Managed Validation Service – Service Description

Our Managed Validation Service is designed for Ultimo customers who may be required to maintain CFR21 Part 11 compliance as part of their business.

The Managed Validation Service is designed to assist Ultimo customers in achieving their validation goals however ensuring CFR21 Part 11 compliance and that validations are passed remains the responsibility of the Customer.

These services shall be classified as Other Services for the purposes of the Agreement. By requesting the Managed Validation Service, the Customer accepts the terms of this Service Description.

1. The Managed Validation Service.

In providing the Managed Validation Service, Ultimo will:

1.1 validate your Ultimo application after every software update to assist with ongoing regulatory adherence.

1.2 test every standard workflow in the Ultimo application to confirm functionality, data integrity, and audit trail reliability—mirroring real-world usage by utilising in-app scenarios.

1.3 produce specific configurations based on documented user stories, along with accompanying scenarios (which shall be created by your authorised partners or internal app managers). These configurations are validated by Ultimo only with each software update, ensuring that validated custom workflows remain compliant and functional with the Ultimo SaaS.

1.4 deliver validation reports after every software update, covering test scope, results, and compliance status giving each customer clear, auditable proof of system readiness.

2. What is required from customers or their designated partners

2.1 Customer or their authorised partner shall produce:

2.1.1 A “user story” which is defined as concise, goal-oriented description of how a user interacts with a SaaS application to achieve a specific outcome. For example:
As a [type of user],
I want to [perform some action],
so that [I can achieve some goal].; and

2.1.2 Am accompanying "scenario" which is defined as a technical test written in xml ensuring the outcome is correct which is further detailed in Annex A.

2.2 Customer acknowledges that Ultimo will not be able to perform the Managed Validation Service unless properly completed user stories and scenarios are provided.

2.3 If Customer chooses to utilise the services of an authorized partner, then the terms of that authorized partner shall apply.

3. Support

Any issues regarding the Managed Validation Service should be addressed to Ultimo Support in the usual way.

4. What is not included by Ultimo:

Creation of user stories or scenarios

Annex A

Introduction

Scenarios are written in XML, and the Workflow Designer is a useful tool for designing, testing, and debugging them. They can be executed within the application through the Workflow Designer and the Scenarios overview in the UCT. This offers a significant advantage over unit testing, as it eliminates the need for Visual Studio or C# knowledge. Additionally, scenarios are executed in the build process just like unit tests. Scenarios consist of several components, but not all are required for every scenario. For each scenario, it is important to determine which components are necessary for successful execution.

Scenario Data

All necessary data for executing a scenario correctly must be added to the **ScenarioData**. The following items can be used to include data:

- **AddTestFile:** Dummy files such as images, XML, XSLT, or text files can be added to a scenario. This is useful for verifying correct image placement, proper attachment linking, or testing XML imports.
- **InsertTestData:** Enables the creation of records and is the most commonly used element in ScenarioData. Each item specifies an entity type and its properties, ensuring the scenario has the correct input data. References to previously created data in ScenarioData can be made using ${TestData.<ItemName>} as the parameter value.
- **SetRelation:** Establishes links between items when direct referencing through InsertTestData is not possible. For example, if Item B references Item A, but Item A also needs to reference Item B, the relationship must be set afterward since Item B does not yet exist when Item A is created.
- **UpdateCompany:** Set parameters related to Company**.** Most used values**:** IndexLongTermAssetPercentage,StartLtapSpaceYearPlan, IndexReplacementPlanning,StartSpacePlanYear.
- **UpdateSettings:** Set parameter values from AET.
- **UpdateProcessSettings:** Set ProcessSettings.
- **UpdateUltimoGlobals:** Overrides values from UltimoGlobals.
- **UpdateUltimoSettings:** Overrides values from UmmSettings (Application settings)
- **UpdateUltimoEnvironment**: Set parameters for the environment mainly used for CurrentDateTime.

ScenarioExecution

In the **Scenario Execution**, you specify the action that needs to be performed. The following actions are available:

- **ExecuteChangeStatus:** Perform a status transition for an item from the **ScenarioData**. This is generally used only when there is no separate workflow that already handles this status transition.
- **ExecuteDelete:** Remove an item from the **ScenarioData**. Use this action to verify whether the business logic (BL) is correctly executed when a deletion is not initiated through a workflow (e.g., removing a row from a grid or deleting a record).
- **ExecuteWorkflow:** The most commonly used action. Specify the workflow number and any required or optional properties for the action you want to execute within the scenario.

