cortex®

The Evolution of Automation

CTX-PRTG User Guide

Contents

Preface

About this Manual

This document is a user guide for the CTX-PRTG module.

Audience

The audience for this document is those wanting to understand how to use CTX-PRTG module.

Related Material

Document
CTX-PRTG – Deployment Plan
CTX- PRTG.studiopkg

Abbreviations used in this Document

None

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
23/04/2019	1.0	First Release

Module Versions

The following revisions have been made to this module

Date	Revision	Notes
26/04/2019	1.0	Creation of: • PGHD-PRTG-Get-Historic-Data • PAA-PRTG-Acknowledge-Alarm • PGSS-PRTG-Get-System-Status • PRA-PRTG-REST-API

Requirements

This section details all the items required to deploy the PRTG Subtasks.

Requirements:

- An instance of PRTG running on the target server
- Cortex connectivity to PRTG server
- A minimum of Cortex v6.4 Installed on the Cortex Server

Integration

Integration with Third-Party Systems

This subtasks in this module interact with PRTG via a REST API. The user can retrieve historical data from PRTG, get live data and acknowledge alarms. Each of these capabilities are contained within a Cortex subtask, detailed below.

To use these subtasks there is a need for a user to have access to perform the actions in PRTG. The three details to make calls to PRTG are:

- Instance URL
- Username
- Password (or hashed password)

Integration with Existing Infrastructure

None Required.

1 PRTG Subtasks

1.1 PGHD-PRTG-Get-Historic-Data

1.1.1 Overview

Gets historic data from PRTG.

1.1.2 Input variables

Name	Type	Comments
PGHD_i_PRTG-URL	Text	e.g. https://10.8.0.161:8443/api
PGHD_i_PRTG-Username	Text	username used to query PRTG
PGHD_i_PRTG-Password	Text	this is the raw password and is required if the hashed password is not passed in
PGHD_i_PRTG-Sensor-ID	Text	ID of the sensor that data is to be retrieved for
PGHD_i_PRTG-Password-Hashed	Text	this is the hashed password and is required if the raw text password is not supplied
PGHD_i_PRTG-Start-Date	Text	start date from which the data should be retrieved. e.g. 2019-04-26-00-00-00
PGHD_i_PRTG-End-Date	Text	end date till which the data should be retrieved. e.g. 2019-04-27-00-00-00
PGHD_i_Return-Format	Text	format in which the data should be returned. This can take the values ‘xml’, ‘json’ or ‘csv’

1.1.3 Output variables

Name	Type	Comments
PGHD_o_Response	Text	raw response from PRTG
PGHD_o_Exception-Details	Structure	details of any exceptions that may occur during execution. If no exceptions occur, an empty structure will be returned

1.2 PAA-PRTG-Acknowledge-Alarm

1.2.1 Overview

This subtask will acknowledge an alarm in PRTG with message ‘Acknowledged by Cortex’.

1.2.2 Input variables

Name	Type	Comments
PAA_i_PRTG-URL	Text	e.g. https://10.8.0.161:8443/api

PAA_i_PRTG-Username	Text	username used to query PRTG
PAA_i_PRTG-Password	Text	this is the raw password and is required if the hashed password is not passed in
PAA_i_PRTG-Sensor-ID	Text	ID of the sensor for which the alarm will be acknowledged.
PAA_i_PRTG-Password-Hashed	Text	this is the hashed password and is required if the raw text password is not supplied

1.2.3 Output variables

Name	Type	Comments
PAA_o_Response	Text	raw response from PRTG.
PAA_o_Exception-Details	Structure	details of any exceptions that may occur during execution. If no exceptions occur, an empty structure will be returned.

1.3 PGSS-PRTG-Get-System-Status

1.3.1 Overview

This subtask will get the current system status along with details of sensors configured in PRTG.

1.3.2 Input variables

Name	Type	Comments
PGSS_i_PRTG-URL	Text	e.g. https://10.8.0.161:8443/api
PGSS_i_PRTG-Username	Text	username used to query PRTG
PGSS_i_PRTG-Password	Text	this is the raw password and is required if the hashed password is not passed in
PGSS_i_PRTG-Password-Hashed	Text	this is the hashed password and is required if the raw text password is not supplied
PGSS_i_Return-Format	Text	format in which the data should be returned. This can take the values ‘xml’ or ‘json’

1.3.3 Output variables

Name	Type	Comments
PGSS_o_System-Status	Text	status of the system
PGSS_o_Sensor-States	Text	state of each sensor in PGRT

PGSS_o_Sensor-Types	Text	types of sensors in PRTG
PGSS_o_Exception-Details	Structure	details of any exceptions that may occur during execution. If no exceptions occur, an empty structure will be returned.

