



CTX-Manual-Intervention User Guide

Cortex Limited
Keypoint
17-23 High Street
Slough
Berkshire
SL1 1DY
United Kingdom
T +44 (0) 1753 513 980
E info@cortex.co.uk

Status: 0.1

Release Date: [Publish Date]

Author: Joe Lees

Document Version: 0.1

Software Version: 0.1

All rights reserved. Passing on and copying of this document, use and communication of its contents are not permitted without written authorisation from Cortex Limited.

Contents

CTX-Manual-Intervention User Guide	1
Contents.....	3
Preface	5
1.1 About this Manual	5
1.2 Audience	5
1.3 Related Material	5
1.4 Abbreviations used in this Document	5
2 Versions.....	6
2.1 Document Revisions	6
2.2 Module Versions.....	6
3 Preface	7
3.1 About this User Guide.....	7
3.2 Audience	7
3.3 Abbreviations.....	7
4 Requirements	8
5 Module Subtasks - UI Actions	9
5.1 MI-WFU-Wait-for-User	14
5.1.1 Overview	14
5.1.2 Input variables	14
5.1.3 Output variables	14
5.2 MI-LUD-Log-User-Data	14
5.2.1 Overview	14
5.2.2 Input variables	14
6 Module Subtasks - Email Actions	16
6.1 MI-WFE-Wait-For-Email	16
6.1.1 Overview	16
6.1.2 Input variables	16
6.1.3 Output variables	16
6.2 MI-LED-Log-Email-Data	16
6.2.1 Overview	16
6.2.2 Input variables	16
6.2.3 Output variables	17
6.3 MI-SHE-Send-HTML-Email.....	18
6.3.1 Overview	18
6.3.2 Input variables	18
6.4 MI-SNE-Search-New-Emails.....	18
6.4.1 Overview	18
6.4.2 Input variables	18
6.5 MI-GEBI-Get-Email-by-ID	18
6.5.1 Overview	18
6.5.2 Input variables	19
6.5.3 Output variables	19

7	Other Module Flows and Subtasks	20
7.1	MI-test-flow.....	20
7.1.1	Overview	20
7.2	MI-UAD-User-Access-Dashboard	20
7.2.1	Overview	20
7.3	MI-GPHG-Generate-Process-HTML-Grid.....	Error! Bookmark not defined.
7.3.1	Overview	Error! Bookmark not defined.
7.3.2	Input variables	Error! Bookmark not defined.
7.3.3	Output variables	Error! Bookmark not defined.
7.4	MI-GSHG-Generate-Stage-HTML-Grid	Error! Bookmark not defined.
7.4.1	Overview	Error! Bookmark not defined.
7.4.2	Input variables	Error! Bookmark not defined.
7.4.3	Output variables	Error! Bookmark not defined.
7.5	MI-GEHG-Generate-Event-HTML-Grid	Error! Bookmark not defined.
7.5.1	Overview	Error! Bookmark not defined.
7.5.2	Input variables	Error! Bookmark not defined.
7.5.3	Output variables	Error! Bookmark not defined.

Preface

1.1 About this Manual

1.2 Audience

1.3 Related Material

1.4 Abbreviations used in this Document

2 Versions

2.1 Document Revisions

The following revisions have been made to this document

Date	Revision	Notes
22/08/2019	0.1	Document automatically generated and populated with additional details
29/08/2019	0.2	Added Configuration Details

2.2 Module Versions

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

Module Version	Release Date	Comments
1.0	22/08/2019	<p>Creation of:</p> <ul style="list-style-type: none">• MI-GEBI-Get-Email-by-ID• MI-GEHG-Generate-Event-HTML-Grid• MI-GPHG-Generate-Process-HTML-Grid• MI-GSHG-Generate-Stage-HTML-Grid• MI-LED-Log-Email-Data• MI-LUD-Log-User-Data• MI-SHE-Send-HTML-Email• MI-SNE-Search-New-Emails• MI-test-flow• MI-UAD-User-Access-Dashboard• MI-WFE-Wait-For-Email• MI-WFU-Wait-for-User

