

BPA
Platform

Technical Overview

Run VBScript Tool

Copyright

The copyright in this document is owned by Orbis Software Ltd T/A Codeless Platforms 2020. 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 1 & 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®".

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The information provided in this publication may contain errors, omissions, or typographical errors or may be out of date.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
Features	1
Working with other Tools	2
Consuming from Other Tools	2
Objects Consumed	3
Exposing to Other Tools	3
Step Configuration	4
About the General Tab	4
About the Main Tab	5
About the Options Tab	6

Introduction

The **Run VBScript** tool is used to create a task step that runs a Visual Basic Script using data produced by other steps in the task.

The step also provides access to the **Memory** feature which allows data produced by the step to be memorised into a Repository.

Features












- ▶ Copy-and-paste script into the editor from external documents.
- ▶ Drag-and-drop common scripts from the **Reference** tab of the Task Browser dialog.
- ▶ Memory feature to memorise task data created at run-time.



Working with other Tools

The **Run VBScript** tool can directly interact with the following tools:

Consuming from Other Tools

The **Run VBScript** tool can consume objects exposed by the following tools:

Icon	Tool Name	Tool Category
	Database Query (ODBC)	Input and Data Connectors
	Database Query (OLEDB)	Input and Data Connectors
	Import Flat File	Input
	Convert XML to Recordset	Format
	Create Workflow Job	Format
	Format as Flat File	Format
	Format as HTML	Format
	Format as HTML Pro	Format
	Format as Text	Format
	Run Crystal Report	Format
	Run Microsoft Reporting Services	Format

Icon	Tool Name	Tool Category
	Run Microsoft Word (Merge)	Format
	Filter Data	General

Objects Consumed

The following objects, outputted by the above tools, can be consumed by the **Run VBScript** tool:

- ▶ **Recordset** — Tabular data from any BPA Platform tool capable of exposing such data (see above)
- ▶ **Documents (Text)** — Plain text documents
- ▶ **Documents (HTML)** — Standalone HTML pages
- ▶ **Documents (Paged HTML)** — Paginated HTML files
- ▶ **Documents (PDF)**
- ▶ **Documents (RTF)** — Rich text documents
- ▶ **Documents (XML)**
- ▶ **Documents (CSV)**

Exposing to Other Tools

As an Output category tool, **Run VBScript** does not expose any objects that can be directly consumed by another tool. However, it does expose the following objects which can be used in a task step sequence:

- ▶ **RecordSource** — If an **Input Recordset** has been selected, this contains the columns included in the recordset
- ▶ **DocumentSource** — If a **Document Source** has been selected, this contains the data in the document as recordset columns
- ▶ **Memory Definitions** — If configured, this is a list of **Memory Definitions** created for the step and the recordset columns to be memorised for each one
- ▶ **Step Properties** — Standard step properties are available allowing you to use statistical data of the tool

