



# CTX-Task-Scheduler User Guide

## Contents

---

CTX-Task-Scheduler User Guide .....	1
Contents .....	2
Versions .....	4
Document Revisions .....	4
Module Versions .....	4
Preface .....	5
About this Manual .....	5
Audience .....	5
Related Material .....	5
Abbreviations used in this Document.....	5
Requirements .....	6
Integration .....	7
Integration with Third-Party Systems .....	7
Integration with Existing Infrastructure .....	7
Task Scheduler Database.....	7
1    Cortex Task Scheduler Overview .....	8
1.1    Using the module .....	8
1.2    User Interface Experience .....	9
2    Cortex Task Scheduler Flows.....	12
2.1    CTS-Manage-Schedules .....	12
2.1.1 Overview .....	12
2.1.2 States.....	13
2.1.3 Inputs.....	14
2.1.4 Outputs.....	14
3    Cortex Task Scheduler Subtasks .....	15
3.1    CTS-DI-Database-Interaction .....	15
3.1.1 Overview .....	15
3.1.2 Inputs.....	15
3.1.3 Outputs.....	15
3.2    CTS-GS-Generate-Schedule .....	15
3.2.1 Overview .....	15
3.2.2 Inputs.....	16
3.2.3 Outputs.....	16
3.3    CTS-GSC-Generate-Schedule-Components .....	16
3.3.1 Overview .....	16
3.3.2 Inputs.....	17

3.3.3 Outputs.....	17
3.4 CTS-AUDS-Add-Update-Delete-Schedule.....	17
3.4.1 Overview .....	17
3.4.2 Inputs.....	18
3.4.3 Outputs.....	18

## Versions

---

### Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
04/12/2018	1.0	First Release

### Module Versions

The following revisions have been made to this module

Date	Revision	Notes
04/12/2018	1.0	<p>Creation of:</p> <ul style="list-style-type: none"><li>• CTS-Manage-Schedules</li><li>• CTS-DI-Database-Interaction</li><li>• CTS-GS-Generate-Schedule</li><li>• CTS-GSC-Generate-Schedule-Components</li><li>• CTS-AUDS-Add-Update-Delete-Schedule</li></ul>

## Preface

---

### About this Manual

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

### Audience

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

### Related Material

Document
CTX-Task-Scheduler – Deployment Plan
CTX-Task-Scheduler.studiopkg

### Abbreviations used in this Document

<b>OCI</b>	Orchestration Communication Interface
<b>UI</b>	User Interface
<b>DB</b>	Database

## Requirements

---

The CTX-Task-Scheduler module requires the following:

- Cortex Database OCI
- Cortex LivePortal

Instructions for how to install these are included in the deployment plan.

## Integration

### Integration with Third-Party Systems

None

### Integration with Existing Infrastructure

#### Task Scheduler Database

For the flows and subtasks to work in the CTX-Task-Scheduler module, the flow must connect to the Reactor DB to read and write data to the Scheduler tables.

The tables involved in the Task Scheduler schema are:

- CFG\_TASK– Table containing the definition of each task
- CFG\_TASK\_EXECUTION – Table storing the current scheduled executions
- CFG\_TASK\_SCHEDULE – Table storing the Schedule details for the Task Scheduler
- CFG\_TASK\_PARAMETERS – Table storing the Parameters for the scheduled tasks
- CFG\_PARAMETER\_TYPE – table storing the definitions of each parameter

