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
2.1 Using the module 8
2.2 User Interface Experience 10
3 Cortex Task Scheduler Flows 15
3.1 CTS-Manage-Schedules 15
3.1.1 Overview 15
3.1.2 States 16
3.1.3 Inputs 17
3.1.4 Outputs 17
4 Cortex Task Scheduler Subtasks 18
4.1 CTS-DI-Database-Interaction 18
4.1.1 Overview 18
4.1.2 Inputs 18
4.1.3 Outputs 18
4.2 CTS-GS-Generate-Schedule 18
4.2.1 Overview 18
4.2.2 Inputs 19
4.2.3 Outputs 19
4.3 CTS-GSC-Generate-Schedule-Components 19
4.3.1 Overview 19
4.3.2 Inputs 20

4.3.3 Outputs 20
4.4 CTS-AUDS-Add-Update-Delete-Schedule 20
4.4.1 Overview 20
4.4.2 Inputs 21
4.4.3 Outputs 21
4.5 CTX-Shared-Library : CSL-GAG-Gather-AD-Groups 22
4.5.1 Overview 22
4.5.2 Inputs 22
4.5.3 Outputs 22
4.6 CTX-Shared-Library : CSL-GAG-Check-AD-Groups 23
4.6.1 Inputs 23
4.6.2 Outputs 23

Versions

Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
04/12/2018	1.0	First Release
9/08/2021	1.1	Added AD control for Scheduling. Creation of monthly option.
16/11/2021	1.2	Updated Section 2.1 for using the module Updated Section 2.2 user interface Updated Section 4 for new Subtasks Added a new config flag to enable Startup Scheduling option
30/11/2021	2.0	Monthly option related references (removed) as it is not tested

Module Versions

The following revisions have been made to this module

Date	Revision	Notes
04/12/2018	1.0	Creation of: • CTS-Manage-Schedules • CTS-DI-Database-Interaction • CTS-GS-Generate-Schedule • CTS-GSC-Generate-Schedule-Components • CTS-AUDS-Add-Update-Delete-Schedule
16/11/2021	1.1	Update of: • CTS-Manage-Schedules • CTS-Manage-Schedules • ts_generic_config.txt file
30/11/2021	2.0	Monthly option related references (removed) as it is not tested

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

OCI Orchestration Communication Interface

UI User Interface

DB Database

Requirements

The CTX-Task-Scheduler module requires the following:

- Cortex LivePortal

Integration

Integration with Third-Party Systems

None

Integration with Existing Infrastructure

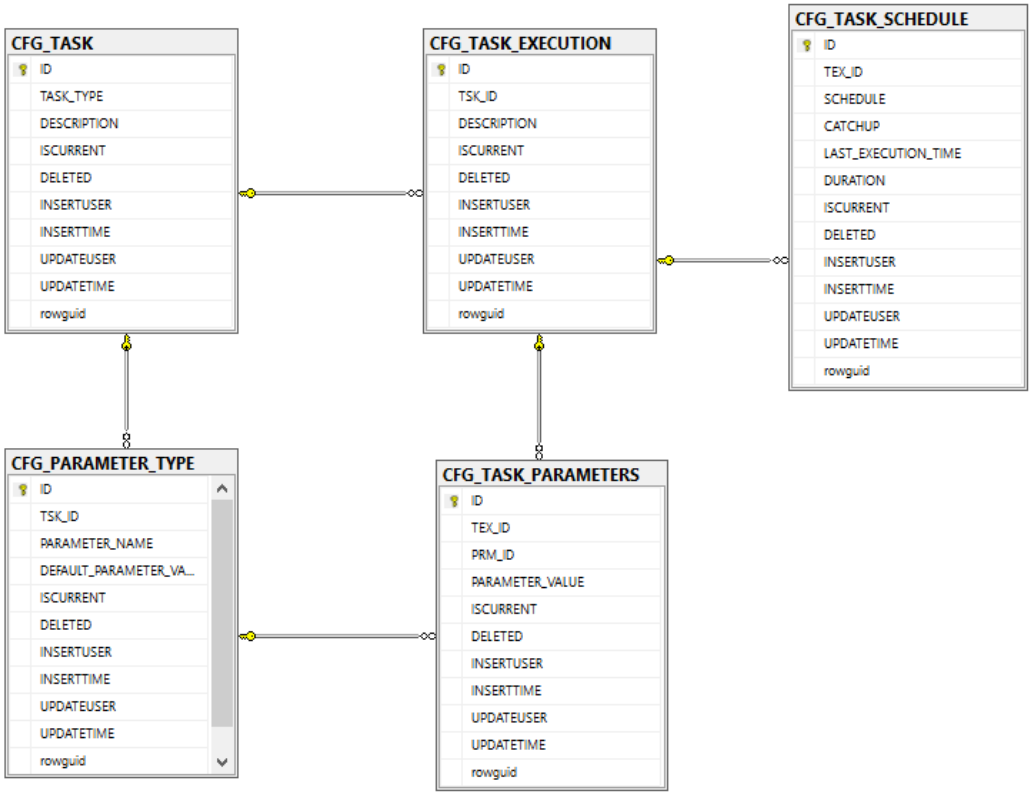
Task Scheduler Database

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.

Task scheduler related Tables in the Reactor database 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

2 The Cortex Task Scheduler Module allows the user to configure a variety of different scheduled flows from a standardised UI. This is a lot easier than configuring directly in the DB, which can be done using the knowledge article titled 'How to schedule a Cortex Flow', 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 a set date and time
- Daily Schedule
  - Run at a specific 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
- Start up Schedule
  - Run a flow with Cortex Server Service Startup

2.1 Using the module

Once the .studiopkg file has been imported and ts_generic_config.txt file is copied to C:\Cortex\, update the ts_generic_config file as per the client needs. See below the parameters that are configured through ts_generic_config file.

S.No	Parameter Name	Description	Default Value if ts_generic_config.txt is not used
1	Show_Db_Server_Selection	DB Server Selection page is presented or skipped based on the parameter value. If set to "true", 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 environment Db	"true"

S.No	Parameter Name	Description	Default Value if ts_generic_config.txt is not used
		servers from the server running the flows. If set to “false”, this UI is not shown to the user. The flow find the Db server that the app server is connected with and uses it for scheduling.	
2	PowerShell Parameters PSDomain PSUsername PSPassword	These parameters needs to be set if AD control for scheduler is required. These parameters are like master flags within the flow. If these parameters are not specified then AD control will not be applied for any scheduling.	None
3	AD Parameters ADUserName ADPassword	If AD control is required, a user with AD Read access is required for querying the AD.	None
4	ADControl_Using _GatewayStudio _Authorisation	If this is set to “true”, then AD control for filtering flows for a user will be based on the Studio Authorisation set for flows in Cortex Gateway.	“false”
5	On_Startup_Schedule_Option	Set this flag to “true” only if the related KB (Patch) for enabling Startup Scheduling option has been deployed in Cortex App Server. Refer to the deployment Plan for the details on the patch.	“false”

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

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

Details of how to changed these permissions are in the deployment guide

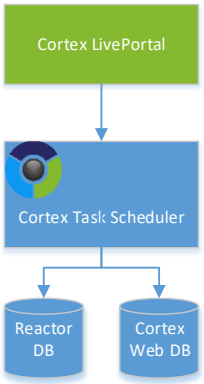
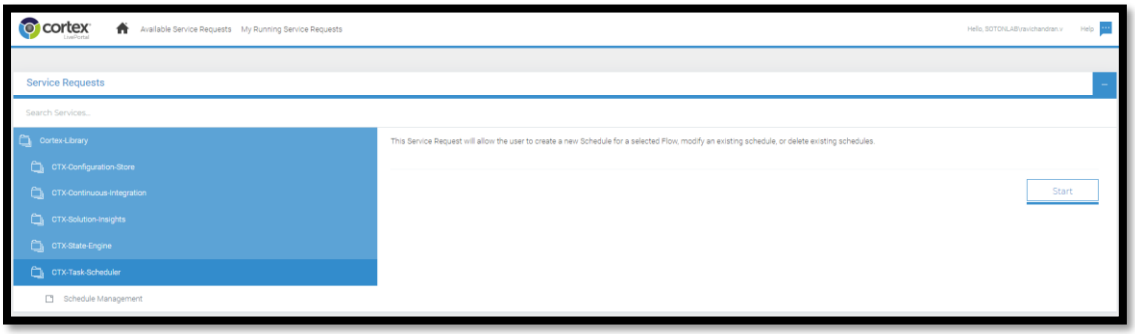