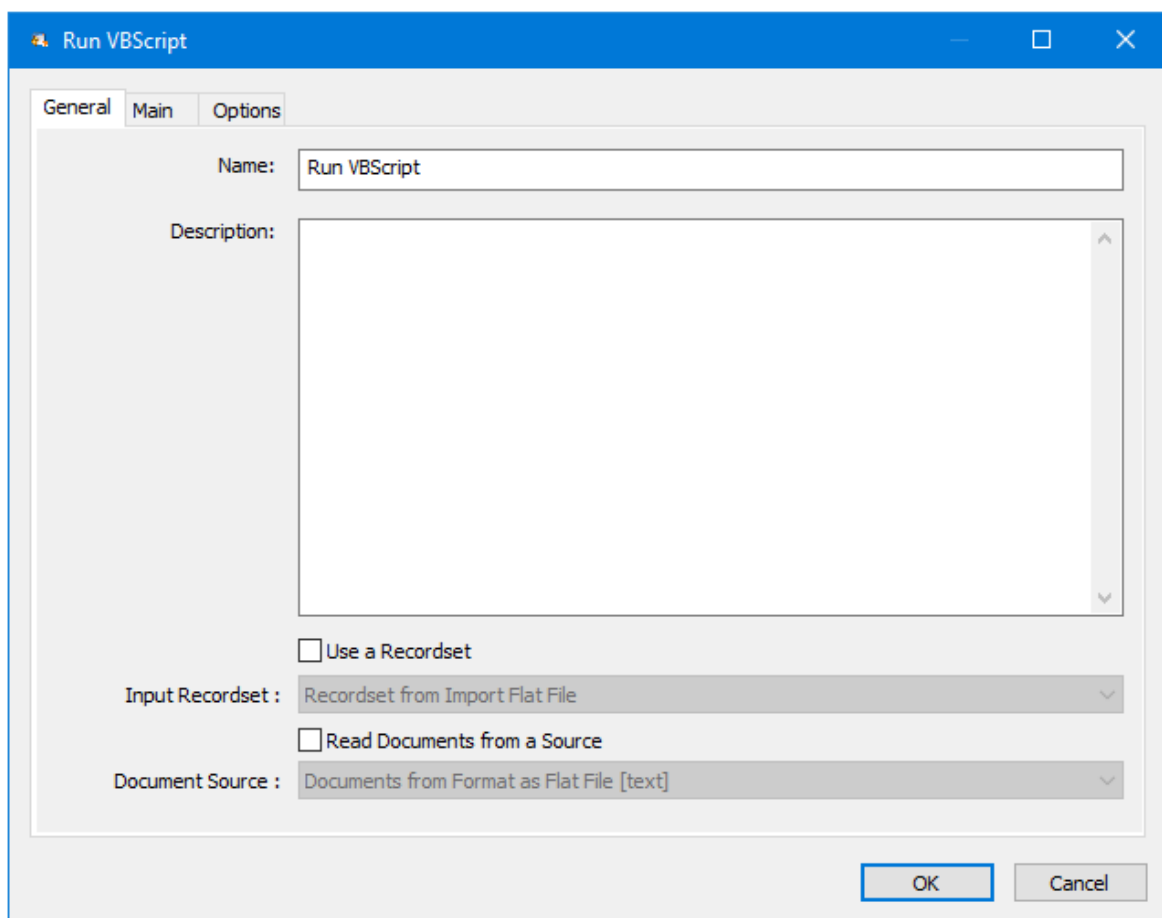
Step Configuration

To add a new **Run VBScript** step to an existing task, you either:

- ▶ Click and drag the **Run VBScript** icon from the **Task Browser** to the task **Design** area.
- ▶ From the task's **Design** tab, right-click on empty space and select **Add > Execute > Run VBScript**.

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

About the General Tab



The screenshot shows the 'Run VBScript' dialog box with the 'General' tab selected. The dialog has three tabs: 'General', 'Main', and 'Options'. The 'General' tab contains the following fields and options:

- Name:** A text box containing 'Run VBScript'.
- Description:** A large text area for entering a description.
- ☐ **Use a Recordset**
- Input Recordset :** A dropdown menu showing 'Recordset from Import Flat File'.
- ☐ **Read Documents from a Source**
- Document Source :** A dropdown menu showing 'Documents from Format as Flat File [text]'.

At the bottom right, there are 'OK' and 'Cancel' buttons.

The **General** tab is used to enter the following details for the step:

