



VILT OTCS Minium Tests 0.0.3

User Guide

VILT-Group

Version 0.0.3, 2017-05-12

Table of Contents

Overview	1
Pre-requisites	2
Using OTCS Minium Tests in your project	3
At the Development	3
At Continuous Integration (Minium Manager)	6
Modules Tested	7
Tests	9
Steps	11
Main Steps	11
Administration	15
Authentication	20
Category	23
Manage Items	28
Reserve	34
Search	35
Security	40
Version	46
Features	49
Administration	49
Authentication	51
Categories	52
Manage Items	54
Reserve	57
Search	58
Security	60
Version	65
Steps Index	67

Overview

OTCS Minium Tests is a test suite for OpenText Content Server (OTCS).

This document provides the instructions on how to setup a minium project and use the OTCS Minium Tests.

Pre-requisites

To start implementing your tests to OTCS with Minium, download [Minium Developer](#).

After, create your project at Minium Developer. Check [here](#), how to create a project at Minium Developer.

To run the tests using continuous integration, it is required Minium Manager. If you intend to use Minium Manager, please contact VILT (minium.team@vilt-group.com).

Using OTCS Minium Tests in your project

Before start using OTCS Minium Tests, first its required to configure your project.

Also, its required to prepare the development environment and the environment to run your tests at continuous integration.

At the Development

To start using OTCS Minium Tests at Minium Developer, download manually the JAR file and copy it to the lib folder of Minium Developer.

After the JAR file is copied, restart Minium Developer.

Configure Minium project

To configure your Minium project, update the `application.yml` file (at the `src/test/resources/config/` path) of the project, with the following configuration:

```
# Web Drivers
minium:
  webdriver:
    desiredCapabilities:
      browserName: chrome
    window:
      maximized: true

# Configuration that can be injected in your tests
config:
  baseUrl: <otcs_host>
  admin:
    username: <username>
    password: <password>
  administratorPassword: <otcs_administration_password>
  slowMotionSeconds: 0
  filesToUpload:
    path: ${minium.resources.dir}/data/
```

After, replace the following placeholders, at the `application.yml` file:

- **otcs_host**: the host where is running the OTCS
- **username**: the username of the Admin
- **password**: the password of the Admin
- **otcs_administration_password**: the administration password



To upload files to OTCS, create a folder (inside the resources project folder) named `data`. After, copy the files to upload to the `data` folder. Finally, use the [this](#) step to upload the file (replace `<parameter1>` with the filename you want to upload inside the `data` folder).

Import the modules to your project

To import a specific step module, use the following instructions:

```
var mc = require("minium/cucumber");  
mc.loadDefinitions("otcs/classic/<step_module_name>-steps");
```

Where the `step_module_name` is the name of the step module (e.g. "main-steps"). Check [here](#), how to import each step module.

But it's recommended the import of all step modules. To import all step modules, use the following instructions:

```
var mc = require("minium/cucumber");  
mc.loadDefinitions("otcs/classic/steps");
```

Configuration of the world.js file

This is the recommended configuration of the world.js file:

```

var timeUnits = require("minium/timeunits");

var base;
var WebDriverException = Packages.org.openqa.selenium.WebDriverException;

World(function () {

    var staleElementReferenceListener = minium.interactionListeners
        .onStaleElementReference()
        .thenRetry();
    var webdriverExceptionListener = minium.interactionListeners
        .onException()
        .thenRetry();

    browser.configure() ①
        .defaultTimeout(5, timeUnits.SECONDS)
        .defaultInterval(100, timeUnits.MILLISECONDS)
        .waitingPreset("fast")
            .timeout(1, timeUnits.SECONDS)
            .interval(100, timeUnits.MILLISECONDS)
        .done()
        .waitingPreset("very-slow")
            .timeout(1, timeUnits.MINUTES)
            .interval(1, timeUnits.SECONDS)
        .done();

    browser.configure()
        .interactionListeners()
            .add(staleElementReferenceListener)
            .add(webdriverExceptionListener);

    var loading = $("#prgImgContainer").visible();
    var initialized = $(":root").applyWebElements(function () { ②
        if (typeof window.__init === 'undefined') {
            window.__init = false;
            setTimeout(function () { window.__init = true }, 200);
            return $();
        }
        return window.__init ? $(this) : $();
    });

    base = $(":root") ③
        .with(minium.interactionListeners.onTimeout()
            .when(loading)
            .waitForUnexistence(loading)
            .withWaitingPreset("very-slow")
            .thenRetry())
        .with(minium.interactionListeners.slowMotion(config.slowMotionSeconds,
            timeUnits.SECONDS))
        .when(initialized);

}, function () {
    base = undefined;
});

```

- ① Configuration of default, fast and very-slow definitions of timeout exception. Check [here](#), how to create waiting presets.
- ② This function stops the test for 200 milliseconds to allow the facet side bar to load and so the OTCS page be completely loaded.
- ③ This instruction declare the [base expression](#). Here, we declare the function stored at the initialized variable and also deal with the loading dialogs. To deal with the loading dialogs, when a timeout occurs, a interaction listener is triggered, and then checks if an loading element exists in the page. If it exists, it will wait for its unexistence with a provided waiting preset and then it will retry the interaction. Check [here](#), how to handle Timeouts.

At Continuous Integration (Minium Manager)

To start using OTCS Minium Tests at Minium Manager, download manually the JAR file and copy the JAR file to your project (e.g. `${project.basedir}/lib/JARfile.jar`).

After, edit the pom file of your project in order to add the dependency:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
  http://maven.apache.org/xsd/maven-4.0.0.xsd">
  ...
  <dependencies>
    <dependency>
      <groupId>com.vilt-group.otcs-tests</groupId>
      <artifactId>otcs-e2e-tests-pkg</artifactId>
      <version>0.0.1-SNAPSHOT</version>
      <scope>system</scope>
      <systemPath>${project.basedir}/lib/otcs-e2e-tests-pkg.jar</systemPath>
    </dependency>
  </dependencies>
</project>
```

Modules Tested

A list of modules tested for each version of OTCS.

Module	version 10.5.0 SP1	version 16
<i>Content Server</i>	✓	✓
<i>Attribute Extensions</i>	✓	✓
<i>Classifications</i>	✓	✓
<i>eLink</i>	✓	✓
<i>Document Properties Synchronization</i>	✓	✓
<i>Monitoring Agent</i>	✓	✓
<i>Physical Objects</i>	✓	✓
<i>Records Management</i>	✓	✓
<i>Search</i>	✓	✓
<i>Security Clearance</i>	✓	✓
<i>WebDAV</i>	✓	✓
<i>WebReports Workflow Extensions</i>	✓	✓
<i>Workflow</i>	✓	✓
<i>Content Server Pulse</i>		✓
<i>Connected Workspaces</i>		
<i>Content Suite Viewer</i>		✓

Module	version 10.5.0 SP1	version 16
<i>Forms and WebForms</i>		✓
<i>Enterprise Connect</i>		✓
<i>Recycle Bin</i>		✓
<i>Renditions</i>		✓
<i>Template Workspaces</i>		

Tests

A list of tests to validate the OTCS functionalities.

- **Administration:**
 - Manage user:
 - List users
 - Create/Delete users
 - Edit user
 - View user
 - View groups of an user
 - Manage group:
 - Create/Delete a groups
 - Add/remove users to the group
- **Authentication:**
 - Login with wrong credentials
 - Login with correct credentials
 - Content server administration login with wrong credentials
 - Content server administration login with correct credentials
 - Logout
- **Categories:**
 - Create/Delete category
 - Add Attribute to category
 - Add category to a folder
 - Propagate category changes to a folder
- **Manage Items:**
 - Create/Delete
 - Move
 - Copy
 - Shortcut
 - zip & download
 - Rate
 - Add/Remove from Favorite
 - Upload
 - Edit a text document
 - Rename
- **Reserve:**

- Reserve a document
- Cannot delete a reserved document
- Unreserve a document
- **Search:**
 - Search for a specific document
 - Advanced search
 - Faceted browsing
 - Save searches
 - Retrieve a saved a search
 - Find similar
- **Security:**
 - Grant/remove access (See, Modify, edit,...) to a document
 - Setting a security clearance on a folder and then trying to add a document
 - Give permissions to group and test if user (that belongs to the group) has permissions
 - Test if the permissions are propagated to sub-items
 - Change the owner user/group
- **Version:**
 - Add a version to a document
 - Delete a version from document
 - Change the description of a versions for a selected document
 - Download a version (TODO)
 - Lock version
 - Try to delete a locked version
 - Cannot delete a document with a locked version
 - Unlock version
 - Purge versions

Steps

This chapter describes the step definitions present at the OTCS Minium Tests.

Foreach step, you will find a description, the assumption (optional) to describe the context that the step can be used and an example.

Main Steps

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/main-steps");
```