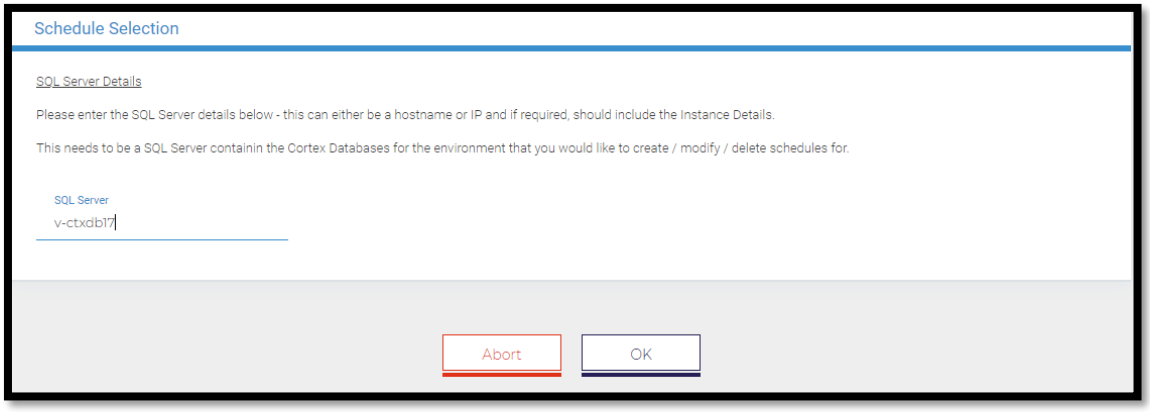
Figure 2 – CTX-Task-Scheduler Architecture

2.2 User Interface Experience

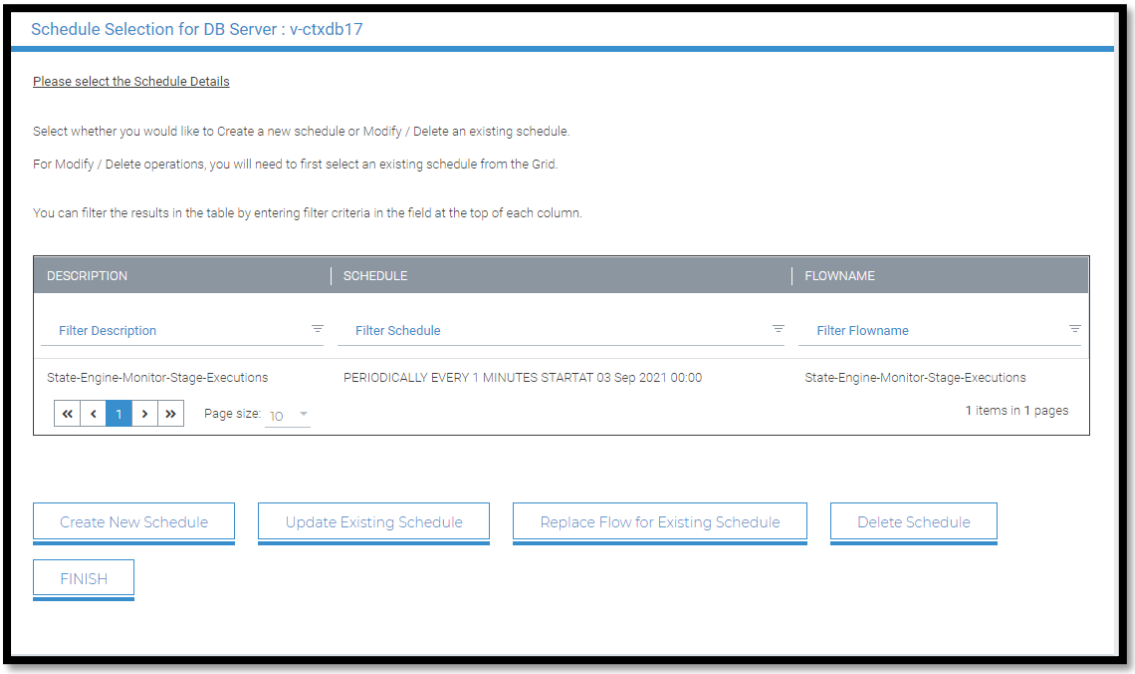
1. Launch the service request from Live Portal UI