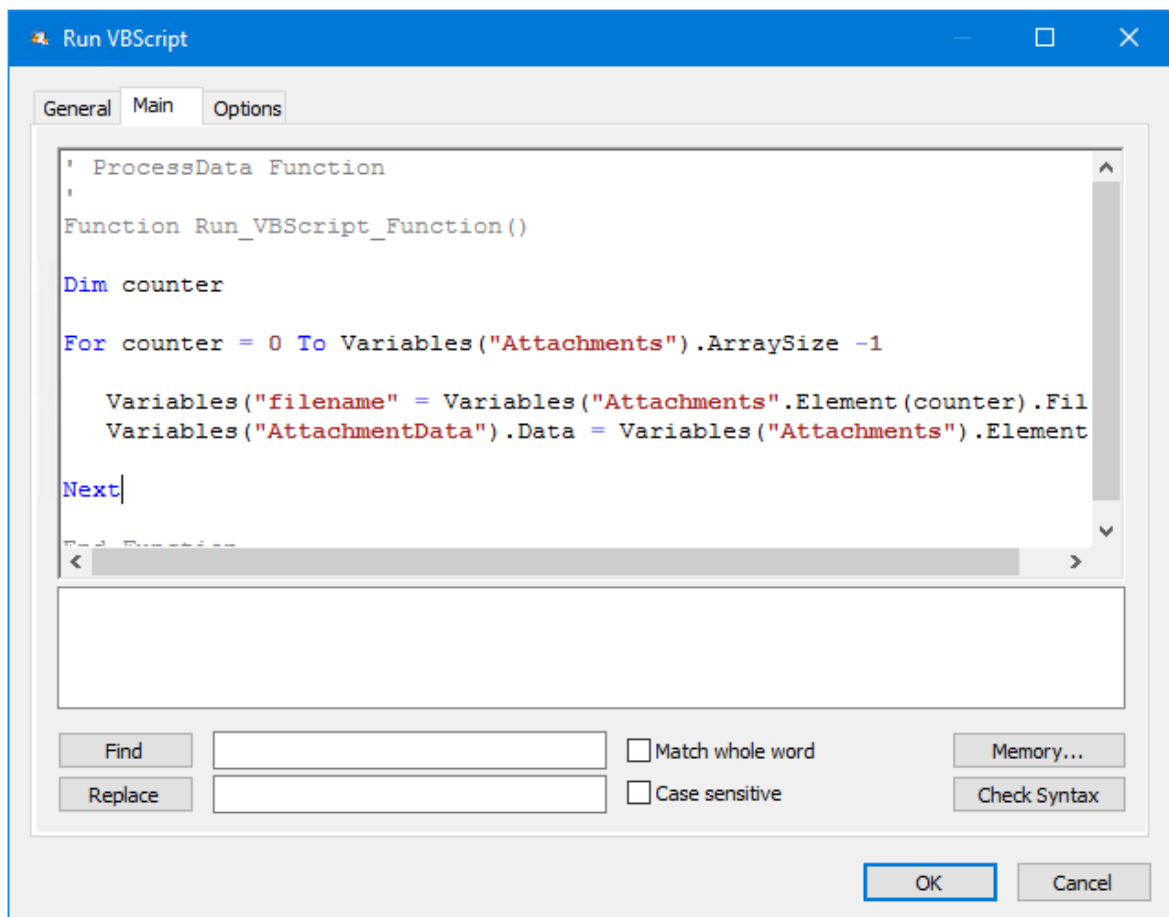
- ▶ **Name** — Enter a meaningful name for the step

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

- ▶ **Description** — If required, enter a description of this step
- ▶ **Use a Recordset** — Enable this parameter if recordset data from a previous task step is required when running the script
 - ☐ **Input Recordset** — Contains all available recordsets from steps previously created in the task
- ▶ **Read documents from a source** — Enable this parameter if document data from a Format task step is required when when running the script
 - ☐ **Document Source** — Contains all available documents from Format steps previously created in the task; note that the document format is displayed after the step name

About the Main Tab

The **Main** tab is used to enter the Visual Basic (VB) script for the step.