3 Preface

3.1 About this User Guide

The User Guide is designed to give an overview of the CTX-Manual-Intervention module

3.2 Audience

This document is designed for people interested in the CTX-Manual-Intervention module

3.3 Abbreviations

4 Requirements

The CTX-Manual-Intervention module requires the following:

- CTX-Logging Module
 - CTX-Logging Database Deployed on DB Server
- CTX-Configuration Module
 - CTX-Configuration Database Deployed on DB Server

5 Manual Intervention Overview

This module is designed to be used for automated flows which may require a UI action (for example as part of an approval or escalation process) or an Email response. This allows flows to run unattended, but still gives control to the user - meaning an operator can login to the Dashboard flow and view all the processes requiring their attention and deal with them from a centralised UI flow.

This means that a single process using this module would consist of the following components:

- **Process Flow**

This is the main automated flow, which can be triggered via schedule, external call or another trigger mechanism. If the process or branch the process takes requires a UI, the main Process Flow will pause until the User Action has been taken.

This architecture means that the token does not have to wait on a LivePortal UI (with timeout loop) until a user opens the page.

An example (**MI-Test-Flow**) is included with the module.

The External Reference should be set as appropriate at the start of the flow using the 'Set Reference' block. This reference should be added when logging the process and will be used to continue executions when required.

- **Dashboard Flow**

This flow allows users to view the logs associated with all the running processes and trigger any UIs, which open in a new tab.

The 1st UI page is a dashboard which allows users to view different queues such as items requiring a User Action, In Progress processes, or queues based on a particular process.

This flow (**MI-UAD-User-Access-Dashboard**) is included with the module but can be customised.

- **UI Flow(s)**

These are individual flows created for any required User Action. These flows will be triggered from the Dashboard Flow which will pass in any required inputs to render in the UI.

Once a user has taken an action, the UI flow will log the event along with any parameters (such as data entered). The Process Flow will then continue and retrieve the parameters to use.

An example UI flow (**MI-UI-Approval-Flow**) is included with the module.

- **Email Classification Flow(s)**

These are flows created to poll the IMAP Server and check any new mail. If any match a Process in the logging database, the data will be logged.

The Process Flow will then continue and retrieve the Email contents based on the logged email ID.

An example of the flow (**MI-CE-Classify-Emails**) and subtasks (**MI-Search-New-Emails**, **MI-GEI-Get-Email-by-ID** and **MI-SHE-Send-HTML-Email**) are included but will not work out of the box.

5.1 Using the Module

This section details how the module should be used. An example flow (**MI-Test-Flow**) is included with the module for reference, along with an example UI Approval flow.

Email Interaction examples also exist but will not work on customer infrastructure without some initial setup and potential modifications to the IMAP connection method.

5.1.1 Manual Intervention Configuration Items

5.1.1.1 Config-RCI-Retrieve-Config-Items subtask

For a process using Manual Intervention, the Config should be loaded in at the start of the flow and stored in a Global Variable Structure.

And separate UI flows should also retrieve the config in the same way.

The inputs to this subtask are:

- RCI_i_Area
 - The Area which should be 'Manual Intervention'
- RCI_i_Environment (optional)
 - The Environment the config items are stored, if applicable
- RCI_i_Customer (optional)
 - The Customer the config items are stored against, if applicable
- RCI_i_Database-Name
 - The name of the Configuration Database
- RCI_i_SQL-Server
 - The Server hosting the Configuration Database

The output for the subtask is:

- RCI_o_Parameters
 - A structure variable containing the retrieved Configuration Items

Once the config has been retrieved, the process should continue as normal (logging data as it goes).

5.1.2 UI Intervention

The Subtask details are covered in the relevant section under **Module Subtasks - UI Actions**.