1.4 PRA-PRTG-REST-API

1.4.1 Overview

This subtask calls PRTG and can be used to make custom calls that are not included in the CTX-PRTG.studipkg file.

1.4.2 Input variables

Name	Type	Comments
PRA_i_URL	Text	e.g. https://10.8.0.161:8443/api
PRA_i_URL-Part	Text	e.g. getstatus.xml?id=0&username=xxxx&passhash=yyyy

1.4.3 Output variables

Name	Type	Comments
PRA_o_PRTG_Response	Text	response from PRTG

2 Appendix - Using PRTG to trigger Cortex

For PRTG to trigger Cortex a Notification needs to be created. Cortex flows can be best triggered via a REST call with a json body, however due to the limitations of the PRTG Rest API call this process needs to include a PowerShell script.

2.1 Setup of PowerShell file

2.1.1 PowerShell Script Configuration

1. The user should copy the PowerShell script named Trigger-Cortex.ps1 to the EXE subfolder under the Notifications folder of the PRTG Network Monitor program location, e.g.

   C:\Program Files (x86)\PRTG Network Monitor\Notifications\EXE

2. The contents of the Trigger-Cortex.ps1 file can be found below in Section 2.1.1.
3. Once the file is copied, the parameters at the top should be configured.
4. Under the `#Set URI` comment, the string `<cortex-server>` should be changed to the location of the cortex application server.

2.1.2 Trigger-Cortex PowerShell Script

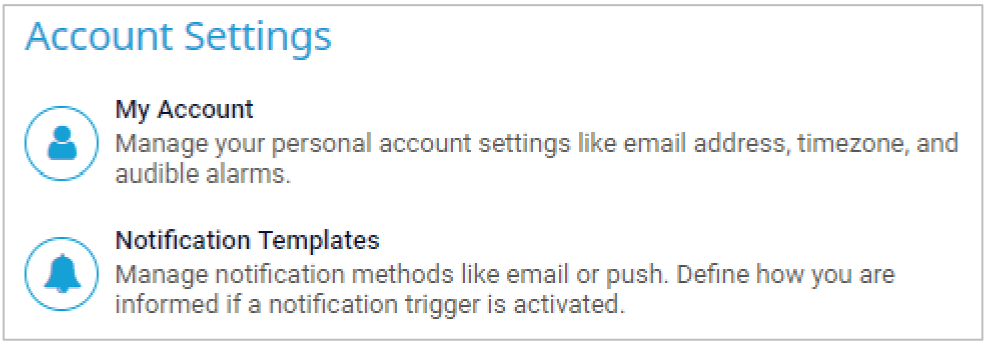
The contents of the Trigger-Cortex.ps1 file can be copied from here:

```
  param (
    [string]$sitename = "Placeholder for Sitename",
    [string]$device = 'Placeholder for Device',
    [string]$name = 'Placeholder for Name',
    [string]$status = 'Placeholder for status',
    [string]$down = '',
    [string]$message = 'Placeholder for Message text',
    [string]$sensor = 'Placeholder for sensor name',
    [string]$sensorid = 'Placeholder for sensorid',
    [string]$datetime = 'Placeholder for datetime'
 )

function Get-BasicAuthCreds {
    param([string]$Username,[string]$Password)
    $AuthString = "{0}:{1}" -f $Username,$Password
    $AuthBytes  = [System.Text.Encoding]::Ascii.GetBytes($AuthString)
    return [Convert]::ToBase64String($AuthBytes)
}

#Set authorization
$username = 'CortexFlow'
$password = 'C0rt3xFl0w'
$BasicCreds = Get-BasicAuthCreds -Username $username -Password $password


#Set URI
$uri = "http://<cortex-server>:10000/api/flow/StartFlowAsync"


#Set headers
$headers = @{}
$headers.Add("Accept","application/json")
$headers.Add("Authorization","Basic $BasicCreds")


#Add variables
$argstructure = @{PRTG_SiteName="$sitename"
            PRTG_Device="$device"
            PRTG_Name="$name"
            PRTG_Status="$status"
            PRTG_Down="$down"
            PRTG_Message="$message"
            PRTG_Sensor="$sensor"
            PRTG_SensorID="$sensorid"
            PRTG_Timestamp=$datetime
            }

$args = @{g_i_MessageType='PRTG'
          'g_i_Message-Received' = $argstructure}


#Build up body
$bodyraw = @{Name='self-healing-process-incoming-loglake-message'
             arguments=$args}
$body = (ConvertTo-Json $bodyraw)


#Trigger Cortex flow
Invoke-RestMethod -Uri $uri -Method Post -Body $body -ContentType
'application/json' -Headers $headers
```

2.2 PRTG Setup

2.2.1 Notification configuration

1. In PRTG, from the **Setup** menu, **Notification Templates** should selected be from the **Account Settings** section.