I'm at the main page

Description

This step loads the URL defined at the `config.baseUrl` configuration at the `application.yml` file.

Example

The following example loads the `baseUrl` and open the Content Server Administration page:

```
Given I'm at the main page
When I navigate to "Admin > Content Server Administration"
```

I'm logged in with Admin

Description

This step log in the admin user, based on the `config.admin.username` and `config.admin.password` configurations at the `application.yml` file.

Example

The following example makes sure that the user Admin is logged in and navigates to the "Content Server Administration" page:

```
Given I'm logged in with Admin
When I navigate to "Admin > Content Server Administration"
```



This step check the OTCS version. This routine is important to run the tests according to the OTCS version. Also, if any user (other than the Admin user) is logged in, the execution of this step, log out that user and log in the Admin user. If no user is logged in, then log in the Admin user. Finally, the user will be redirected to the `config.baseUrl`.

I click the button "<parameter1>"

Description

This step clicks in button named `<parameter1>`.

Assumption

The button named `<parameter1>` exists in the current page.

Example

The following example enters the Admin credentials and click at the button "Sign in" to submit the log in credentials:

When I enter the correct administrator credentials
And I click the button "Sign in"

*I navigate to "<parameter1>"***Description**

This step navigates to a page defined by the path `<parameter1>`.

Assumption

The navigation bar must be visible.

Example

The following example log in as Admin. After the login, it navigates to the "Content Server Administration" page:

Given I'm logged in with Admin
When I navigate to "Admin > Content Server Administration"

*I accept the operation***Description**

This step confirm the operation (Click OK at the alert box).

Assumption

The alert box of the browser must be opened.

Example

The following example clicks at the "Purge Versions" button to delete one or more versions of a document. This will trigger a alert box to confirm the operation and we click in the "OK" button at the alert box to confirm the operation:

When I click the button "Purge Versions"
And I accept the operation

*I open the folder "<parameter1>"***Description**

This step opens a folder defined at the `<parameter1>`.

Assumption

The folder must exists at the browser table.

Example 1

The following example navigates to the Enterprise Workspace and open the folder "Minium Tests". After, the folder "Minium Tests" is opened, it opens the folder "New folder". And, after the folder "New folder" is opened, it opens the folder "New folder (2)":

When I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests > New folder > New folder (2)"

Example 2

The following example navigates to the Enterprise Workspace and open the folder "Minium Tests":

When I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests"



The <parameter1> could contain a path with the complete tree of the folder that we want to open. To open a folder one level deep, just define the name of the folder only (e.g. I open the folder "Minium Tests"). To open a folder several levels deep, create a path with the ">" string as separator (e.g. I open the folder "Minium Tests > New folder > New folder (2)").

I open the item "<parameter1>"

Description

This step opens an item defined at the <parameter1>.

Assumption

The item must exists at the browser table.

Example

The following example navigates to the Enterprise Workspace and open the item "document.txt".

When I navigate to "Enterprise > Workspace"

And I open the item "document.txt"

I (cannot)? see the item "<parameter1>"

Description

This step check if an item named <parameter1> exists (or not) in the browser table.

Assumption

The browser table must be opened at the correct path.

Example 1

The following example opens the folder "Test Move" and validates that the item named "test.docx" exists:

Then I open the folder "Test Move"

And I see the item "test.docx"

Example 2

The following example confirms the deletion of the item "Minium Tests" and validates that the item "Minium Tests" no longer exists:

And I confirm the deletion
Then I cannot see the item "Minium Tests"

*I select the item "<parameter1>"***Description**

This step selects an item named <parameter1>.

Assumption

The browser table must be opened at the correct path with an item named <parameter1> visible.

Example

The following example navigates to the Enterprise Workspace and selects the item named "Minium Tests":

When I navigate to "Enterprise > Workspace"
And I select the item "Minium Tests"

*I perform the action "<parameter1>"***Description**

This step performs an action (Copy, Move, Delete, ...) to an item named <parameter1>.

Assumption

The browser table must be opened at the correct path with an item named <parameter1> visible.

Example

The following example selects the item named "Minium Tests" and perform the Delete action which will delete the item:

When I select the item "Minium Tests"
And I perform the action "Delete"

*I confirm the deletion***Description**

This step will confirm the deletion. It will respond to the confirmation dialogs before the deletion.

Assumption

The confirmation dialog that leads to the delete action of the item must be opened.

Example

The following example perform the Delete action and confirm the deletion:

When I perform the action "Delete"
And I confirm the deletion

I create the folder "<parameter1>" if not exists

Description

This step will create a folder, with a name <parameter1>, if the folder not exists.

Assumption

The browser table must be opened at the correct path.

Example

The following example navigates to the Enterprise Workspace and create the folder "Minium Tests", if the folder not exists:

And I navigate to "Enterprise > Workspace"
And I create the folder "Minium Tests" if not exists



This step goes through all process of creation of a folder, if this folder doesn't exists. First check if the folder exists. If not, opens the creation form to sets the name of the folder and click at the "Add" button. After the folder is created, the test continues at the browser table where the folder was created.

Administration

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/administration-steps");
```

I should(not)? see the (user|group)? "<parameter1>"

Description

This step validates that the user or group named <parameter1> exists (or not), in the system.

Assumption

The "Users & Groups" page must be opened.

Example 1

The following example navigates to the "Users & Groups" page and validates that the user "Admin" exists:

When I navigate to "Enterprise > Users & Groups"
Then I should see the user "Admin"

Example 2

The following example deletes the group "group1", navigates to the "Users & Groups" page and validates that the group "group1" don't exist:

When I delete the group "group1"
Then I navigate to "Enterprise > Users & Groups"
And I should not see the group "group1"



At the "Users & Groups" page, this step uses the filter to search for the user or group. After the filter, we check if the element exists in the list.

I delete if the (user|group)? "<parameter1>" exists

Description

This step deletes an user or a group named <parameter1> if exists.

Assumption

The "Users & Groups" page must be opened.

Example 1

The following example navigates to the "Users & Groups" page and deletes the user "user1" if exists:

When I navigate to "Enterprise > Users & Groups"
And I delete if the user "user1" exists

Example 2

The following example navigates to the "Users & Groups" page and deletes the group "group1" if exists:

When I navigate to "Enterprise > Users & Groups"
And I delete if the group "group1" exists



At the "Users & Groups" page, this step check if the element exists in the list. If exists, it opens the edit page for the element and deletes the element. Otherwise, do nothing.

At "Add Item", I select "(User|Group)"

Description

This step opens the create form for the "User" or the "Group".

Assumption

The "Users & Groups" page must be opened.

Example

The following example navigates to the "Users & Groups" page and opens the create form for the "Group":

When I navigate to "Enterprise > Users & Groups"
And At "Add Item", I select "Group"

I fill the "Add New User" form with:

Description

This step fills the user create form.

Assumption

The user create form page must be opened.

Example

The following example opens the create form for the "User" and fills several fields in order to create a new user:

