Cortex Service Request Catalog

Contents

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
16/10/2020	0.1	First Draft & Release

Module Versions

This version of the Cortex Service Request Catalog is relevant up to version 1.0 of the Cortex Service Request Catalog module.

Preface

About this Manual

This document provides a guide on how to deploy the Cortex Service Request Catalog module in your Cortex system.

Audience

This document is intended for those who require the use of Cortex Service Request Catalog module.

Related Material

Document
Cortex Service Request Catalog – User Guide
Cortex Service Request Catalog.studiopkg
CTX-MD-SRC_create_schema.sql
Cortex Material Design Subtasks (https://github.com/IntelligentAutomationCommunity/CortexMaterialDesignSubtasks)

Abbreviations used in this Document

SQL Structured Query Language

DB Database

1 Requirements

This document details all the steps required to deploy the Cortex Service Request Catalog module.

Requirements:

- SQL Server Management Studio Access to the Cortex Database Server
- Minimum Cortex v7 installed on the Cortex Application Server
- Minimum SQL Server 2012 (version 11.0.7001.0) installed on the Cortex Database Server
- CTX-Configuration-Store installed on server
- Cortex Material Design Subtasks (download from GitHub https://github.com/IntelligentAutomationCommunity/CortexMaterialDesignSubtasks)
- Latest material-dashboard-Cortex.css

2 Import Cortex-Service-Request-Catalog

To deploy the Cortex Service Request Catalog module on your Cortex system, Cortex-Service-Request-Catalog Studio Package needs to be imported on your Cortex system. To do this:

- Download the Cortex-Service-Request-Catalog Studio Package
- Import the Studio Package in Cortex Gateway
- Ensure the relevant users have the required permissions in 'Studio Authorisation'

After this, all users in the authorised groups will be able to view and execute the subtasks.

3 Create CRTX-Extras Database

3.1 Overview

For the Cortex Service Request Catalog module to work, the CRTX-Extras database must exist on the server where the Cortex databases exist. The following steps instruct you how to deploy the database and schema.

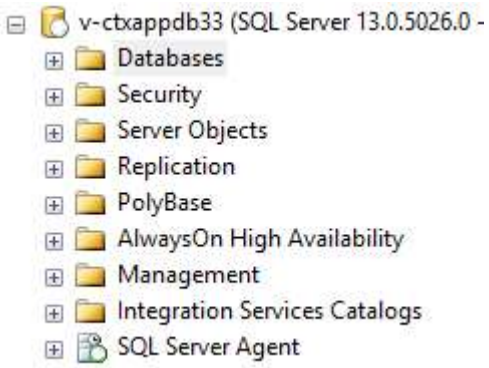
3.2 Create Database

1 Open Remote Desktop Connection to the Cortex database server or SQL Server Management Studio Remote Access

2 Open the script **CTX-MD-SRC_create_schema.sql** in SQL Server Management Server (SSMS) and connect to the DB engine where the query should be executed.

3 Execute the script. On the messages panel, you should see no errors

4 In the left-hand panel, click the plus to the left of 'Databases' to expand 'Databases'



5 Right click 'Databases' and click 'Refresh'.

6 There should now be a new database called CRTX-Extras containing 2 tables (cortex_MD_menu and cortex_MD_widget)

7 IMPORTANT:
The Cortex Service user needs to be assigned rights to read and write on this CRTS_Extras database.
In the left-hand panel, click the plus to the left of 'Security' to expand 'Security'.