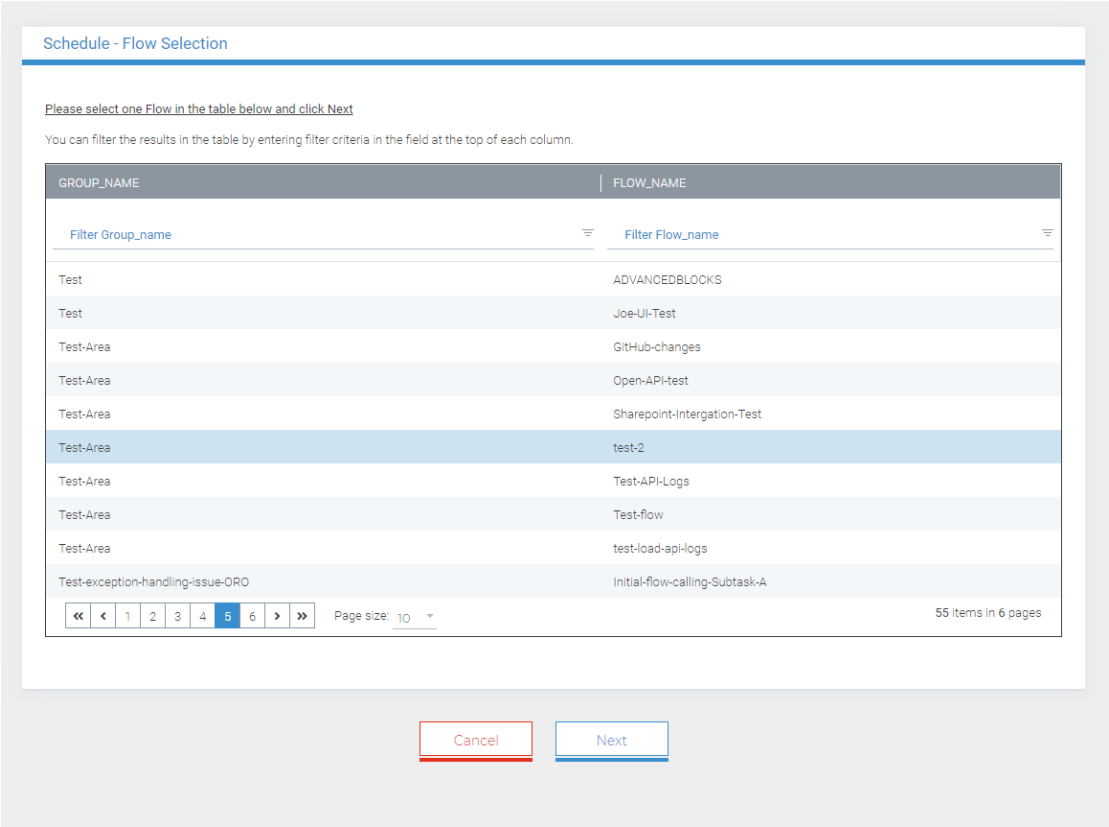
2. In the 1st UI Screen, the user can select which Database to connect to
    - This DB must contain the Reactor Database, he CortexWeb and CortexWeb.Auth Databases which are added as part of the Cortex Installation process.