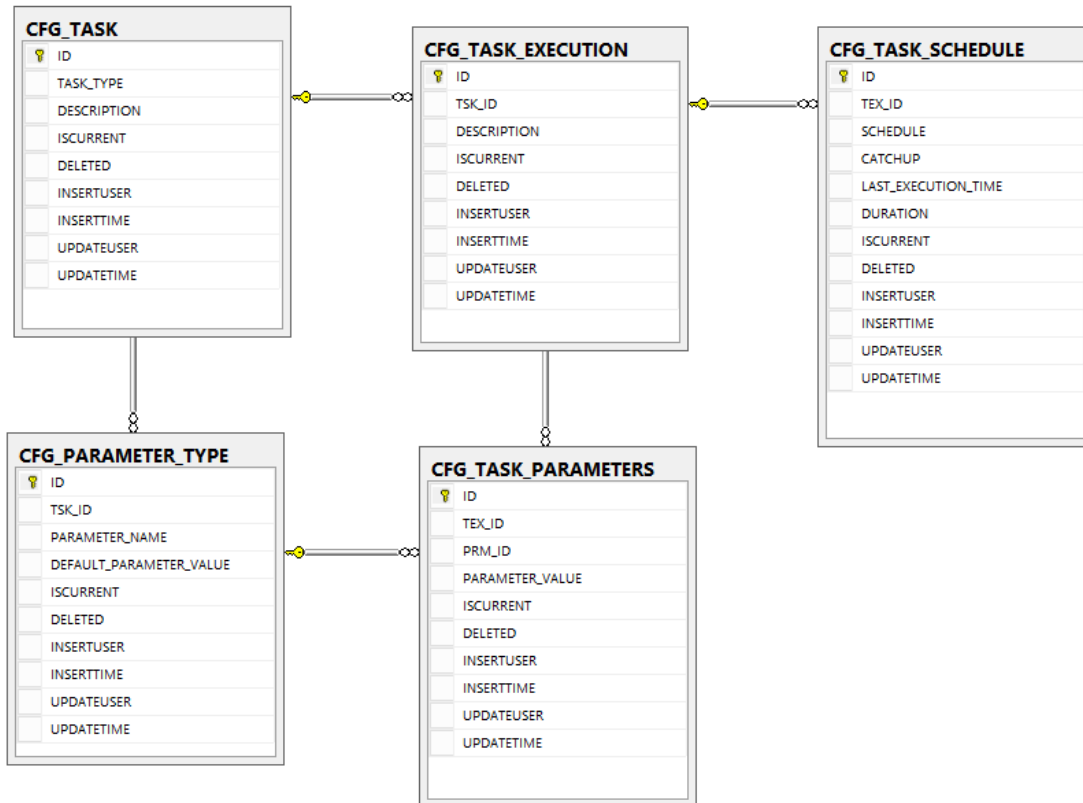


Figure 1 - Cortex Task Scheduling database schema

## 1 Cortex Task Scheduler Overview

---

The Cortex Task Scheduler Module allows the user to configure a variety of different schedules from a standardised UI. This is a lot easier than configuring directly in the DB, and can handle most of the simpler cases:

- One Off Schedule
  - Configured to run once at a set date and time
- Periodic Schedule
  - Run every x seconds / minutes / hours starting at set date and time
- Daily Schedule
  - Run at a time or multiple times on one or more days of the week
- Weekly Schedule
  - Run every x weeks at a set time for the selected days of the week

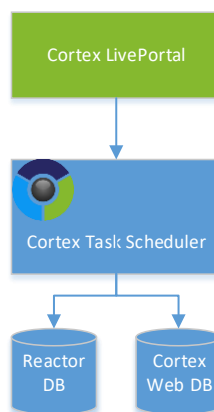
### 1.1 Using the module

Once the .studiopkg file has been imported, the global variable 'G\_SQL\_Server' should be set to an appropriate value. This is the SQL Server containing the Reactor and CortexWeb Databases, although the user is prompted to enter a different value if required in the 1<sup>st</sup> UI Screen.

This allows the flows to be deployed on one system (e.g. Development) while also allowing them to configure schedules on other environments (e.g. UAT or Production), if it is possible to connect to those environments from the server running the flows.

Authentication is handled using the Cortex Database Interface service user. This user must have:

- read permissions to CortexWeb database
- read and write permissions to Reactor database



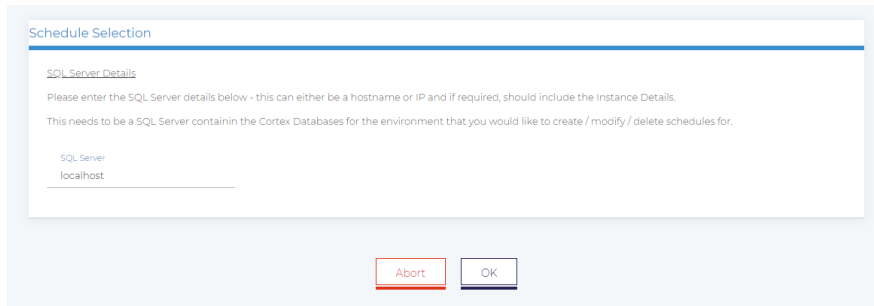
*Figure 2 – CTX-Task-Scheduler Architecture*



