



CTX-UiPath User Guide

Contents

CTX-UiPath User Guide	1
Contents	2
Preface	3
About this Manual	3
Audience	3
Related Material	3
Abbreviations used in this Document	3
1 Versions	4
1.1 Document Revisions	4
1.2 Module Versions	4
2 Requirements	5
3 Integration	6
3.1 Integration with Third-Party Systems	6
3.1.1 Orchestrator Web API	6
3.1.2 Orchestrator PowerShell Module	6
3.1.3 PowerShell Remoting	6
3.1.4 Cortex PowerShell Agent - Remote Script Execution	7
3.2 Integration with Existing Infrastructure	7
4 Module Subtasks	8
4.1 DRCP-Deploy-Robot-Cortex-PSAgent	8
4.1.1 Overview	8
4.1.2 Input variables	8
4.1.3 Output Variables	9
4.2 EUP-Execute-UiPath-Process	9
4.2.1 Overview	9
4.2.2 Input variables	9
4.2.3 Output Variables	10

Preface

About this Manual

The User Guide is designed to give an overview of the CTX-UiPath module.

Audience

This document is designed for people interested in using the CTX-UiPath module to execute a UiPath process on a remote machine.

Related Material

Document
CTX-UiPath Deployment Plan
CTX-UiPath.studiopkg

Abbreviations used in this Document

None.

1 Versions

1.1 Document Revisions

The following revisions have been made to this document.

Date	Revision	Notes
07/06/2019	1.0	First Draft

1.2 Module Versions

The updates have been made to this module, starting with the most recent.

Module Version	Release Date	Comments
1.0	27/06/2019	Creation of: <ul style="list-style-type: none">• EUP-Execute-UiPath-Process• DRCP-Deploy-Robot-Cortex-PSAgent

2 Requirements

This document details all the items required to use the CTX-UiPath module.

Requirements:

- A minimum of UiPath Studio Enterprise edition 19.4.3 installed on the UiPath Server.
- A minimum of Cortex v6.5 installed on the Cortex Server.
- A minimum of PowerShell v5 installed on the Cortex Server.
- Windows Task Scheduler for a minimum of Windows Server 2008 installed on the UiPath Server.
- Connectivity between the Cortex Server and the UiPath Server.

3 Integration

3.1 Integration with Third-Party Systems

There are several potential ways to dynamically interact with UiPath, detailed below along with the reasoning why certain methods were not chosen.

3.1.1 Orchestrator Web API

The initial avenue of integration, this would have involved creating of a Cortex Subtask per API endpoint, allowing complete access to the Cloud Platform's functionality using Cortex HTTP request blocks.

However, the change from the UiPath Orchestrator client to a cloud platform had the disadvantage that if the user's UiPath machine does not have external internet access, then their robots will not have access to the cloud platform. As this configuration is relatively common in enterprise systems, this integration method was abandoned.

3.1.2 Orchestrator PowerShell Module

The 'Orchestrator-PowerShell' PowerShell module has been provided on the official UiPath GitHub, found [here](#), promising interaction with UiPath Orchestrator. But, upon further investigation, the module simply provides a simpler way to interact with the Orchestrator Web API, so the same disadvantage as in section 3.1.1 applies.

3.1.3 PowerShell Remoting

In order to execute a process directly from it's .xml workflow file as opposed to with a robot from UiPath Orchestrator, the following command must be executed in the CMD console on the UiPath machine:

```
Call "<Install location>\UiRobot.exe" -file "<Save location>\<Process name>\Main.xaml" -input "<Input JSON>" --rdp
```

The command calls the UiRobot service, telling it to run a process with the provided input arguments in a remote desktop session. The user that this session is run under is the same as the one that the CMD console is opened by.

The following table details the highlighted parameters:

Parameter	Description	Example Value
<Install location>	The location in which the UiRobot service has been installed	C:\Program Files (x86)\UiPath\Studio
<Save location>	The location in which UiPath process are saved	C:\Users\<USER>\Documents\UiPath
<Process name>	The name of the UiPath process to be executed	TestProcess
<Input JSON>	Values of the input arguments to the process in a compressed JSON format.	{'inputText':'Hello World','inputInt':1}

To remotely run the above command, the following 2 step process was to be followed:

1. Execute the `Invoke-Command` PowerShell cmdlet using Cortex on the Cortex application server, targeting the UiPath machine with relevant credentials.
2. In the `-ScriptBlock` parameter, use the `Start-Process` cmdlet to open a CMD console and pass it the above command.

However, attempting this and multiple permutations of the many possible parameters of both Cmdlets continuously returned the following exception:

```
There was no endpoint listening at  
net.pipe://localhost/UiPath/service/duplex/agent/<USER> that could  
accept the message.
```

After some investigation it was concluded that in order to make the command execute correctly, the process opening the CMD line must be started in administrator mode, even if the user being utilised is an administrator themselves.


As can be seen in the Microsoft documentation, found [here](#), it is seemingly impossible to use the `-Verb` and `-RedirectStandardOutput` parameters simultaneously. Simply put, recording the output of the process and running it from an administrator command line was not conceivable.

Since the output of processes' execution is hugely important for their orchestration by Cortex, this method of integration with UiPath was changed slightly, as discussed in Section 3.1.4.

3.1.4 Cortex PowerShell Agent - Remote Script Execution

The Cortex PowerShell Agent is used to allow different Cortex application servers to run scripts remotely on each other's PowerShell consoles. It is installed as a windows service, logging on as Local System by default, and can be used to run the CMD command in Section 3.1.3. As described there, this will trigger the execution of processes in their own Windows sessions, and it is possible to trigger multiple executions at the same time running under different users.