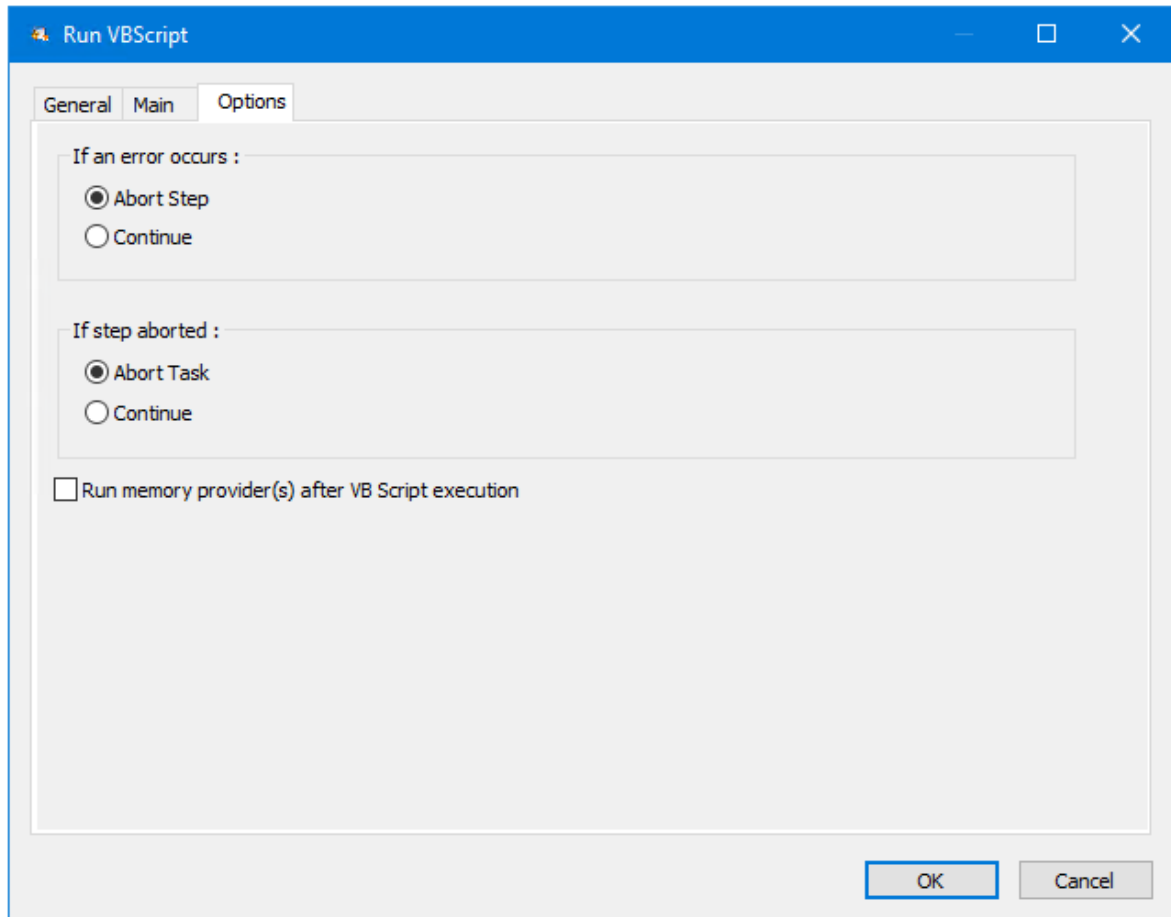


Enter the script in the text input area. Variables, properties, recordset columns, and so on can be dragged and dropped into the script from the Task Browser. Use **Check Syntax** for any errors that may be in the script — results are displayed in the box underneath your script.

Use **Memory** to memorise some or all of the step output for use in other task steps. This is particularly useful when the **Run VBScript** step is called multiple times — use **Memory** to compare the record previously processed to the new record currently being processed to avoid duplicates.

About the Options Tab

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



The screenshot shows a Windows-style dialog box titled "Run VBScript". It has three tabs: "General", "Main", and "Options", with "Options" currently selected. The dialog contains two main sections. The first section, "If an error occurs :", has two radio buttons: "Abort Step" (which is selected) and "Continue". The second section, "If step aborted :", also has two radio buttons: "Abort Task" (selected) and "Continue". Below these sections is a checkbox labeled "Run memory provider(s) after VB Script execution", which is currently unchecked. At the bottom right of the dialog are "OK" and "Cancel" buttons.

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**).

If **Run memory provider(s) after VB Script execution** is enabled, then configured memory providers are run after script execution.

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