ScenarioExpectations

In **Scenario Expectations**, you define the user interactions expected during scenario execution. These can include the following:

- **CollectInputExpectation:** Use this when expecting a **CollectInput** dialog. Specify the dialog name as defined in the workflow and confirm that it is correctly closed by entering **OK** as the **CollectInputResult**. Include an **Expectation Value** with a corresponding input for each requested item in the dialog.
- **DialogExpectation:** Use this when expecting a generic dialog. Specify the dialog name as defined in the workflow and confirm that it is correctly closed by entering **OK** as the **DialogResult**. Include an **Expectation Value** with a corresponding input for each requested item in the dialog.
- **MessageExpectation:** Use this when expecting an informational message that allows the workflow to continue. Provide the message name as defined in the workflow and enter the corresponding message code.
- **QuestionExpectation:** Use this when expecting a question. Set the **QuestionResult** to either **Yes** or **No**, depending on the scenario path you want to test. Provide the question name as defined in the workflow and enter the corresponding message code.
- **TerminationExpectation:** Use this when expecting a validation check. Specify the validation name as defined in the workflow and enter the corresponding message code. The **StopMode** parameter determines whether the entire action should stop or only a part of it. Available options are displayed in the workflow designer.
- **ViewExpectation:** Use this when expecting that, at the end of the scenario, a screen, document, or viewer will open. The available actions are displayed in the workflow designer.

ScenarioValidation

In **Scenario Validation**, you verify whether the scenario results meet the expected outcomes. This can be checked using the following validations:

- **AreEqual:** Check if a value matches the expected result. Use **Expected** to define the expected value and **Actual** to specify the actual value, which is often a reference to an input data property or a predefined value.
- **AreNotEqual:** Check if a value does not match the expected result. Use **NotExpected** to define the unexpected value and **Actual** to specify the actual value, which is often a reference to an input data property or a predefined value.
- **GetCreatedItems:** Use this when the scenario action creates new records and you need to count them or validate specific values per record. Define the entity type under **Type** and specify where the created items should be stored using **OutputProperty**. Unlike workflows, this **OutputProperty** does not need to be pre-declared and can be named freely. The resulting list can then be used for further validations. It is also possible to filter values from this list, such as records that were already present in the input data. If these records belong to the same entity, they will be included in the list unless a **Condition** is provided to filter them out.
- **IsEmpty/IsNotEmpty:** Check if a value is empty or not empty. Use **Value** to specify which field or record should be empty.
- **IsFalse/IsTrue:** Check if a value is **False** or **True**. Use **Condition** to specify which checkbox should be unchecked.
- **TimeIsNow:** Verify if the current date/time is correctly set. Use **Time** to indicate which date/time field should contain the current timestamp.
- **AssetFileExists/AssetFileNotExists**: Check if a file exists or not. Use **FullFileName** to check the path.
- **AssetFolderExists/AssetFolderNotExists:** Check if a folde rexists or not. Use FullDirectoryName to check the path.

ScenarioCleanup

In **ScenarioCleanup**, you can remove data that could not be handled by the automatic cleanup process. This is sometimes necessary when specific data relationships prevent proper cleanup. If the scenario fails due to such issues, the following actions can be used:

- **ClearRelation**
  Remove a reference to another item. Specify the relationship name under **Relation**.
- **DeleteTestData**
  Delete a specific item that was created during the scenario execution. Specify the item under **DomainObject**.
- **DeleteTestDataType**
  Delete all items of a specific entity type. Specify the entity name under **ObjectType**.

Guidelines

When creating scenarios, consider the following guidelines:

- Scenarios must be written in English. This applies not only to the name and description but also to the test data used.
- Scenarios always start with the workflow name, optionally followed by a meaningful description, separated by an underscore (e.g., **ActionField501_AskToContinue#SuccessIfStatusChanged**).
  - After the workflow name, always use a **#**.
  - After the **#**, always include **FailIf** or **SuccessIf**, followed by the subject being tested in the scenario.
  - Do not use sequential numbering in scenario names (e.g., **ActionField001#SuccessIfActionSucceeds1**, **ActionField001#SuccessIfActionSucceeds2**).
- Provide a **Description** in the scenarios to clarify what is being tested.
- Use the comment block if certain parts require additional explanation.
- Scenarios only need to pass on an empty database, not on the demo database.
- Always use the table prefix in lowercase followed by two incrementing numbers for the Ids of the **TestData** blocks. Always use numeric sequential numbers, not alphanumeric ones (e.g., 1a, 1b).
  If the data is related, group it together where possible.
- When using context/status as parameters, place them at the top of the parameter list. First the context, then the status. The Workflow Designer will handle this automatically.
- When using date/time fields, prefer making them dynamic rather than fixed to a specific time. You can add a comment to indicate the intended date (e.g., Today, Tomorrow, Last week). Keep in mind that the tests should also pass around 00:00.
- Always use written-out values for contexts, statuses, booleans, and enumerations instead of numeric values (also known as magical numbers).

Do's

- Follow the naming conventions for scenarios and the items used within them.
- Use dynamic date/time fields wherever applicable.
- Validate all items that are created, modified, or deleted by the scenario, unless other scenarios or tests already verify those aspects.
- Use unique messages for validation purposes.
- Avoid unnecessary TestData.
- Use the Workflow designer to create scenarios to ensure proper alignment, order, and encoding (UTF-8 BOM).
- Ensure that the hash is not included in the scenario when committing to Git.
- There are numerous examples of scenarios available in the Web repository (subfolder Scenarios) or in the Scenario overview within the UCT of each deployment. Use them for inspiration.

Dont's

- Scenarios are not applicable if the business logic (BL) involves:
    - DataEntry dialogs
    - Web dialogs
    - Reservations
    - Web service calls (via UBI or an HTTP request in a workflow)