## 1.2 User Interface Experience

1. In the 1st UI Screen, the user can select which Database to connect to

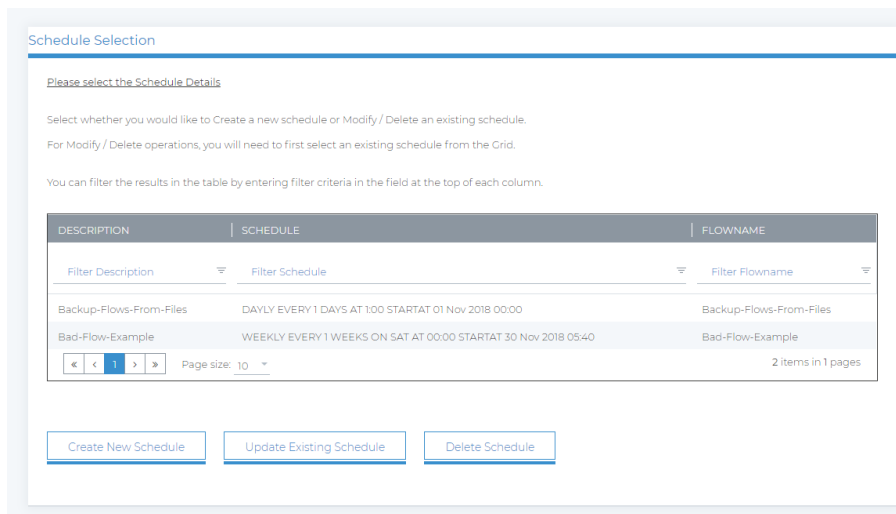
✎ This DB must contain both the Reactor Database and the CortexWeb Database which are added as part of the Cortex Installation process, and this field is pre-populated based on the global variable 'G\_SQL\_Server'.



2. The user is then prompted to select whether to Add, Modify or Delete a schedule

✎ For Modify operations, one existing schedule must be selected

✎ For Delete operations, one or more existing schedule(s) can be selected.



DESCRIPTION	SCHEDULE	FLOWNAME
Filter Description	Filter Schedule	Filter Flowname
Backup-Flows-From-Files	DAVLY EVERY 1 DAYS AT 1:00 STARTAT 01 Nov 2018 00:00	Backup-Flows-From-Files
Bad-Flow-Example	WEEKLY EVERY 1 WEEKS ON SAT AT 00:00 STARTAT 30 Nov 2018 05:40	Bad-Flow-Example

### 3. The user is then prompted to select the required flow for an Add or Modify Schedule

**Schedule - Flow Selection**

Please select one Flow in the table below and click Next

You can filter the results in the table by entering filter criteria in the field at the top of each column.

GROUP_NAME	FLOW_NAME
Filter Group_name	Filter Flow_name
Access User Management	AUM-Authenticate-User
Access User Management	AUM-User-Authentication
Admin-Flows	Backup-Flows-From-Files
Admin-Flows	Cortex-Backup-Flows
Admin-Flows	LP-UI-CWVC-MANAGE
Admin-Flows	VRM-Edit-Config-Parameters
Admin-Flows	VRM-Replication-KeepAlive
Admin-Flows	VRM-Replication-Keep-Alive
Admin-Flows	VRM-TTTTCode-Management
Advanced Training	Bad-Flow-Example

Page size: 10 167 items in 17 pages

Cancel Next

### 4. In the next UI Screen (for Add / Modify) the user is prompted to select the preferred schedule and insert the required values

✎ If a Modify operation is selected, Cortex will parse the existing schedule string and where possible, pre-populate the values

✎ Validation is performed using inline JavaScript.