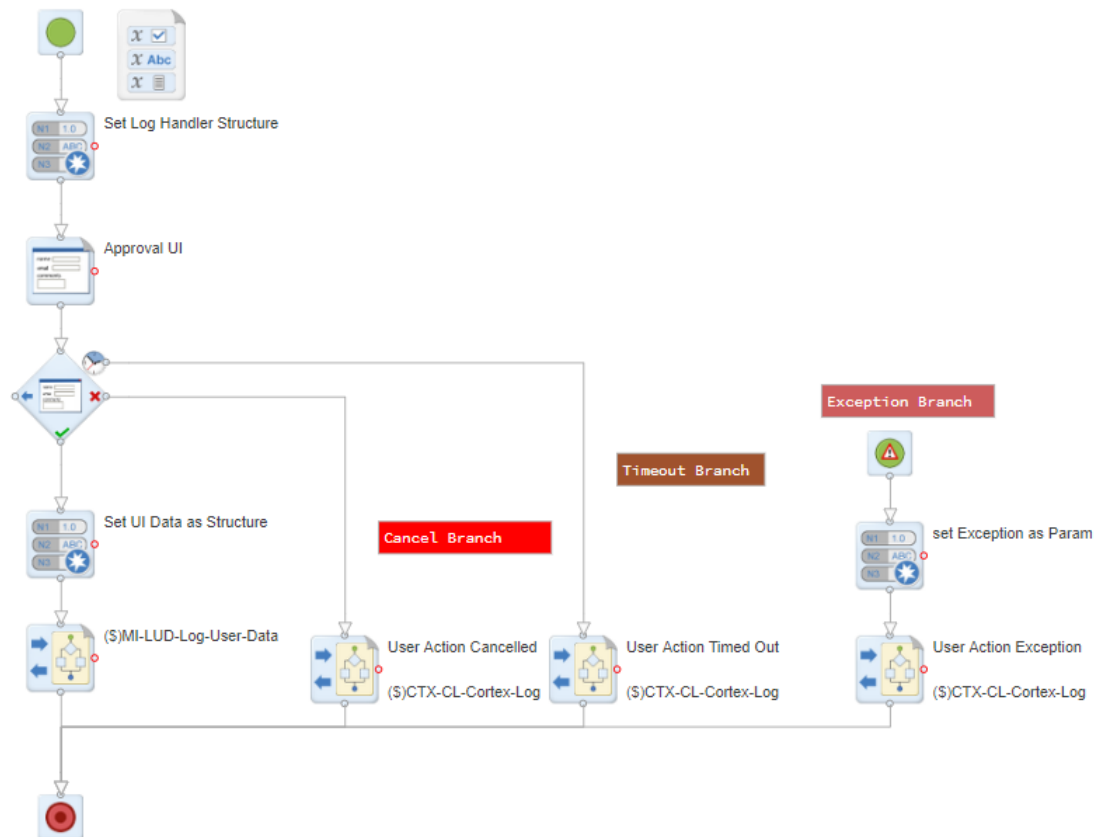
5.1.2.1 MI-WFU-Wait-For-User Subtask

If a User Interaction is required at a stage in the process, the process will pause while calling a subtask which logs the relevant data and waits for the UI to be dealt with.

5.1.2.2 MI-LUD-Log-User-Data Subtask

The separate UI flow will then call a subtask once it is triggered, which logs the data and resumes the original execution.

5.1.3 Configuration of a separate UI Flow



1. (Separate State) Retrieve Manual Intervention Configuration Items and set Connection String
2. Use the Create Structure block to initialise the Logging Structure based on the UI Flow inputs.
3. User Interaction - configured as required.
4. For 'Continue' branch, set the relevant params in a structure and call the MI-LUD-Log-User-Data subtask
5. For Cancel, Timeout, and Exception branch, configure the Logging subtask as required.

5.1.4 Email Intervention

The Subtask details are covered in the relevant section under **Module Subtasks - Email Actions**.