```
When At "Add Item", I select "User"
And I fill the "Add New User" form with:
| Log-in Name | <user_name> |
| Department | DefaultGroup |
| Password | 123456 |
| Verify Password | 123456 |
```

*I edit the (user|group)? "<parameter1>"***Description**

This step opens the form to edit an user or a group named <parameter1>.

Assumption

The "Users & Groups" page must be opened.

Example

The following example navigates to the "Users & Groups" page and opens the edit form for the user "user1":

```
When I navigate to "Enterprise > Users & Groups"
And I edit the user "user1"
```

*I view the (user|group)? "<parameter1>"***Description**

This step opens a popup with the information of an user or a group named <parameter1>.

Assumption

The "Users & Groups" page must be opened.

Example

The following example navigates to the "Users & Groups" page and opens the popup with the information of the user "user1":

```
When I navigate to "Enterprise > Users & Groups"
And I view the user "user1"
```

*I should see the "<parameter1>" with the value "<parameter2>" at the popup "<parameter3>"***Description**

This step validates that the popup (named <parameter3>) with the information of an user or a group, has one field (named <parameter1>) with the value <parameter2>. After the validation, this step closes the popup.

Assumption

The popup with the information of an user or group must be opened.

Example

The following example opens the popup with the information of the user "user1" and validates that the user "user1" has one property named "Title" with the value "Minium User":

```
When I view the user "user1"
Then I should see the "Title" with the value "Minium User" at the popup
"user1"
```

*I search for the users that startsWith "<parameter1>"***Description**

This step search for the users that startsWith <parameter1>.

Assumption

The edit form of a group must be opened.

Example

The following example opens the edition form for the group "group1" and search for the users that startsWith "user":

```
When I edit the group "group1"
And I search for the users that startsWith "user"
```

*I select the user "<parameter1>" to Add to group***Description**

This step selects an user named <parameter1> to add to a group.

Assumption

The edit form of a group must be opened and the user must exist (and must be visible).

Example

The following example opens the edit form for the group "group1" and search for the users that startsWith "user". After, selects the user "user1" from the result list:

```
When I edit the group "group1"
And I search for the users that startsWith "user"
And I select the user "user1" to Add to group
```

*I submit the selection of the users***Description**

This step submits the selection of users to add to a group.

Assumption

The edit form of a group must be opened and the users must be selected.

Example

The following example selects the user "user1" from the result list and submit the selection, in order to add the users to a group:

When I select the user "user1" to Add to group
And I submit the selection of the users

*I open the groups of the user "<parameter1>"***Description**

This step opens the list of groups for an user named <parameter1>.

Assumption

The "Users & Groups" page must be opened.

Example

The following example navigates to the "Users & Groups" page and opens the list of groups for the user "user1":

Then I navigate to "Enterprise > Users & Groups"
And I open the groups of the user "user1"

*The user(not)? belongs to the group "<parameter1>"***Description**

This step validates that an user belongs (or not) to a group named <parameter1>.

Assumption

The list of groups for the user must be opened.

Example

The following example opens the list of groups for the user "user1" and validates that the "user1" don't belong to the group "group1":

Then I open the groups of the user "user1"
And The user not belongs to the group "group1"

*I remove the user "<parameter1>" from the group***Description**

This step removes an user named <parameter1> from a group.

Assumption

The edit form of a group must be opened and the user must belong to the group.

Example

The following example opens the edit form for the group "group1" and removes the user "user1" from the group:

When I edit the group "group1"
And I remove the user "user1" from the group

I delete the (user|group)? "<parameter1>"

Description

This step deletes an user or a group named <parameter1> from the system.

Assumption

The "Users & Groups" page must be opened.

Example

The following example navigates to the "Users & Groups" page and opens the edit form for the user "user1" to delete the user:

When I navigate to "Enterprise > Users & Groups"
And I delete the user "user1"

Authentication

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/authentication-steps");
```

I'm at the login page

Description

This step loads the URL defined at the `config.baseUrl` configuration at the `application.yml` file.

Example

The following example loads the login page and enters the login credentials of the admin user, defined at `application.yml`:

Given I'm at the login page
When I enter the correct administrator credentials



This step check the OTCS version. This routine is important to run the tests according to the OTCS version. Also, if any user is logged in, the execution of this step, log out of that user.

I enter the (correct|wrong)? administrator credentials

Description

This step types the correct (or wrong) username and password at the login form.

Assumption

The login page must be opened.

Example 1

The following example loads the login page and enters the login credentials of the admin user, defined at `application.yml`:

Given I'm at the login page

When I enter the correct administrator credentials

Example 2

The following example loads the login page and enters the wrong login credentials:

Given I'm at the login page

When I enter the wrong administrator credentials

I should see a warning message at signin

Description

This step validates that we tried to login with the wrong credentials by checking if we have the invalid credentials warning message at the login form.

Assumption

The wrong credentials were typed and the login form was submitted.

Example

The following example enters the wrong login credentials, submit the login form and validates that the wrong login credentials were typed:

When I enter the wrong administrator credentials

And I click the button "Sign in"

Then I should see a warning message at signin

I should be redirected to the main page

Description

This step validates that we were redirected to the main page after enter the correct login credentials.

Assumption

The correct credentials were typed and the login form was submitted.

Example

The following example enters the correct login credentials, submits the login form and validates that we went redirected to the main page:

When I enter the correct administrator credentials

And I click the button "Sign in"

Then I should be redirected to the main page

I should be asked to type the Content Server administration password

Description

This step validates that when we try to access an Administration page for the first time, we will be asked to type the administration password.

Assumption

Access an Administration page and we didn't type the correct administration password before.

Example

The following example navigates to the "Content Server Administration" page and validates that we were asked to type the administration password:

When I navigate to "Admin > Content Server Administration"

Then I should be asked to type the Content Server administration password

*I enter the (correct|wrong)? administration password***Description**

This step types the correct (or wrong) administration password.

Assumption

Access an Administration page and we didn't type the administration password before.

Example 1

The following example navigates to the "Content Server Administration" page and type the wrong administration password:

When I navigate to "Admin > Content Server Administration"

And I enter the wrong administration password

Example 2

The following example navigates to the "Content Server Administration" page and type the correct administration password:

When I navigate to "Admin > Content Server Administration"

And I enter the correct administration password

*I should see an authentication error***Description**

This step validates that we type the wrong administration password by checking if we have the invalid password warning message.

Assumption

The wrong administration password were typed and the form was submitted.

Example

The following example enters the wrong administration password, submits the form and validates that the wrong administration password were typed:

When I enter the wrong administration password

And I click the button "Log-in"

Then I should see an authentication error

I should see the Content Server Administration page

Description

This step validates that we were redirected to the Content Server Administration page after we enter the correct administration password.

Assumption

The correct administration password were typed and the form was submitted.

Example

The following example enters the correct administration, submits the form and validates that we are at the Content Server Administration page:

When I enter the wrong administration password
And I click the button "Log-in"
Then I should see the Content Server Administration page

I logout

Description

This step logout the current logged in user.

Assumption

An user must be logged in at OTCS.

Example

The following example makes sure that the user Admin is logged in and then logouts the Admin user from OTCS:

Given I'm logged in with Admin
When I logout

I'm no longer authenticated

Description

This step checks if the login form exists.

Assumption

No user is logged in at OTCS.