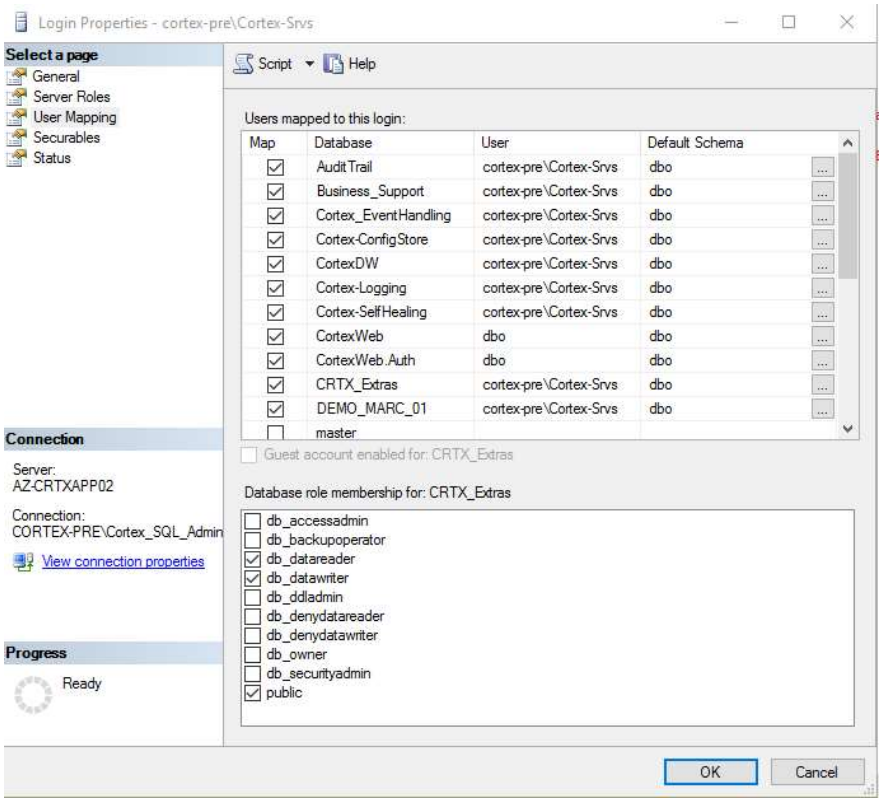
8 Click the plus to the left of 'Logins' to expand 'Logins'.

9 Right-click the cortex service user (e.g. Cortex-Srvs) and select Properties

10 From the popup window click *User Mapping* located on the left-hand upper corner

11 From the *Users mapped to this login* section click the CRTX_Extras database

12 From the *Database role membership for: CRTX_Extras* select **db_datareader** and **db_datawriter** and click the **OK** button



4 Additional Files

4.1 LivePortal Components

There is a need to copy the material-dashboard-Cortex.css style sheet onto the Cortex server running the LivePortal.

Item	Type	Server Destination (example)
material-dashboard-Cortex.css	Folder (CSS)	C:\inetpub\Cortex\Cortex\Styles

Note that the destination for the files may be different based on the Cortex LivePortal Site configuration in IIS.

4.1.1 Update custom.css Files

The custom.css file should be updated to include a reference to the material-dashboard-Cortex.css file. Open the custom.css that can be found in the same folder as above.

Add the following as the first line and save the file:

@import url('material-dashboard-Cortex.css');

4.1.2 Update MainSite.Master file

To ensure that the right fonts are loaded update the MainSite.Master file located in the parent folder of the styles folder above. Add the following lines before the </head> closing tag:

```
  <!--     Fonts and icons     -->

  <link rel="stylesheet" type="text/css"
href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700|Roboto
+Slab:400,700|Material+Icons" />

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-
awesome/latest/css/font-awesome.min.css">

</head>
```


5 Configuration Store

5.1 Config Setup

To use the dashboard, the Cortex Configuration Store is needed. Once this has been deployed, the UI flow should be used to enter the following Config Parameters and the relevant Values:

Area	Customer	Environment	Param_Name
Manual-Intervention	NULL	NULL	Cortex-URL
Manual-Intervention	NULL	NULL	LivePortal-URL

Note that Environment should be provided for instances where more than one Cortex Server exists.

Cortex-URL should be the base URL used to trigger a REST request to the Flow API, and LivePortal-URL is used to open the relevant UI pages when an execution has been triggered.

As an example, if you would normally connect to https://myApp.myDomain.com/liveportal you would want the following values:

Cortex-URL: https://myApp.myDomain.com

LivePortal-URL:
https://myApp.myDomain.com/liveportal/SequenceHandling.aspx?execution=

Note that IP Addresses can also be used for these values.