Weekly **Daily** Periodically Once

**Schedule - Daily**

Create a schedule for flow: **Bad-Flow-Example**

Selected Frequency: **Daily**

For a Daily schedule, multiple times can be entered (if required)

Enter the time(s) required using the dropdowns and button below:

Every: 3 Day(s)

Scheduled Trigger: 12 : 00 Add Time +

**Schedule - Daily - Time Selection**

To remove a time, select the entered time(s) with the checkbox and press Remove Selected Time

When you are finished adding time, click the Next button

☐ 09:00

☐ 12:00

Remove Selected Time

Cancel Continue

**Schedule - Start Date**

Please select the Date/Time for this schedule to be active

Note that this is separate for the time to run the schedule

Start Date \*

30 Nov 2018

Time

08 : 40

Powered by Cortex

Cancel Continue

5. The Initiator and any Parameters can then be set in the next UI Screen (if required)

UPDATE Schedule - Parameters

Initiator  
Please enter the Initiator (text) below - if left blank this will not be added.

SCHEDULED

Flow Parameters  
Please enter the required Parameter Names and Values one by one using the textbox. To add the entered parameter data, use the button:

Parameter Name	Parameter Value

Add Parameter

☐ test-prm: Example Value

Remove Selected Parameter

Cancel Next

6. After this stage, Cortex will perform an operation on the DB to either Insert, Modify or Delete as required. The results of this will be presented to the user

UPDATE Schedule - Results

Flow Schedule:  
DAYLY EVERY 3 DAYS AT 09:00:12:00 STARTAT 30 Nov 2018 05:40

Description / Flow Name:  
Bad-Flow-Example

Old Flow Schedule:  
WEEKLY EVERY 1 WEEKS ON SAT AT 00:00 STARTAT 30 Nov 2018 05:40

Description / Flow Name:  
Bad-Flow-Example

Finish

## 2 Cortex Task Scheduler Flows

---

### 2.1 CTS-Manage-Schedules

#### 2.1.1 Overview

Allows the user to Add / Modify / Delete schedules, including any parameters. This is controlled via LivePortal UIs and performs processing and DB operations.

##### Add Schedule

- Select DB Server to run on
- Select Flow to Schedule
- Select Schedule Type and populate schedule details for the selected schedule type
- Add any Parameters / Initiator details if required
- Insert to the DB

##### Modify Schedule

- Select DB Server to run on
- Select Schedule to Modify
- Select Flow to Schedule
- Select Schedule Type and populate schedule details for the selected schedule type
  - Where possible this will be pre-populated with existing details
- Add / Modify any Parameters / Initiator details if required
- Insert / Update operation to the DB