Example

The following example logouts the current user from OTCS and checks if the login form exists:

When I logout
Then I'm no longer authenticated

Category

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/category-steps");
```

Type the Content Server administration password

Description

This step types the administration password, if we are asked to type the administration password.

Assumption

Access an Administration page.

Example

The following example navigates to the "Content Server Administration" page and types the administration password, if we are asked to type the administration password:

```
When I navigate to "Admin > Content Server Administration"
And Type the Content Server administration password
```

Select "<parameter1>"

Description

This step opens a page named <parameter1> at the Content Server Administrator.

Assumption

The "Content Server Administrator" page must be opened.

Example

The following example navigates to the "Content Server Administration" page and opens the page named "Open the Categories Volume":

```
When I navigate to "Admin > Content Server Administration"
And Select "Open the Categories Volume"
```

I open "<parameter1>" with the type "<parameter2>"

Description

This step opens an item named <parameter1> that has the type <parameter2>.

Assumption

The browser table must be opened at the correct path with an item named <parameter1> that has the type <parameter2>.

Example

The following example opens the page named "Open the Categories Volume" and opens an item named "test-category" that has the type "Category":

```
When Select "Open the Categories Volume"
And I open "test-category" with the type "Category"
```

I choose to add a new "<parameter1>" Attribute

Description

This step opens the create form to a new attribute with the type <parameter1> to the category.

Assumption

The category page must be opened.

Example

The following example opens an item named "test-category" that has the type "Category" and open the create form of a new attribute with the type "Text: Field" to the category:

```
When I open "test-category" with the type "Category"
And I choose to add a new "Text: Field" Attribute
```

I open the category "<parameter1>" and see the attribute "<parameter2>"

Description

This step opens a category named <parameter1> and checks if the category contains the attribute <parameter2>.

Assumption

The category <parameter1> page must be opened.

Example

The following example opens an item named "test-category" that has the type "Category", opens the create form of a new attribute with the type "Text: Field", sets the name "test-attribute" to the new attribute and confirms the create. Finally, validates that the "test-category" contains the attribute "test-attribute":

```
When I open "test-category" with the type "Category"
And I choose to add a new "Text: Field" Attribute
And I fill the field "Name" with "test-attribute"
And I click the button "OK"
And I click the button "Submit"
Then I open the category "test-category" and see the attribute "test-attribute"
```

I add the category "<parameter1>" to the item

Description

This step adds a category named <parameter1> to an item.

Assumption

The categories of the item page must be opened.

Example

The following example opens the categories of the item "Folder with Category" and adds the "test-category" category to the "Folder with Category":

When I open the "Categories" function for the item "Folder with Category"
And I add the category "test-category" to the item



This step checks if the item contains the category before adds it. If the item don't have the category, it open the "Add Categories" popup and select the category defined at the <parameter1>. Finally, submits the operation and we will be redirected to the browser table where the item is.

I see that the item "<parameter1>" has the category "<parameter2>"

Description

This step opens the categories of the item named <parameter1> and checks if the item contains the category <parameter2>.

Assumption

The browser table must be opened at the correct path with an item named <parameter1> visible.

Example

The following example adds the "test-category" category to the item and check if the "Folder with Category" contains the category "test-category":

When I open the "Categories" function for the item "Folder with Category"
And I add the category "test-category" to the item
Then I see that the item "Folder with Category" has the category "test-category"

I apply the categories to Sub-Items of the folder "<parameter1>"

Description

This step applies the categories of an folder named <parameter1> to its sub-items.

Assumption

The categories of the folder page must be opened.

Example

The following example opens the categories of the item "Folder with Category" and applies the categories of the folder "Folder with Category" to all its sub-items:

When I open the "Categories" function for the item "Folder with Category"
And I apply the categories to Sub-Items of the folder "Folder with Category"

I remove the category "<parameter1>" of the item "<parameter2>"

Description

This step opens the categories of the item named <parameter2> and removes the category named <parameter1>.

Assumption

The browser table must be opened at the correct path with an item named <parameter2> visible.

Example

The following example opens the folder "Minium Tests > Folder with Category", opens the categories of the item "Document_with_Category.txt", selects the "test-category" and click at remove. Finally, click in the "OK" button at the alert box to confirm the action.

```
When I open the folder "Minium Tests > Folder with Category"
And I remove the category "test-category" of the item
"Document_with_Category.txt"
```

I have a category named "<parameter1>"

Description

This step goes through all process of create a new category named <parameter1> and adds it an attribute named "test-attribute" of the type "Text: Field".

Assumption

Access to the Administration Categories page.

Example

The following example creates the "test-category-search" category to use later in the Scenario execution:

```
Given I have a category named "test-category-search"
When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder for Searches"
And I open the categories of the item "document.docx"
And I add the category "test-category-search" to the item
```

I remove a category named "<parameter1>"

Description

This step goes through all process of deletion of a category named <parameter1>.

Assumption

Access to the Administration Categories page.

Example

The following example deletes the "test-category-search" category and checks that the category doesn't exist in the system:

```
When I remove a category named "test-category-security"
Then I navigate to "Admin > Content Server Administration"
And Select "Open the Categories Volume"
And I cannot see the item "test-category-security"
```

I select and remove the category "<parameter1>"

Description

This step selects the category named <parameter1> and removes it from an item.

Assumption

The categories page for an item must be opened.

Example

The following example opens the categories of the item "document.txt" and checks if the item contains the category "test-category". Finally, removes the "test-category" from the item "document.txt":

Then I see that the item "document.txt" has the category "test-category"
And I select and remove the category "test-category"

Manage Items

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/manage-items-steps");
```

I click in add item and select "<parameter1>"

Description

This step opens the create form to an item of the type named <parameter1>.

Assumption

The browser table must be opened at the correct path.

Example

The following example opens the "Minium Tests" folder and opens the create form of the "Text Document" type to create a "Text Document" inside the folder "Minium Tests":

When I open the folder "Minium Tests"
And I click in add item and select "Text Document"

[[I-check-the-radio-button-Text]] ===== *I check the radio button Text* Description:: This step checks the "Text" option at the MIME type. Assumption:: The create form to a new "Text Document" must be opened. Example:: The following example opens the create form to create a "Text Document" and checks the "Text" option at the MIME type:

When I click in add item and select "Text Document"
And I check the radio button Text

I fill the field "<parameter1>" with "<parameter2>"

Description

This step fills a field named <parameter1> with a value defined at <parameter2>.

Assumption

A form with a field named <parameter1> must exists.

Example

The following example opens the create form to create a "Text Document" and fills the Name field with "Text Document.txt":

When I click in add item and select "Text Document"
And I fill the field "Name" with "Text Document.txt"

I can see the file "<parameter1>" inside the folder "<parameter2>"

Description

This step checks if an item named <parameter1> exists inside the folder <parameter2>, located at the `config.baseUrl`.

Assumption

The item named <parameter1> must exist inside the folder <parameter2>.

Example

The following example clicks "Add" to create the item "Text Document.txt" and then we navigate to `config.baseUrl`, open the folder "Minium Tests" and check if the file "Text Document.txt" exists:

When I click the button "Add"

Then I can see the file "Text Document.txt" inside the folder "Minium Tests"

I open the "<parameter1>" function for the item "<parameter2>"

Description

This step opens the <parameter1> function of the item named <parameter2>.

Assumption

The browser table must be opened at the correct path with an item named <parameter2> visible.

Example

The following example opens the folder "Minium Tests" and opens the edit function to the item "document.txt":

When I open the folder "Minium Tests"

And I open the "Edit" function for the item "document.txt"

I update the text to "<parameter1>"

Description

This step updates the document content with <parameter1>.

Assumption

The edit form of the document must be opened.