This method is currently used by the **EUP-Execute-UiPath-Process** subtask included in the CTX-UiPath module to run processes in the background.

 Since processes executed like this will be running in the background, it is necessary to construct UiPath process workflows accordingly, i.e. as if they were to be executed by a back-office robot from orchestrator. A helpful UiPath knowledge base article on 'Background Automation' can be found [here](#).

In order to execute a process remotely, and have it run in a logged-on user's remote desktop session (i.e. not in the background), it can be run as a task once created as such in Windows Task Scheduler. So long as the user that is logged in to the remote desktop session on the UiPath machine is the same as the one that a specified Cortex PowerShell Agent uses, then Cortex will use this method to force UiPath to use their desktop to execute the process.

These methods require the installation of a Cortex PowerShell Agent on a UiPath machine. These are the only Cortex components required on the UiPath machine, see the CTX-UiPath Deployment Plan for more information on how to set them up.

3.2 Integration with Existing Infrastructure

None required.

4 Module Subtasks

4.1 DRCP-Deploy-Robot-Cortex-PSAgent

4.1.1 Overview

Given that a Cortex PowerShell Agent is installed on the UiPath machine, this subtask will remotely install a custom agent for a different user. For more information on using this subtask, see the CTX-UiPath Deployment Plan, Section 4.2.

4.1.2 Input variables

Name	Type	Comments
DRCP_i_Domain	Text	Required, The domain on which the UiPath machine is located.
DRCP_i_UiPath-Host	Text	The name of the UiPath machine. Default value is 'Localhost'
DRCP_i_UiPath-Host-User	Text	Required, The username of a user that has PowerShell remoting rights.
DRCP_i_UiPath-Host-Password	Text	Required, The password associated with the user specified in DRCP_i_UiPath-Host-Password.
DRCP_i_GSI-Port	Integer	Required, The GSI port that the Cortex application server's Generic Interface will use to communicate with the Cortex PowerShell Agent on the UiPath machine. This MUST be between 22102 and 65535 and allowed inbound access through the UiPath machine's firewall. It is recommended that this is one less than DRCP_i_GSI-Port-Next for convenience.
DRCP_i_GSI-Port-Next	Integer	Required, The secondary GSI port that the Cortex application server's Generic Interface will use to communicate with the Cortex PowerShell Agent on the UiPath machine. This MUST be between 22102 and 65535 and allowed inbound access through the UiPath machine's firewall. It is recommended that this is one more than DRCP_i_GSI-Port for convenience.
DRCP_i_Deploy-Package-Location	Text	Required, The location of the Cortex PowerShell Agent .msi on the UiPath machine.
DRCP_i_Deploy-Package-Name	Text	The name of the Cortex PowerShell Agent .msi on the UiPath machine, not including the file extension. Default value is 'Cortex PowerShell Agent'.
DRCP_i_PSAgent-Target-Location	Text	The location where the Cortex PowerShell Agent Service for the robot is to be installed. Default value is 'C:\Program Files (x86)\Cortex'.
DRCP_i_Service - User	Text	Required, The user that the Cortex PowerShell Agent being installed will run under.
DRCP_i_Service - Password	Text	Required, The password associated with DRCP_i_Service-User.

4.1.3 Output Variables

None.

4.2 EUP-Execute-UiPath-Process

4.2.1 Overview

Executes a user-specified process on a UiPath machine.

- ✎ In order for the process to be executed successfully in the background, i.e. with the EUP_i_Run-In-Background input variable set to "True", it is necessary for it to be constructed for Background Automation, as discussed in Section 3.1.4.
- ✎ The UiRobot should **NOT** be connected to UiPath orchestrator since Cortex takes the place of this platform regarding the execution of processes.

4.2.2 Input variables

Name	Type	Comments
EUP_i_UiPath-Host	Text	The name of the machine on which the UiPath process is to be run. Default value is 'LocalHost'.
EUP_i_UiPath-Port	Integer	The GSI port of the desired Cortex PowerShell Agent. Default value is 22100, for the regular Cortex PowerShell Agent. Any further installed Agents will have had their ports defined when deploying them either manually or as the input 'DRCP_i_GSI-Port' to the subtask DRCP-Deploy-Robot-Cortex-PSAgent.
EUP_i_Process-Inputs	List	Input arguments for the Process. Each element must be a structure containing exactly two elements. One of these elements must have the name "Argument-Name" and contain the name of an input argument. The second element must have the name "Argument-Value" and contain the value of the corresponding input argument. In JSON format, this is of the form: <pre>[{ "Argument-Name": "<Name of first argument>", "Argument-Value": "<Value of first argument>" }, { "Argument-Name": "<Name of second argument>", "Argument-Value": "<Value of second argument>" }, etc]</pre>
EUP_i_UiPath-Install-Directory	Text	The UiPath service install directory. Default value is 'C:\Program Files (x86)\UiPath\Studio'.
EUP_i_Process-Directory	Text	Required, the directory in which the process is saved. Typically, this would be: 'C:\Users\<user>\Documents\UiPath\<process name>'

EUP_i_Process-Name	Text	Required, the name of the process.
EUP_i_Run-In-Background	Text	Whether the process should run in the background or not, as in section 3.1.4. Default value is "True", should only take the value of either "True" or "False".
EUP_i_Timeout	Text	Number of seconds before the process execution should time out. Default value is "30", this parameter is only used if EUP_i_Run-In-Background is "False" or not provided.

4.2.3 Output Variables

Name	Type	Comments
EUP_o_Process-Outputs	Structure	Contains the output arguments of the process given than it executed correctly.