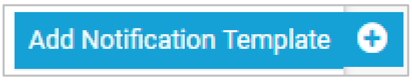


Account Settings

My Account
Manage your personal account settings like email address, timezone, and audible alarms.

Notification Templates
Manage notification methods like email or push. Define how you are informed if a notification trigger is activated.

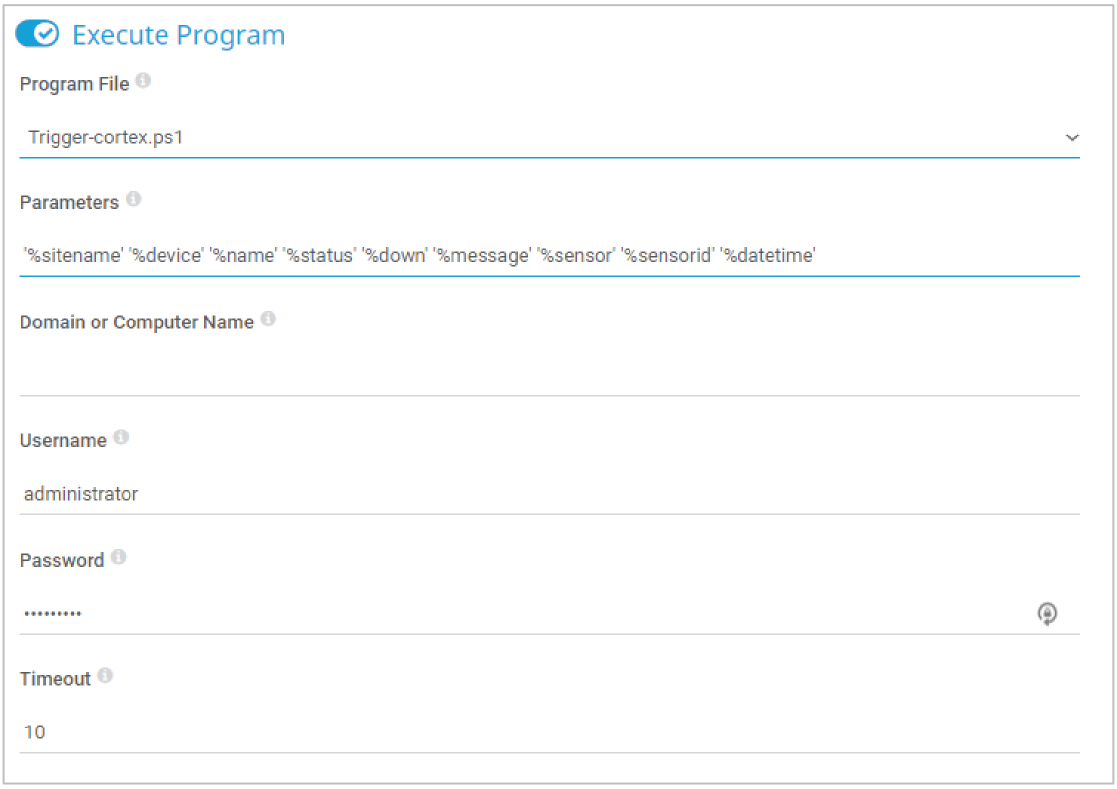
2. From the right-hand side select the **+** sign to add a new Notification template.



3. Enter a **Template Name**.
4. Under **Notification Summarization** select any value, although the option *"Always notify ASAP, never summarize"* is recommended.
5. Scroll further down and activate the **Execute Program** option.
6. From the Program File drop down select **Trigger-cortex.ps1**.
7. Next configure at least the following parameters (you can copy and paste the string below):

   '%sitename' '%device' '%name' '%status' '%down' '%message' '%sensor' '%sensorid' '%datetime'

⚐ You can add parameters at the end of the string. However, for that to work the Trigger-Cortex.ps1 script needs to be updated to reflect that too.



Execute Program

Program File: Trigger-cortex.ps1

Parameters: '%sitename' '%device' '%name' '%status' '%down' '%message' '%sensor' '%sensorid' '%datetime'

Domain or Computer Name:

Username: administrator

Password: •••••••••

Timeout: 10

8. Enter a Username and Password that have the rights to execute the PowerShell script and click the Create button to save the Notification Template.

2.2.2 Update Sensors to trigger Cortex

Once the Notification Template is created you can update the PRTG sensors to trigger Cortex.