Example

The following example opens the folder "Minium Tests" and opens the edit function to the item "document.txt". Finally, updates the document content with "TEST EDIT":

When I open the folder "Minium Tests"

And I open the "Edit" function for the item "document.txt"

And I update the text to "TEST EDIT"

I see that the text contains "<parameter1>"

Description

This step checks that the document content contains the text <parameter1>.

Assumption

The edit form of the Text Document must be opened.

Example

The following example opens the folder "Minium Tests" and opens the edit function to the item "document.txt". Finally, checks that the document content contains the text "TEST EDIT":

When I open the folder "Minium Tests"
And I open the "Edit" function for the item "document.txt"
Then I see that the text contains "TEST EDIT"

I can see that the file "<parameter1>" no longer exists

Description

This step checks if an item named <parameter1> not exists inside the current folder.

Assumption

The browser table must be opened at the correct path.

Example

The following example opens the folder "Minium Tests" and checks that the "document.txt" not exist:

When I open the folder "Minium Tests"
Then I can see that the file "document.txt" no longer exists

I add an item of type "<parameter1>"

Description

This step opens the create form to the type named <parameter1>.

Assumption

The browser table must be opened.

Example

The following example opens the folder "Minium Tests" and then opens the create form to the "Document" type:

When I open the folder "Minium Tests"
Then I add an item of type "Document"

I check the radio button "<parameter1>"

Description

This step checks a radio button with a label named `<parameter1>`.

Assumption

The create form of the document must be opened.

Example

The following example opens the folder "Minium Tests" and then opens the create form to the "Document" type. Finally, it checks the "New" option at the create document form:

```
When I open the folder "Minium Tests"
And I add an item of type "Document"
And I check the radio button "New"
```

*I set the file "<parameter1>" to upload***Description**

This step selects a file named `<parameter1>` to upload.

Assumption

The file named `<parameter1>` must exists inside of the folder data **at the project**.

Example

The following example opens the folder "Minium Tests" and then opens the create form to the "Document" type. Finally, we set the "document.docx" file to upload:

```
When I open the folder "Minium Tests"
And I add an item of type "Document"
And I set the file "document.docx" to upload
```

*I set the "<parameter1>" as destiny name***Description**

This step defines the new item name with `<parameter1>`, generated from the copy.

Assumption

The copy form for an item must be opened.

Example

The following example performs the Copy action to copy the item and sets the "test_copied.docx" as the name of the new item:

```
When I perform the action "Copy"
And I set the "test_copied.docx" as destiny name
```

*I select the current folder as destiny to copy the "<parameter1>"***Description**

This step defines the current folder as the destiny folder to copy the item named `<parameter1>`.

Assumption

The content server browser to select the destiny folder must be opened.

Example

The following example opens the content server browser to select the destiny folder and sets the current folder as destiny folder to copy the "test.docx" item:

When I click the button "Browse Content Server..."
And I select the current folder as destiny to copy the "test.docx"

*I see the button Download the zip file enabled***Description**

This step checks if the button named "Download the zip file" is enable. This validates that the "Zip & Download" feature finished the compress operation and the download of the archive is available.

Assumption

The button "Zip & Download" (at the items to be Zipped & Downloaded form) was clicked.

Example

The following example clicks at the "Zip & Download" button to start the compress operation and then, we check if the "Download the zip file" is enabled to download the archive.

When I click the button "Zip & Download"
Then I see the button "Download the zip file" enabled

*I rate the item with "<parameter1>" with the comment "<parameter2>"***Description**

This step rates an item with the classification of <parameter1> and with a comment <parameter2>.

Assumption

The rate it form must be opened to an item.

Example

The following example opens the rate it properties for the item "document.txt" and rates the item with the classification of "5 - Excellent" and with the comment "Nice Document.":

When I open the "Rate it" function for the item "document.txt"
And I rate the item with "5 - Excellent" with the comment "Nice Document."

I can see a rating

Description

This step checks if an item has one rating.

Assumption

The rate it form must be opened to an item.

Example

The following example opens the rate it properties for the item "document.txt" and checks if the "document.txt" item has one rating:

When I open the "Rate it" function for the item "document.txt"
Then I can see a rating

*I (cannot)? see the item "<parameter1>" at My Favorites***Description**

This step checks if an item named <parameter1> exists (or not) at My Favorites.

Assumption

The "My Favorites" page must be opened.

Example

The following example navigates to the "My Favorites" page and validates if the "Folder created by Minium" item exists:

Then I navigate to "Personal > Favorites"
And I see the item "Folder created by Minium" at My Favorites

*I open the Edit/Organize option at My Favorites***Description**

This step opens the edit page of the favorite items.

Assumption

The "My Favorites" page must be opened.

Example

The following example navigates to the "My Favorites" page and then, opens the edit page of the favorite items:

When I navigate to "Personal > Favorites"
And I open the Edit/Organize option at My Favorites

*I select the item "<parameter1>" to remove from favorites***Description**

This step selects an item named <parameter1> to remove from favorites.

Assumption

The edit page of "My Favorites" must be opened and an item named <parameter1>

must exist.

Example

The following example opens the edit page of the favorite items and select the "Folder created by Minium" item to remove from favorites:

When I open the Edit/Organize option at My Favorites
And I select the item "Folder created by Minium" to remove from favorites

I select the "<parameter1>" as destiny to move the "<parameter2>"

Description

This step selects the folder named <parameter1> to be the destiny folder of the move action to the item named <parameter2>.

Assumption

The content server browser to select the destiny folder must be opened.

Example

The following example opens the content server browser to select the destiny folder and sets the "Folder created by Minium (2)" folder as destiny folder to move the "Folder created by Minium (1)":

When I click the button "Browse Content Server..."
And I select the "Folder created by Minium (2)" as destiny to move the "Folder created by Minium (1)"

I create a text document with the name "<parameter1>"

Description

This step goes through all process of create a Text Document named <parameter1>.

Assumption

The browser table must be opened.

Example

The following example opens the folder "Minium Tests" and creates a Text Document named "document.txt" inside the folder "Minium Tests":

When I open the folder "Minium Tests"
And I create a text document with the name "document.txt"

Reserve

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/reserve-steps");
```

I select current user as Reserve By

Description

This step defines that the current user reserved the item.

Assumption

The reserve page of an item must be opened.

Example

The following example opens the reserve function of the item "document.docx" and sets the current user as reserve by:

```
When I open the "Reserve" function for the item "document.docx"
And I select current user as Reserve By
```

*I cannot delete the item***Description**

This step checks if we have the warning message and the action was not completed.

Assumption

The action delete was triggered.

Example

The following example clicks delete to try to delete an item, check if the delete fails and if a warning message is shown:

```
When I click the button "Delete"
Then I cannot delete the item
```

Search

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/search-steps");
```

*I select "<parameter1>" from the "<parameter2>" filter***Description**

This step selects the facet item named <parameter1> at the facet list named <parameter2>.

Assumption

The browser table must be opened.

Example

The following example navigates to the Enterprise Workspace page and then, selects the Document at the Content Type facet list:

```
When I navigate to "Enterprise > Workspace"
And I select "Document" from the "Content Type" filter
```

The (search|filter)?(not)? returns the item "<parameter1>"

Description

This step check if an item named <parameter1> exists (or not) at the search or the filter results.

Assumption

The search/filter was executed.

Example 1

The following example selects the Document at the Content Type facet list and checks if the filter returns the item "document.docx":

When I select "Document" from the "Content Type" filter
Then The filter returns the item "document.docx"

Example 2

The following example performs the advance search and checks if the search don't returns the item "document.docx":

And I perform the advanced search
Then The search not returns the item "document.docx"

I search the text "<parameter1>" at the Content Filter

Description

This step filter the items at the browser table by name.

Assumption

The browser table must be opened.

