if previously selected, the input data source showing the XML structure. Within this pane you can also drag & drop task variables from the task browser.

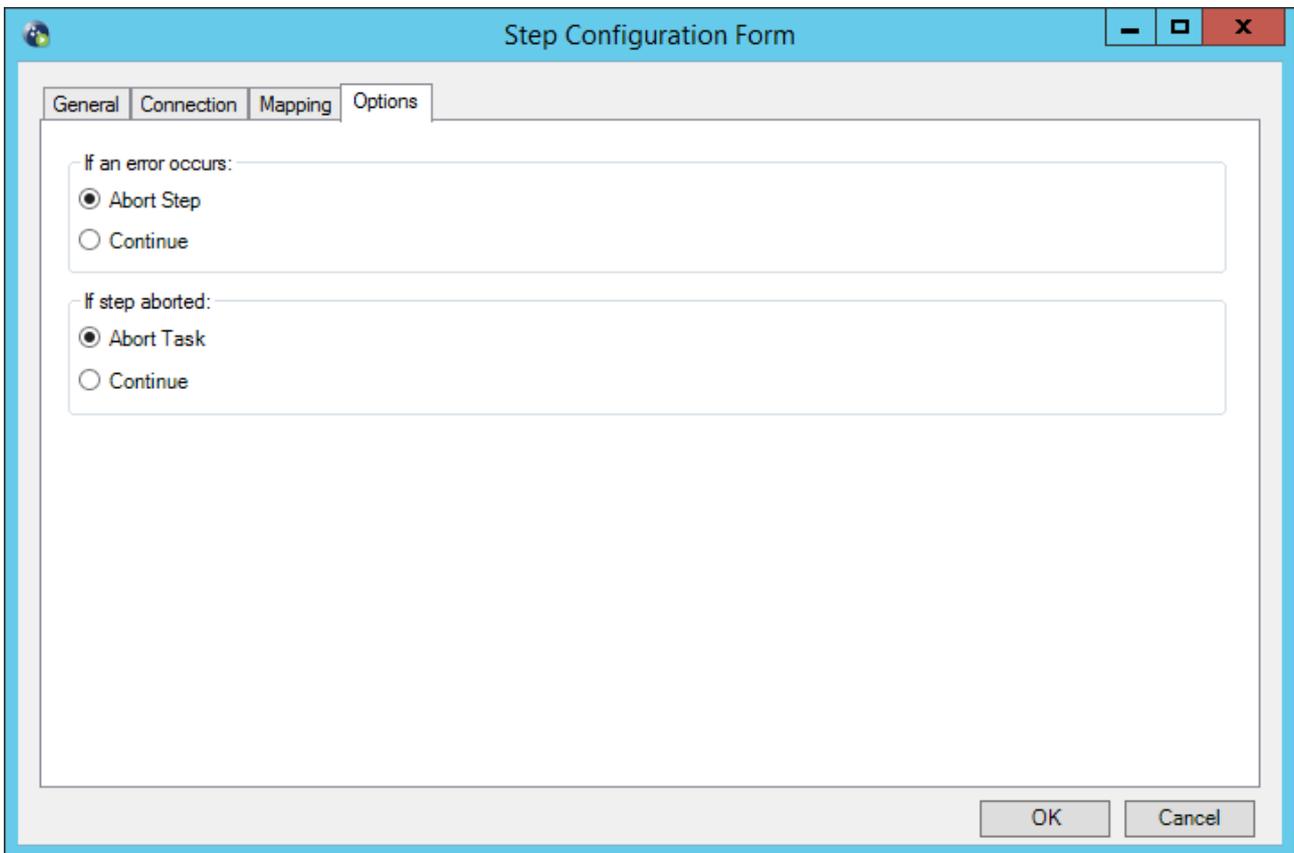
The right-hand pane is the input to the called task. It will display available variables in the task to have data mapped to.

**NOTE:** The variables must have previously been created in the called task and be of a 'parameter' type in order to be available.