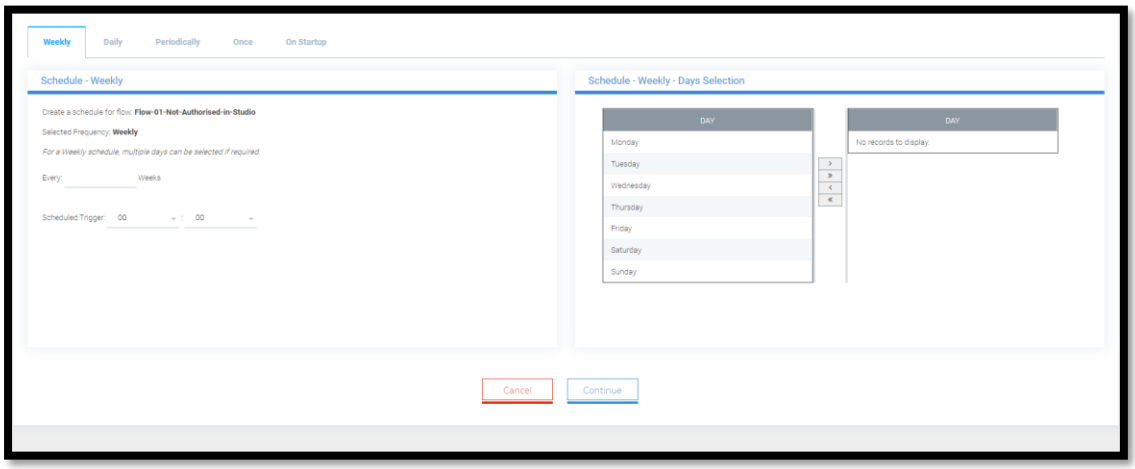
3. The user is then prompted to select whether to Add, Modify or Delete a schedule
   - Update Existing Schedule : Select this option if you wish to update the scheduling parameters for the selected schedule (without changing the flow)
   - Replace Flow for Existing Schedule : Select this option if you wish to update the flow for an existing schedule and / or optionally any other scheduling parameters for the selected schedule.
   - For Delete operations, one or more existing schedule(s) can be selected.