Example

The following example opens the folder "Minium Tests" and search the items containing the substring "search" at the name:

When I open the folder "Minium Tests"
And I search the text "search" at the Content Filter

I wait that the document is indexed

Description

This step waits 90 seconds to allow the Content Server to index an item.

Example

The following example adds an item and wait for the item be indexed:

When I add an item of type "Document"
And I set the file "document.docx" to upload
And I click the button "Add"
And I wait that the document is indexed

I perform the search:

Description

This step fills the search form at the navigation bar.

Assumption

The navigation bar must be visible.

Example

The following example fills the fields in order to perform the search:

```
When I perform the search:
| Log-in Name      | <user_name> |
| Department      | DefaultGroup |
| Password        | 123456      |
| Verify Password | 123456      |
```

I add the location "<parameter1>" to the search

Description

This step adds the location <parameter1> as query filter to the advanced search.

Assumption

The advanced search page must be opened.

Example

The following example navigates to the advanced search page and adds the "Enterprise" location query filter to the search:

```
When I navigate to "Tools > Search"
And I add the location "Enterprise" to the search
```

I'm looking for the "<parameter1>" with the text "<parameter2>"

Description

This step selects the Full Text Search type named <parameter1> with the text <parameter2>.

Assumption

The advanced search page must be opened.

Example

The following example navigates to the advanced search page and sets the "Look For" with the value "Exact Phrase". Finally, adds the text "minium can" to search at the full text.

```
When I navigate to "Tools > Search"
And I'm looking for the "Exact Phrase" with the text "minium can"
```

I perform the advanced search

Description

This step submits the advanced search.

Assumption

The advanced search page must be opened.

Example

The following example navigates to the advanced search page, set the "Look For" with the value "Exact Phrase" and adds the text "minium can" to search at the full text. Finally, submits this search to retrieve the search results.

When I navigate to `"Tools > Search"`

And I'm looking for the `"Exact Phrase"` with the text `"minium can"`

And I perform the advanced search

I (add|remove)? the component "<parameter1>" to the search

Description

This step adds or removes a component named `<parameter1>` to the advanced search.

Assumption

The advanced search page must be opened.

Example

The following example navigates to the advanced search page and adds the "System Attributes" component to the search:

When I navigate to `"Tools > Search"`

And I add the component `"System Attributes"` to the search

I select the category "<parameter1>" to the search

Description

This step selects a category to the advanced search.

Assumption

The advanced search page must be opened and the component "Categories..." must be selected.

Example

The following example navigates to the advanced search page and adds the "Categories..." component to the advanced page. Finally, selects the "test-search" to filter the items with this category:

When I navigate to `"Tools > Search"`

And I add the component `"Categories..."` to the search

And I select the category `"test-search"` to the search

I remove the attribute "<parameter1>" from the component category

Description

This step removes an attribute from the category component of the advanced search.

Assumption

The advanced search page must be opened and the category component must be visible.

Example

The following example selects the "test-search" to filter the items with this category and remove from the filter the category attribute named "test-attribute":

```
When I select the category "test-search" to the search
And I remove the attribute "test-attribute" from the component category
```

I save the search with the name "<parameter1>" at the folder "<parameter2>"

Description

This step saves a search named <parameter1> at the folder <parameter2>.

Assumption

The search results must be opened.

Example

The following example fills the fields in order to perform the search and then, saves the search at the Enterprise "Minium Tests" folder.

```
When I perform the search:
| Search          | minium          |
| Slices         | From Here      |
| Object type    | Everything      |
| Last modified  | Modified in the past month |
And I save the search with the name "save-search" at the folder "Enterprise >
Minium Tests"
```

I open the Advanced Search at the search results

Description

This step opens the search configurations for a saved search.

Assumption

The save search item with the search results page must be opened.

Example

The following example opens the "save-search" previously saved and open the configurations for that search.

```
When I open the item "save-search"
And I open the Advanced Search at the search results
```

The search returns an item "<parameter1>" at the folder "<parameter2>"

Description

This step checks that the item named <parameter1> belongs to the folder <parameter2>.

Assumption

The search results must be opened and the <parameter1> is visible.

Example

The following example opens the similar items for the item "document1.docx" and check if we have an item "document2.docx" that belongs to the folder "Folder for Searches":

When I open the "Find Similar" function for the item "document1.docx"
Then The search returns an item "document2.docx" at the folder "Folder for Searches"

Security

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/security-steps");
```

Login as "<parameter1>" for the first time

Description

This step login the username named <parameter1> for the first time, after it's create.

Assumption

The user was created before.

Example

The following example creates an user named "user1", logout and login with the user1 the first time.

When At "Add Item", I select "User"
And I fill the "Add New User" form with:

Log-in Name	user1
Department	DefaultGroup
Password	123456
Verify Password	123456

And I click the button "Submit"
Then Login as "user1" for the first time

I change the password for "<parameter1>" at the first login

Description

This step sets a new password for the user named <parameter1>, after logged in for the first time.

Assumption

The user logged in for the first time.

Example

The following example logout and login with the "user1" the first time and changes the login password.

When Login as "user1" for the first time

Then I change the password for "user1" at the first login

*I open the permissions for "<parameter1>"***Description**

This step opens the permissions for <parameter1>.

Assumption

The permissions page of an item must be opened.

Example

The following example opens the permissions page of the item "document.txt" and then, opens the permissions for "Public Access":

When I open the "Permissions" function for the item "document.txt"

And I open the permissions for "Public Access"

*I (set|remove)? the access "<parameter1>"***Description**

This step adds or removes the access to an user, a group or the public access.

Assumption

The permission access page must be visible.

Example

The following example opens the permissions page of the item "document.txt" and then, opens the permissions for "Public Access". Finally, we remove the access "See" to "Public Access":

When I open the "Permissions" function for the item "document.txt"

And I open the permissions for "Public Access"

And I remove the access "See"

*I save the permissions***Description**

This step updates the permissions for an item.

Assumption

The permission page for an user, a group or the public access of an item must be opened.

Example

The following example removes the access "See" to the public access and updates the permissions for the item:

```
When I open the permissions for "Public Access"
And I remove the access "See"
And I save the permissions
```

*I close the permissions page***Description**

This step closes the permissions page for an item.

Assumption

The permission page of an item must be opened.

Example

The following example removes the access "See" to the "Public Access" and updates the permissions for the item. Finally, closes the permissions page:

```
When I open the permissions for "Public Access"
And I remove the access "See"
And I save the permissions
And I close the permissions page
```

*Login with the user "<parameter1>"***Description**

This step login the username named <parameter1>.

Assumption

The user was created before.

Example

The following example logout and login with the "user1" and navigates to the Enterprise Workspace:

```
Then Login with the user "user1"
And I navigate to "Enterprise > Workspace"
```

*I assign access for the (group|user)? "<parameter1>"***Description**

This step assigns the access to the user or the group named <parameter1>.

Assumption

The permission page of an item must be opened.

Example

The following example opens the permissions page of the item "document.txt" and then, assigns the access for the user "user1":

When I open the "Permissions" function for the item "document.txt"
And I assign access for the user "user1"

I can download the item "<parameter1>"

Description

This step validates that an item <parameter1> can be downloaded.

Assumption

The browser table must be opened at the correct path with an item named <parameter1> visible.

Example

The following example opens the folder "Minium Tests" and then, checks that the item "document.txt" can be downloaded:

Then I open the folder "Minium Tests"
And I can download the item "document.txt"

The user cannot create a document at Folder with Security

Description

This step checks that the Add Item not exists at the option bar.

Assumption

The browser table must be opened in the correct path.

Example

The following example opens the folder "Minium Tests" and validate that the "Add Item" is not visible:

Then I open the folder "Minium Tests"
And The user cannot create a document at Folder with Security

I select apply the permissions to "<parameter1>"

Description

This step sets the apply to permission named <parameter1> to an item. This value can be "This Item", "Sub-Items" or "This Item & Sub-Items".

Assumption

The permission page for an user or a group of an item must be opened.

Example

The following example assigns the access "See Contents" to the group "group2" and then, sets to apply the permissions to "This Item & Sub-Items":

When I assign access for the group "group2"
And I set the access "See Contents"
And I select apply the permissions to "This Item & Sub-Items"

I confirm the Apply Permissions to Sub-Items

Description

This step closes the dialog "Apply Permissions to Sub-Items" after we apply the permission access to sub-items.

Assumption

The confirm the Apply Permissions to Sub-Items must be opened.

Example

The following example sets to apply the permissions to "This Item & Sub-Items" and update the permissions for an item. Finally, confirms the Apply Permissions to Sub-Items:

```
When I select apply the permissions to "This Item & Sub-Items"
And I save the permissions
And I close the permissions page
And I confirm the Apply Permissions to Sub-Items
```

I see the permissions for "<parameter1>" has the access "<parameter2>"

Description

This step checks that the permissions for the user or the group named <parameter1> has the access named <parameter2>.

Assumption

The permission page for an item must be opened.

Example

The following example opens the permissions page for the item "document.txt" and then, validates that the access "See Contents" for the group "group2" is checked:

```
When I open the "Permissions" function for the item "document.txt"
Then I see the permissions for "group2" has the access "See Contents"
```

I remove the permission

Description

This step removes the assigned access for an user or a group.

Assumption

The permission page for an assigned user or a group of an item must be opened.

Example

The following example opens the assigned access for the group "group2" and removes this assigned access. Finally, confirms the deletion:

```
When I open the permissions for "group2"
And I remove the permission
And I accept the operation
```

I cannot see the permissions for "<parameter1>"

Description

This step validates that the user or group named <parameter1> not exists for an user.

Assumption

The permission page of an item must be opened.

Example

The following example opens the permissions page for the item "document.txt" and then, validates that the assigned access for the group "group2" not exists:

When I open the "Permissions" function for the item "document.txt"
Then I cannot see the permissions for "group2"

I update the document owner (user|group)? to "<parameter1>"

Description

This step updates the user or group owner of an item to an user or a group named <parameter1>.

Assumption

The permission page of an item must be opened and the user or the group <parameter1> must exist.

Example

The following example opens the permissions page for the item "document.txt" and then, changes the user owner to "user1":

When I open the "Permissions" function for the item "document.txt"
Then I update the document owner user to "user1"

I see that the document owner (user|group)? is "<parameter1>"

Description

This step checks that the user or group owner of an item is an user or a group named <parameter1>.

Assumption

The permission page of an item must be opened.

Example

The following example opens the permissions page for the item "document.txt" and then, checks if the group owner is "group2":

When I open the "Permissions" function for the item "document.txt"
Then I see that the document owner group is "group2"

Version

To import this step module, use the following instruction:

```
mc.loadDefinitions("otcs/classic/version-steps");
```

I (cannot)? see the version "<parameter1>"

Description

This step checks if a version named <parameter1> exists.

Assumption

The versions page of an item must be opened.

Example

The following example opens the versions for the item "test.docx" and validates that the version "version1.docx" exists:

```
Then I open the "Versions" function for the item "test.docx"
And I see the version "version1.docx"
```

I open the "<parameter1>" function for the version "<parameter2>"

Description

This step opens the function "<parameter1>" for the version named <parameter1>.

Assumption

The versions page of an item must be opened and the version named <parameter1> must exist.

Example

The following example opens the versions for the item "test.docx" and opens the "Unlock" version for the version "version1.docx":