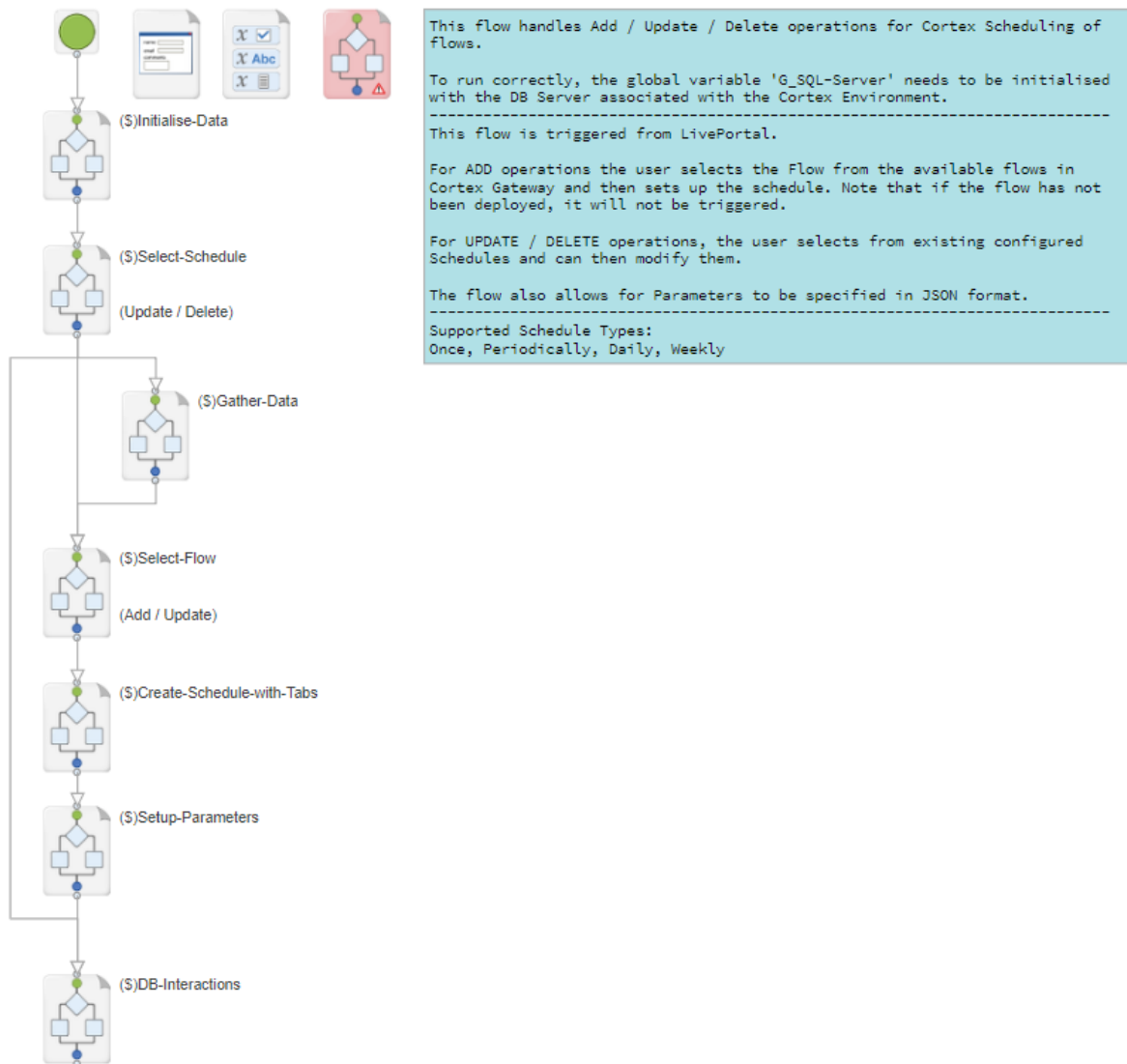
##### Delete Schedule

- Select DB Server to run on
- Select Schedule to Delete
- Delete DB Entries

Exceptions will be raised inline to the User Interface screens where possible (Data Entry Validation).

If any other exceptions occur during the flow, the details will be shown in a separate UI Screen.

### 2.1.2 States



- **Initialise-Data**

Sets a table containing the days of the week to be used later and prompts the user to enter a SQL Server to connect to. Pre-populated based on G\_SQL\_Server variable.

- **Select-Schedule**

Shows a table containing existing schedules on the system and prompts the user to select whether to Add / Modify / Delete. Table selection is validated based on the operation selected (Add does not required selection, modify required 1 to be selected and Delete required 1 or more to be selected).

- **Gather-Data**

This is used for Update operations and will parse the existing schedule details to extract Initiator, Parameters, Flow Name, and all Schedule components. These are used in the next state to pre-populate the relevant controls.

- **Create-Schedule-With-Tabs**

Prompts the user to select the Schedule Type from the tabs at the top. Based on the selected type, the user needs to enter a set of schedule parameters which are used to generate the Schedule text. Validation occurs on each tab based on the selection. If the Schedule Type is not Once, the user will also have to enter the StartAt date (Date and Time for the schedule to be active).

- **Setup-Parameters**

Prompts the user to enter any Parameter data and an Initiator. These are both optional. Parameter entry is restricted to simple Parameter Name / Values and will then be processed into the correct format. If complex Parameters are required, these would have to be entered manually into the DB.

- **DB-Interactions**

Performs the selected operation in the DB and presents results to the user.

### 2.1.3 Inputs

There are no required inputs to the process, however the global variable 'G\_SQL\_Server' should be set to an appropriate value.

### 2.1.4 Outputs

There are no required outputs to the process

## 3 Cortex Task Scheduler Subtasks

### 3.1 CTS-DI-Database-Interaction

#### 3.1.1 Overview

This subtask is used throughout the flow and performs any DB operations. This subtask can support Get / Insert / Update operations.