Additionally, a field is present named `SupplementaryReference`. It allows for traceability throughout the process of transferring data. When mapped, data is not passed to the called task but instead it will reside locally at run time. It is added to the output of the Call Task step and creates a record for your reference purposes only, this means you can choose to map any kind of information to this field to assist with, for example, advising where the data originated from or at what time data transfer occurred.

## Options Tab

The **Options** tab contains options for run time error handling. The options dictate whether the current task should continue or abort in case of a processing failure.



The image shows a screenshot of a software window titled "Step Configuration Form". The window has a blue title bar with standard minimize, maximize, and close buttons. Below the title bar, there are four tabs: "General", "Connection", "Mapping", and "Options". The "Options" tab is currently selected. The main content area of the "Options" tab contains two sections, each with a label and a list of radio button options:

- If an error occurs:**
  - Abort Step
  - Continue
- If step aborted:**
  - Abort Task
  - Continue

At the bottom right of the window, there are two buttons: "OK" and "Cancel".

Figure 8. Call Task step — Options tab

# Objects Exposed

Objects that a Call Task step exposes which can be utilised by other steps are listed below:

- **OutputData** — This object exposes the results of the Call Task step output in XML. This is particularly useful when used with the option **RunSync** within the tool. This will allow the object to capture data stored in any output variable from the secondary task after it completes. This can then be utilised by another step in the original task that can take advantage of an XML output. Compatible steps are listed in the 'Working with other steps' section towards the beginning.
- **ErrorData** - If the task fails and produces an error, this data is exposed via an XML format.

Standard step properties are also available allowing you to utilise statistical data of the Call Task step.

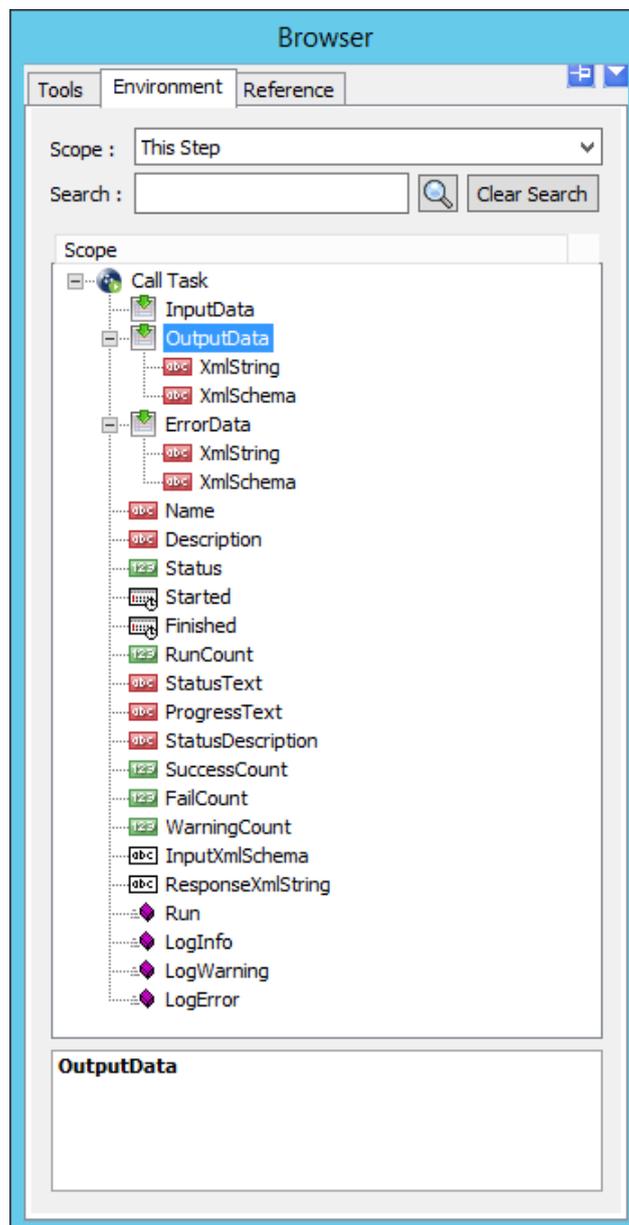


Figure 9. Call Task step — Task Browser — Showing **OutputData** and **ErrorData** objects

# User Security

The **Call Task** tool will appear within the **User Properties** dialog and under the **Tool Access** tab. Only users with the tool checked will have permission to use the tool when building a task.