5.1.4.1 MI-WFE-Wait-For-Email

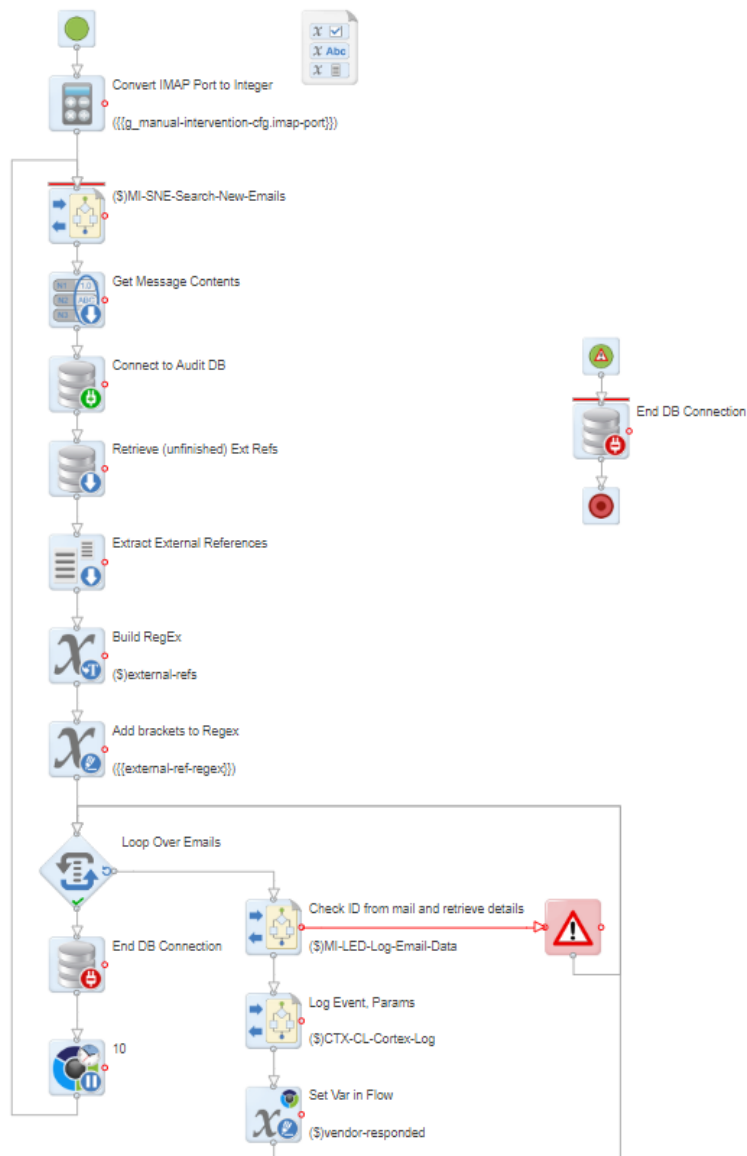
If an Email Interaction is required at a stage in the process, the process will pause while calling a subtask which logs the relevant data and waits for the email to be received. A separate polling flow is required to check for related emails and take the appropriate action.

Once continued, the main process flow will need to extract the Email contents based on the Email ID (which should be logged in the polling flow)

5.1.4.2 MI-LED-Log-Email-Data

The separate polling flow will then call a subtask once a matching email is found, which logs the data and resumes the original execution.

5.1.5 Configuration of an email polling Flow



1. (Separate State) Retrieve Manual Intervention Configuration Items and set Connection String
2. Connect to IMAP Server
3. Retrieve unread emails from relevant folder
4. Loop over each unread email
 - a. Get Subject / Body contents
 - b. Check if any email matches a process in the Logging DB
 - i. Match should preferably be done based on the External Reference of processes
 - c. If so, log data and resume that execution
 - d. Else, mark as unread or process as an exception

6 Module Subtasks - UI Actions

6.1 MI-WFU-Wait-for-User

This subtask should be used in the main process flow when a UI action is required.

6.1.1 Overview

This subtask connects to the Cortex Audit Logging Solution and adds an event to the current stage, marking it as 'Awaiting User Action'.

The execution will then pause until the User Action is complete - this is defined in a separate UI Flow, which is added as a Parameter along with any inputs to the UI Flow.

This is designed to be triggered from a separate portal or dashboard, where any items marked as 'Awaiting User Action' can be launched separately.

The end flow will then log any data as parameters, and then continue this execution - this will in turn retrieve those parameters and output them.