Exceptions will be raised if

- The query provided cannot be run successfully
- The Database cannot be connected to

#### 3.1.2 Inputs

Input Variables	Type	Description
DI_i_action	Text	Text containing the DB Operation required e.g. Insert Only, Query Value.
DI_i_query	Text	The SQL query to run against the target DB.
DI_i_db_name	Text	The DB Name to connect to.
DI_i_sql_server	Text	The DB Server to connect to.

#### 3.1.3 Outputs

Output Variables	Type	Description
DI_o_table	Table	A table containing the result (if Action is Query Table)
DI_o_out_id	Integer	The ID of the entered item (if Action is Insert Get ID or Update Get ID)
DI_o_out_val	List	A list containing the result (if Action is Query Value, Insert only or Update Only)

### 3.2 CTS-GS-Generate-Schedule

#### 3.2.1 Overview

This subtask generates the schedule string for the user selection in the format the Cortex Reactor DB requires. This Subtask branches based on the Schedule Type. For more complex items (multiple days for a Weekly Schedule or multiple times for a Daily Schedule) this will

also call the CTS-GSC-Generate-Schedule-Components to convert this data into the correct format.

Exceptions will be raised if

- Some of the parameters are missing.

### 3.2.2 Inputs

Input Variables	Type	Description
GS_i_Schedule-Type	Text	The Schedule Type – Once, Periodically, Daily or Weekly
GS_i_Frequency-Num-Txt	Text	How often to run (depending on the selected Schedule Type)
GS_i_Start-At	Text	The StartAt Date and Time – when the schedule becomes active
GS_i_Trigger-Time	Text	The time to trigger the schedule
GS_i_Days-List	List	The days to run (for a Weekly Schedule)
GS_i_Times-List	List	The times to run (for a Daily Schedule)
GS_i_Periodic-Frequency	Text	Frequency (Seconds / Minutes / Hours) for a Periodic Schedule.

### 3.2.3 Outputs

Output Variables	Type	Description
GS_o_Schedule-String	Text	The schedule string to add to the DB.

## 3.3 CTS-GSC-Generate-Schedule-Components

### 3.3.1 Overview

This subtask is called by the CTX-GS-Generate-Schedule subtask and will convert a list of Days or Times to run into the delimited format. This can support 1 or more values for Days or Hours.



### 3.3.2 Inputs

Input Variables	Type	Description
GSC_i_Selection-List	List	The List of either Days or Times to run the schedule on (for Weekly or Daily schedules)
GSC_i_Selection-Type	Text	The Type of list to process – Days or Times.

### 3.3.3 Outputs

Output Variables	Type	Description
GSC_o_Component	Text	The processed text containing Days or Times to add to the Schedule String.

## 3.4 CTS-AUDS-Add-Update-Delete-Schedule

### 3.4.1 Overview

This subtask is called at the end of the process and performs the DB Operations for the schedule operation. This handles Insert / Update / Delete operations. Based on the operation selected, the subtask will generate SQL Statements to perform the operations (for Update operations these will include 'If Exists' statements).

For New and Update operations, some data is passed back to display in the UI

For Delete operations, this subtask is called on a loop to support deleting multiple schedules. in one go.

### 3.4.2 Inputs

Input Variables	Type	Description
AUDS_i_Schedule	Text	The Schedule String generated from the previous subtasks.
AUDS_i_DB-Server	Text	DB Server to perform the operation on.
AUDS_i_DB-Name	Text	DB Name (Reactor)
AUDS_i_Operation	Text	A value containing either New, Update or Delete
AUDS_i_FlowName	Text	The name of the flow to Add or Modify the schedule for. (Add / Modify)
AUDS_i_TEX-ID	Integer	The ID of the existing schedule. (Modify / Delete)
AUDS_i_Params-Text	Text	The string containing all params to enter, in the format structure (<param data>)
AUDS_i_Initiator	Text	The Initiator for the Scheduled Flow.

### 3.4.3 Outputs

Output Variables	Type	Description
AUDS_o_Schedule-Data	Structure	Results of the operation to be shown to the user. (Add / Modify)