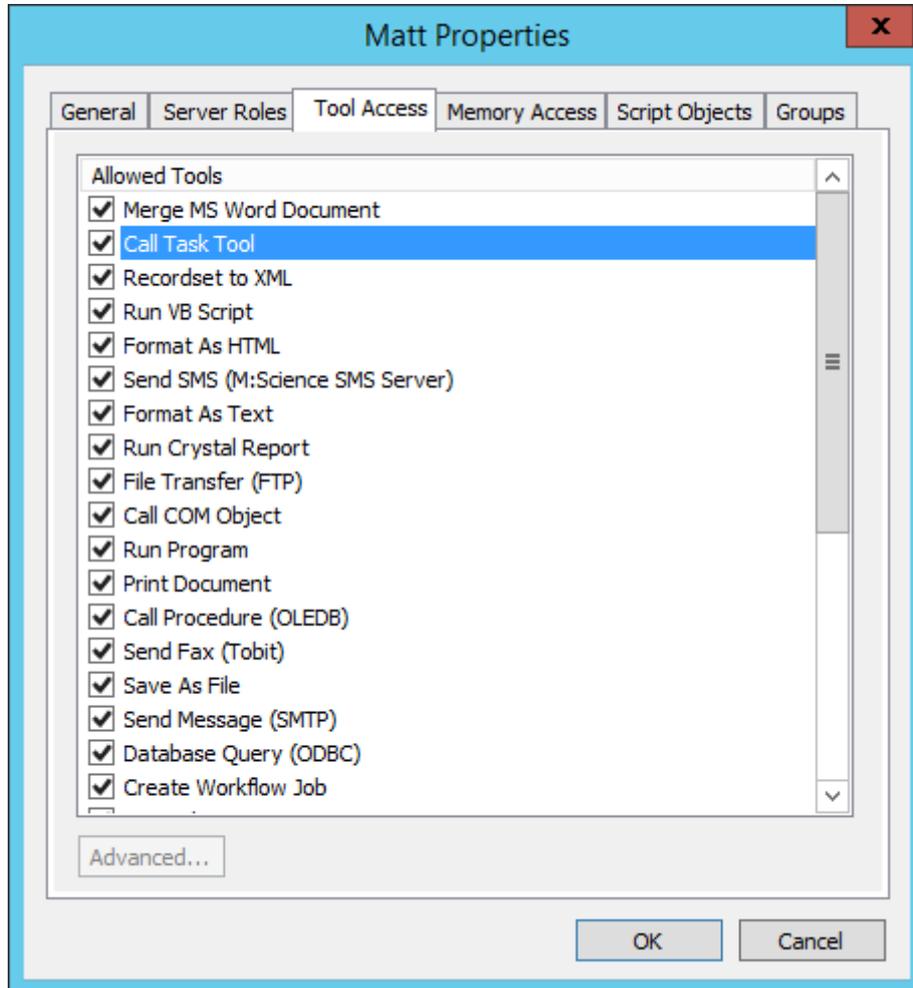


Figure 10. User Properties — Tool Access

## Summary

The **Call Task** tool provides a method of calling another task to run, allowing variables and XML based data to be passed between the task making the call and the task being called.

It offers a user interface allowing for the precise configuration of the TaskCentre API meaning a VB script is now no longer a requirement when a task needs to trigger another.

For further information about this tool, or about TaskCentre in general, please do not hesitate to contact us:

Codeless Platforms  
Suite 1 & 2 Bourne Gate  
25 Bourne Valley Road  
Poole  
BH12 1DY  
United Kingdom  
Tel: +44 (0) 330 99 88 700  
Email: [enquiries@codelessplatforms.com](mailto:enquiries@codelessplatforms.com)