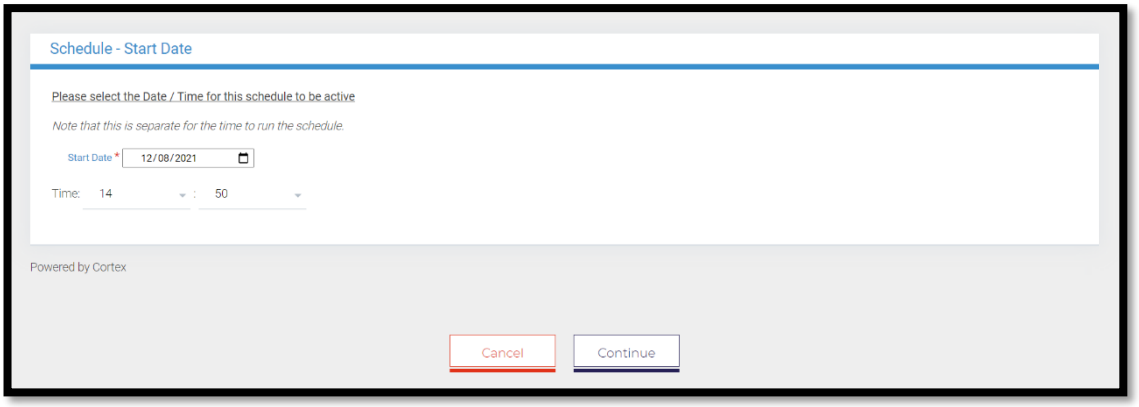


4. The user is then prompted to select the required flow for an Add Or Replace Flow for existing Schedule

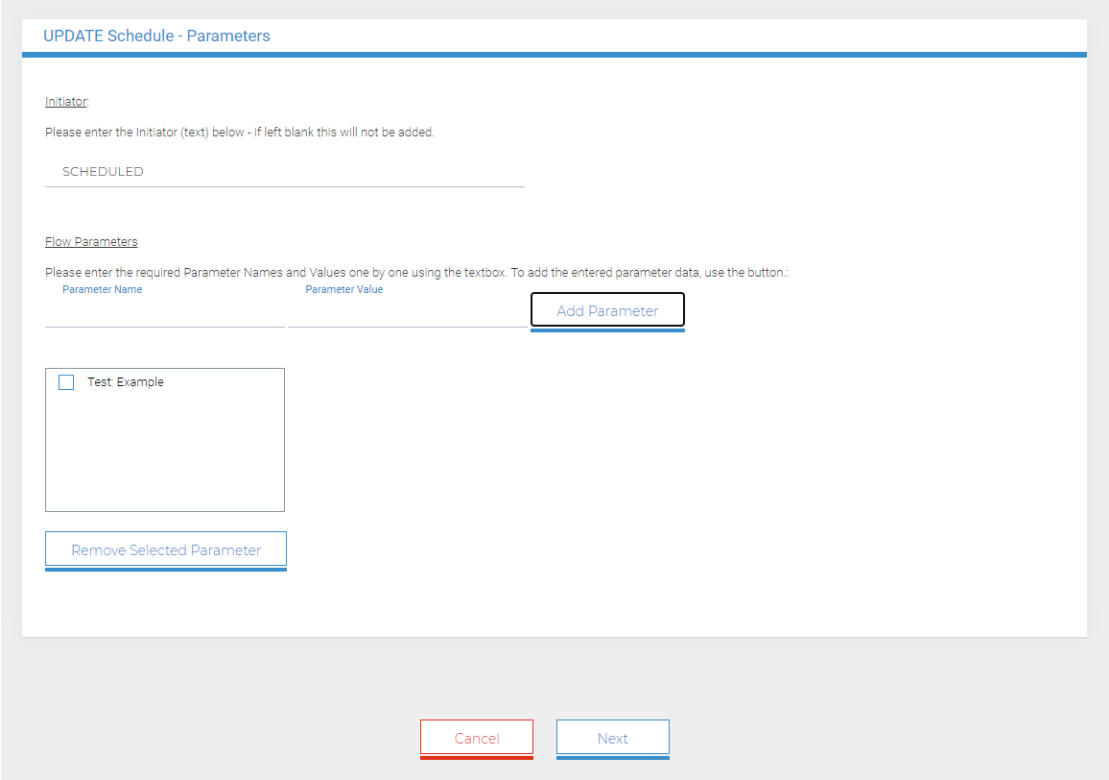


5. In the next UI Screen (for Add / Replace Flow for existing Schedule) the user is prompted to select the preferred schedule and insert the required values
   - If a Modify operation is selected, Cortex will parse the existing schedule string and where possible, pre-populate the values
   - Validation exceptions will be shown to the user, in case user misses to specify any mandatory parameters.