6.1.2 Input variables

Name	Type	Comments
WFU_i_Log-Structure	Structure	The existing log handler so Event Logs are tied to the correct stage.
WFU_i_UI-Flow-Name	Text	UI Flow Name to be called while the original process execution is waiting.
WFU_i_UI-Params	Structure	Any Parameters for the UI Flow ('Attribute: Value').
WFU_i_Connection-String	Text	Connection String for the Cortex Audit Logging DB.

6.1.3 Output variables

Name	Type	Comments
WFU_o_User-Response	Structure	UI data extracted from the DB.
WFU_o_log-str	Structure	Logging DB Handle to be returned to main flow.

6.2 MI-LUD-Log-User-Data

This subtask should be used in the separate UI flows to continue the main flow and log data once the UI action is complete.

6.2.1 Overview

This subtask is designed to be used in the standalone UI Flows as part of the Manual Intervention solution.

It should be used after the User has interacted with the UI, and will mark the Action as completed, log any data, and continue the original execution.

6.2.2 Input variables

Name	Type	Comments
------	------	----------

LUD_i_ui-data-struc	Structure	UI Data entered by the user
LUD_i_process-details	Structure	Details (log, flow name, reference) etc for the Process.
LUD_i_ConnectionString	Text	Cortex Audit Logging DB Connection String.

7 Module Subtasks - Email Actions

7.1 MI-WFE-Wait-For-Email

This subtask should be used in the main process flow when an email action is required.

7.1.1 Overview

This subtask connects to the Cortex Audit Logging Solution and adds an event to the current stage, marking it as waiting for an email.

The execution will then pause until the email is received - this needs to be handled separately, such as a polling email classification flow.

The email flow will then log any data as parameters, such as the Email ID (or contents) and then continue this execution.

7.1.2 Input variables

Name	Type	Comments
WFE_i_connection-string	Text	Connection String for Audit Logging Database.
WFE_i_log-str	Structure	The existing log handler so Event Logs are tied to the correct stage.

7.1.3 Output variables

Name	Type	Comments
WFE_o_Email-Contents	Structure	The contents of the email including Subject, Body, any Attachments etc

7.2 MI-LED-Log-Email-Data

This subtask should be used in the email polling / classification flow to continue the main process and log the email ID once an email is classified against a running process.

7.2.1 Overview

This subtask is designed to be used in a flow which checks emails and classifies them.

The subtask will first check to see if the REFERENCE (stored against Process Executions in the Logging DB) can be found in the email passed in, and if so, it will continue the relevant execution in the main process flow and log the Email ID.

7.2.2 Input variables

Name	Type	Comments
LED_i_Email	Structure	Retrieved email details, e.g. Subject, Body, Attachments etc
LED_i_regex-String	Text	Regex string containing the items to look for in the REFERENCE column

LED_i_db-handle	Structure	The DB Handle for the Logging DB
LED_i_connection-string	Text	Connection String for the Audit Logging DB

7.2.3 Output variables

Name	Type	Comments
LED_o_Log-str	Structure	All the information needed to continue logging against the original process

8 Module Subtasks - Email Examples

8.1 MI-SHE-Send-HTML-Email

8.1.1 Overview

This subtask send an HTML email to the email address(es) provided.

This is done using PowerShell and has been tested with gmail accounts. Other custom subtasks can be created for other Mail Servers or the default Cortex Email block can be used.

8.1.2 Input variables

Name	Type	Comments
shm_i_subject	Text	Subject of the Email.
shm_i_body-html	Text	HTML contents of the Email.
shm_i_to	Text	Comma Separated list of 'To' addresses.
shm_i_smtp-server	Text	SMTP Server to send the mail from.
shm_i_username	Text	Username / From address.
shm_i_password	Text	Password for User.

8.2 MI-SNE-Search-New-Emails

8.2.1 Overview

An example subtask created to search for new emails on an IMAP Server. This can be created as a custom subtask for a customer, but this example uses PowerShell to connect to Gmail.

8.2.2 Input variables

Name	Type	Comments
SNE_i_IMAP-Server	Text	The IMAP Server to connect to
SNE_i_IMAP-Port	Integer	The IMAP Port to connect to
SNE_i_IMAP-Username	Text	Username for the IMAP Server
SNE_i_IMAP-Password	Text	PAssword for the IMAP Server
SNE_i_Folder	Text	Inbox Folder to connect to (default is Inbox)

8.3 MI-GEBI-Get-Email-by-ID

8.3.1 Overview

This is an example subtask which connects to the IMAP Server to retrieve all emails, then returns the email matching the relevant ID.

8.3.2 Input variables

Name	Type	Comments
GEBl_i_Message-ID	Text	Message ID to retrieve
GEBl_i_IMAP-Server	Text	IMAP Server to connect to
GEBl_i_IMAP-Port	Integer	IMAP Port to connect to (default = 993)
GEBl_i_Secure-Bool	Boolean	Boolean specifying whether connection is Secure (default = true)
GEBl_i_Username	Text	Username for IMAP Account
GEBl_i_Password	Text	Password for IMAP Account

8.3.3 Output variables

Name	Type	Comments
GEBl_o_Message	Structure	Structure containing the email data

9 Module Flows

9.1 MI-test-flow

9.1.1 Overview

This flow is an example for how to use the Manual Intervention. Currently it is a barebones flow showing a UI interaction, as the Email will be specific to customer requirements.

This can be extended for more examples.

9.2 MI-UAD-User-Access-Dashboard

9.2.1 Overview

This flow is both useable and an example, and it allows a user to view the processes running in the Audit Logging DB. It shows the status (requires user action / email action) as well as all the logs, and a user can interact with it by clicking a row in the relevant grid to drill down.

Process Logs > Stage Logs -> Event and Param Logs

If a User Actions is required, this can be triggered from the Stage Logs UI or the Event and Param Logs UI, opening up the relevant flow in a new tab.

Queries can be tweaked, or different UIs could be offered to provide access controls and different user queues.

10 Configuration

10.1 Configuration Setup

The CTX-Manual-Intervention Module should be used along with both the CTX-Logging Module and the CTX-Configuration Module. The Configuration Module is used to gather data from a central location, rather than initialise global variables in every flow - this also makes it easier to manage these and change the values easily if required.

The Manual Intervention config params are stored under the area 'Manual-Intervention' and should be loaded into every process flow which uses the Manual Intervention subtasks. There are some core parameters, but more can be added based on the customer requirements.

These can also be stored against different Environments (/ Customers) if required, but at a minimum the Area is required.

10.2 Manual Intervention Configuration Parameters

Below shows an extract from the Config DB (CFG_View), with the Parameter Values removed. The Config Items shown here are the minimum required for the Manual Intervention module.

Area	Customer	Environment	Param_name	Param_value
Manual-Intervention	NULL	NULL	Cortex-URL	
Manual-Intervention	NULL	NULL	Database-Server	
Manual-Intervention	NULL	NULL	Database-Name	
Manual-Intervention	NULL	NULL	IMAP-Password	
Manual-Intervention	NULL	NULL	IMAP-Port	
Manual-Intervention	NULL	NULL	IMAP-Server	
Manual-Intervention	NULL	NULL	IMAP-Username	
Manual-Intervention	NULL	NULL	LivePortal-URL	

Note that Cortex-URL is used to make REST requests when triggering UI flows asynchronously, and LivePortal-URL is the base LivePortal URL used to open the triggered UI in a new tab.

IMAP Details are only required if using the Email Intervention logic, and Database Server / Name is used to generate connection strings used in the Logging Module.

10.3 Example Values

The following examples may help when creating the Configuration Parameters

Param_name	Param_value
Cortex-URL	https://CortexDev.Domain.com
Database-Server	CortexDev.Domain.com
Database-Name	CTX-Logging (other values could include UI-Escalations, Logging)
IMAP-Password	<i>(this should contain the IMAP Password, encrypted using the Cortex Encryption Tool)</i>
IMAP-Port	993 (/ 143)

IMAP-Server	Imap.gmail.com / imap-mail.outlook.com
IMAP-Username	imapUser@email.com
LivePortal-URL	https://CortexDev.Domain.com /liveportal/SequenceHandling.aspx?execution=