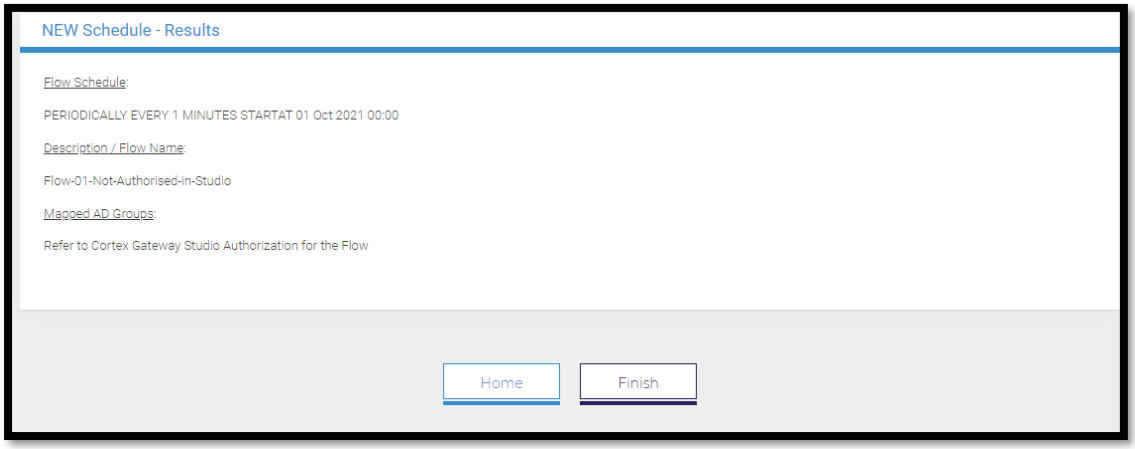




6. The Initiator and any Parameters can then be set in the next UI Screen (if required)



7. 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



3 Cortex Task Scheduler Flows

3.1 CTS-Manage-Schedules

3.1.1 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
    - Option 1: Select schedule parameters
    - Option 2: Replace flow for existing schedules
        - 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
    - Add / Update AD groups, if the AD control is implemented without Gateway Studio Authorization based AD control for scheduler.
- Insert / Update operation to the DB

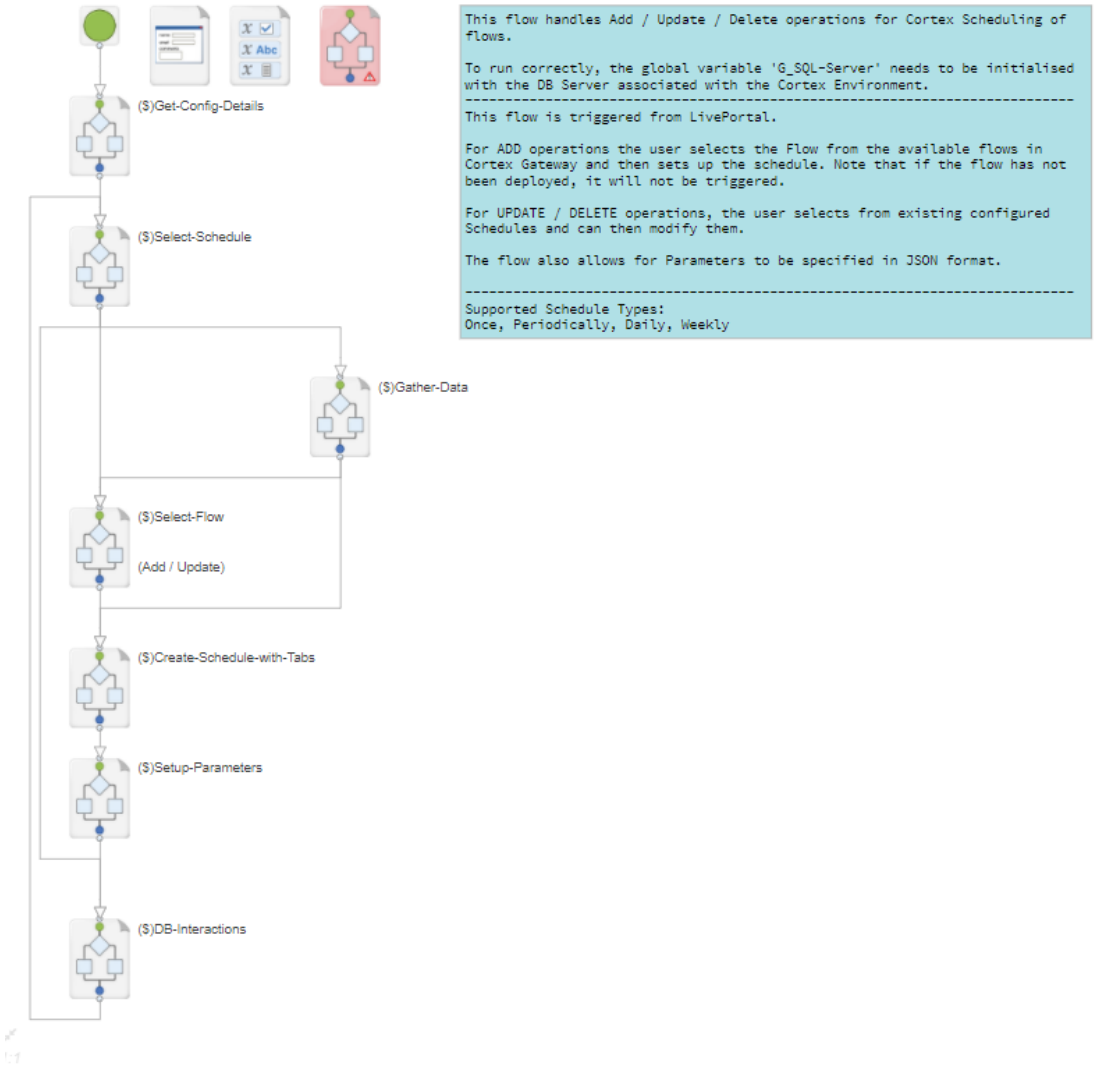
Delete Schedule

- 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.

3.1.2 States



- Get-Config-Details

  Gets configuration parameters from ts_generic_config.txt file if exists. Otherwise, initialises the global config parameters.

  Prompts the user to enter a SQL Server to connect to. Pre-populated based on G2 variable for the connected DB server.

- Select-Schedule

  AD Group membership details for the user is fetched from AD domain. Set of queries are formatted for schedule filtering and flow filtering.

  Sets a table containing the days of the week to be used later are initialized.

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. Complex parameters are any parameters other than text.

- DB-Interactions

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

3.1.3 Inputs

There are no required inputs to the process. However optionally ts_generic_config.txt file be configured.

3.1.4 Outputs

There are no required outputs to the process

4 Cortex Task Scheduler Subtasks

4.1 CTS-DI-Database-Interaction

4.1.1 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

4.1.2 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.

4.1.3 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)

4.2 CTS-GS-Generate-Schedule

4.2.1 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.

4.2.2 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.

4.2.3 Outputs

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

4.3 CTS-GSC-Generate-Schedule-Components

4.3.1 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.

4.3.2 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.

4.3.3 Outputs

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

4.4 CTS-AUDS-Add-Update-Delete-Schedule

4.4.1 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.

4.4.2 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.

4.4.3 Outputs

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

4.5 CTX-Shared-Library : CSL-GAG-Gather-AD-Groups

4.5.1 Overview

This subtask is used to gather all AD Groups a given user belongs to

Exceptions will be raised if

- If Powershell scripts fail
- Or Powershell scripts provide errors

4.5.2 Inputs

Input Variables	Type	Description
GAG_i_ADUserName	Text	Text containing the AD query service user account
GAG_i_ADPassword	Text	Text containing the AD query service user account password
GAG_i_PowerShellDomain	Text	Text containing PowerShell Domain
GAG_i_PSUsername	Text	Text containing the PowerShell block service user account
GAG_i_PSPassword	Text	Text containing the PowerShell block service user account password
GAG_i_UserToCheck	Text	Text containing the input user name

4.5.3 Outputs

Output Variables	Type	Description
GAG_o_Groups	List	List of AD group membership for the input user.

4.6 CTX-Shared-Library : CSL-GAG-Check-AD-Groups

This subtask checks if an input AD group exists

Exceptions will be raised if

- If PowerShell scripts fail
- Or PowerShell scripts provide errors
- OR if the user – AD group doesn't exist

4.6.1 Inputs

Input Variables	Type	Description
CAG_i_ADUserName	Text	Text containing the AD query service user account
CAG_i_ADPassword	Text	Text containing the AD query service user account password
CAG_i_PowerShellDomain	Text	Text containing PowerShell Domain
CAG_i_PSUsername	Text	Text containing the PowerShell block service user account
CAG_i_PSPassword	Text	Text containing the PowerShell block service user account password
CAG_i_GroupToCheck	Text	Text containing the input AD group name

4.6.2 Outputs

Output Variables	Type	Description
CAG_o_GroupIsValid	Boolean	Boolean indicating existence of the input AD Group name