```
When I open the "Versions" function for the item "test.docx"
And I open the "Unlock" function for the version "version1.docx"
```

I open the functions of the version "<parameter1>"

Description

This step opens the functions of the version named <parameter1>.

Assumption

The versions page of an item must be opened and the version named <parameter1> must exist.

Example

The following example opens the versions for the item "test.docx" and opens the functions of the version "document.txt":

When I open the "Versions" function for the item "test.docx"
And I open the functions of the version "version1.docx"

I see that the version description is "<parameter1>"

Description

This step checks if the description of a version is <parameter1>.

Assumption

The general properties of a version must be opened.

Example

The following example opens the general properties of the version "version1.docx" and checks if the description of the version is "New Version of test.docx Updated":

Then I open the "General" function for the version "version1.docx"
And I see that the version description is "New Version of test.docx Updated"

I can see that the version "<parameter1>" is(not)? locked

Description

This step checks if a version named <parameter1> is (or not) locked.

Assumption

The functions of the version named <parameter1> must be opened.

Example

The following example opens the functions of the version "document.docx" and checks if the version is not locked:

Then I open the functions of the version "version1.docx"
And I can see that the version "version1.docx" is not locked

I cannot see the function "<parameter1>" to the version

Description

This step checks if a function named <parameter1> not exists to a version.

Assumption

The functions of a version must be opened.

Example

The following example opens the functions of the version "test.docx" and validates that the function "Delete" is not visible:

When I open the functions of the version "version1.docx"
Then I cannot see the function "Delete" to the version

I set "<parameter1>" to purge all versions except the most recent

Description

This step sets the number of versions <parameter1> to purge except the most recent.

Assumption

The versions page of an item must be opened.

Example

The following example opens the versions for the item "test.docx" and sets "2" to purge all versions except the most recent:

When I open the "Versions" function for the item "test.docx"
And I set "2" to purge all versions except the most recent

I can see "<parameter1>" versions available

Description

This step checks that the number of versions that an item has is <parameter1>.

Assumption

The versions page of an item must be opened.

Example

The following example opens the versions for the item "test.docx" and validates that the number of versions for the item "test.docx" is "2":

Then I open the "Versions" function for the item "test.docx"
And I can see "2" versions available

Features

An example to the features that OTCS Minium Tests supports.

Administration

Feature: Administration

Background:

Given I'm logged in with Admin

Scenario: List users

When I navigate to "Enterprise > Users & Groups"

Then I should see the user "Admin"

Scenario Outline: Create users

Given I navigate to "Enterprise > Users & Groups"

And I delete if the user "<user_name>" exists

When At "Add Item", I select "User"

And I fill the "Add New User" form with:

Log-in Name	<user_name>
Department	DefaultGroup
Password	123456
Verify Password	123456

And I click the button "Submit"

And I navigate to "Enterprise > Users & Groups"

Then I should see the user "<user_name>"

Examples:

user_name	
test-user	
test-user-2	

Scenario: Edit user

When I navigate to "Enterprise > Users & Groups"

And I edit the user "test-user"

And I fill the field "Title" with "Minium User"

And I click the button "Update"

Then I navigate to "Enterprise > Users & Groups"

And I view the user "test-user"

And I should see the "Title" with the value "Minium User" at the popup "test-user"

Scenario: View user

When I navigate to "Enterprise > Users & Groups"

And I view the user "test-user"

Then I should see the "Title" with the value "Minium User" at the popup "test-user"

Scenario Outline: Create groups

Given I navigate to "Enterprise > Users & Groups"

And I delete if the group "<group_name>" exists

When At "Add Item", I select "Group"

And I fill the field "Group Name" with "<group_name>"

And I click the button "Submit"

Then I navigate to "Enterprise > Users & Groups"

And I should see the group "<group_name>"

Examples:

group_name	
test-group	
test-group-2	

Scenario Outline: Add users to a group

When I navigate to "Enterprise > Users & Groups"

And I edit the group "test-group"

And I search for the users that startsWith "test-user"

And I select the user "<user_name>" to Add to group

And I submit the selection of the users
Then I navigate to "Enterprise > Users & Groups"
And I open the groups of the user "<user_name>"
And The user belongs to the group "test-group"

Examples:

user_name	
test-user	
test-user-2	

Scenario: Check the groups of an user

When I navigate to "Enterprise > Users & Groups"
And I open the groups of the user "test-user"
Then The user belongs to the group "test-group"

Scenario: Remove user from group

When I navigate to "Enterprise > Users & Groups"
And I edit the group "test-group"
And I remove the user "test-user-2" from the group
Then I navigate to "Enterprise > Users & Groups"
And I open the groups of the user "test-user-2"
And The user not belongs to the group "test-group"

Scenario Outline: Delete groups

When I navigate to "Enterprise > Users & Groups"
And I delete the group "<group_name>"
Then I navigate to "Enterprise > Users & Groups"
And I should not see the group "<group_name>"

Examples:

group_name	
test-group	
test-group-2	

Scenario Outline: Delete users

When I navigate to "Enterprise > Users & Groups"
And I delete the user "<user_name>"
Then I navigate to "Enterprise > Users & Groups"
And I should not see the user "<user_name>"

Examples:

user_name	
test-user	
test-user-2	

Authentication

Feature: Authentication

Scenario: Login with wrong credentials

Given I'm at the login page
When I enter the wrong administrator credentials
And I click the button "Sign in"
Then I should see a warning message at signin

Scenario: Login as administrator

Given I'm at the login page
When I enter the correct administrator credentials
And I click the button "Sign in"
Then I should be redirected to the main page

Scenario Outline: Content server administration log-in

Given I'm logged in with Admin
When I navigate to "Admin > <option>"
Then I should be asked to type the Content Server administration password

Examples:

option	
Content Server Administration	
Search Admin Browser	

Scenario: Content server administration log-in with wrong credentials

Given I'm logged in with Admin
When I navigate to "Admin > Content Server Administration"
And I enter the wrong administration password
And I click the button "Log-in"
Then I should see an authentication error

Scenario: Content server administration log-in with correct credentials

Given I'm logged in with Admin
When I navigate to "Admin > Content Server Administration"
And I enter the correct administration password
And I click the button "Log-in"
Then I should see the Content Server Administration page

Scenario: Logout

Given I'm logged in with Admin
When I logout
Then I'm no longer authenticated

Categories

Feature: Categories

Background:

Given I'm logged in with Admin

Scenario: Create a Category

When I navigate to "Admin > Content Server Administration"

And Type the Content Server administration password

And Select "Open the Categories Volume"

And I add an item of type "Category"

And I fill the field "Name" with "test-category"

And I click the button "Add"

Then I see the item "test-category"

Scenario: Add an Attribute to the new Category

When I navigate to "Admin > Content Server Administration"

And Type the Content Server administration password

And Select "Open the Categories Volume"

And I open "test-category" with the type "Category"

And I choose to add a new "Text: Field" Attribute

And I fill the field "Name" with "test-attribute"

And I click the button "OK"

And I click the button "Submit"

Then I open the category "test-category" and see the attribute "test-attribute"

Scenario: Add Category to a Folder

Given I navigate to "Enterprise > Workspace"

And I create the folder "Minium Tests" if not exists

When I open the folder "Minium Tests"

And I create the folder "Folder with Category" if not exists

And I open the "Categories" function for the item "Folder with Category"

And I add the category "test-category" to the item

Then I see that the item "Folder with Category" has the category "test-category"

Scenario: Propagate Category changes to a Folder

When I open the folder "Minium Tests > Folder with Category"

And I create a text document with the name "Document_with_Category.txt"

And I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests"

And I open the "Categories" function for the item "Folder with Category"

And I apply the categories to Sub-Items of the folder "Folder with Category"

And I click the button "Submit"

Then I open the folder "Folder with Category"

And I see that the item "Document_with_Category.txt" has the category "test-category"

Scenario: Delete a Category

When I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests > Folder with Category"

And I remove the category "test-category" of the item "Document_with_Category.txt"

And I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests"

And I remove the category "test-category" of the item "Folder with Category"

And I navigate to "Admin > Content Server Administration"

And Type the Content Server administration password

And Select "Open the Categories Volume"

And I select the item "test-category"

And I perform the action "Delete"

And I confirm the deletion

Then I navigate to "Admin > Content Server Administration"

And Select "Open the Categories Volume"

And I cannot see the item "test-category"

Scenario: Delete data of this test

When I navigate to "Enterprise > Workspace"
And I select the item "Minium Tests"
And I perform the action "Delete"
And I confirm the deletion
Then I cannot see the item "Minium Tests"

Manage Items

Feature: Manage Items

Background:

Given I'm logged in with Admin
And I navigate to "Enterprise > Workspace"

Scenario: Create a Text Document

Given I create the folder "Minium Tests" if not exists
When I open the folder "Minium Tests"
And I click in add item and select "Text Document"
And I check the radio button Text
And I fill the field "Text" with "Lorem Ipsum is simply dummy text of the printing and typesetting industry."
And I fill the field "Name" with "Text Document created by Minium.txt"
And I click the button "Add"
Then I can see the file "Text Document created by Minium.txt" inside the folder "Minium Tests"

Scenario: Edit a text document

When I open the folder "Minium Tests"
And I open the "Edit" function for the item "Text Document created by Minium.txt"
And I update the text to "Lorem Ipsum is simply dummy text of the printing and typesetting industry. - TEST EDIT"
And I click the button "Add Version"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Edit" function for the item "Text Document created by Minium.txt"
And I see that the text contains "TEST EDIT"

Scenario: Create a Shortcut to Text Document

When I open the folder "Minium Tests"
And I open the "Make Shortcut" function for the item "Text Document created by Minium.txt"
And I fill the field "Name" with "Text Document created by Minium - shortcut.txt"
And I click the button "Add"
Then I can see the file "Text Document created by Minium - shortcut.txt" inside the folder "Minium Tests"

Scenario: Delete the Shortcut of the Text Document

When I open the folder "Minium Tests"
And I select the item "Text Document created by Minium - shortcut.txt"
And I perform the action "Delete"
And I confirm the deletion
Then I can see that the file "Text Document created by Minium - shortcut.txt" no longer exists

Scenario: Rename a text document

When I open the folder "Minium Tests"
And I open the "Rename" function for the item "Text Document created by Minium.txt"
And I fill the field "Name" with "Text Document renamed.txt"
And I click the button "Update"
Then I can see the file "Text Document renamed.txt" inside the folder "Minium Tests"

Scenario: Delete a Text Document

When I open the folder "Minium Tests"
And I select the item "Text Document renamed.txt"
And I perform the action "Delete"
And I confirm the deletion
Then I can see that the file "Text Document renamed.txt" no longer exists

Scenario: Create a new Document

When I open the folder "Minium Tests"
And I add an item of type "Document"
And I check the radio button "New"

```

And I fill the field "Name" with "test.docx"
And I click the button "Add"
Then I can see the file "test.docx" inside the folder "Minium Tests"

```

Scenario: Upload a new Document

```

When I open the folder "Minium Tests"
And I add an item of type "Document"
And I set the file "travel-approvals.docx" to upload
And I click the button "Add"
Then I can see the file "travel-approvals.docx" inside the folder "Minium
Tests"

```

Scenario: Copy the Document

```

When I open the folder "Minium Tests"
And I select the item "test.docx"
And I perform the action "Copy"
And I set the "test_copied.docx" as destiny name
And I click the button "Browse Content Server..."
And I select the current folder as destiny to copy the "test.docx"
And I click the button "Copy"
And I click the button "OK"
Then I can see the file "test_copied.docx" inside the folder "Minium
Tests"

```

Scenario: Zip and Download

```

When I open the folder "Minium Tests"
And I select the item "test.docx"
And I perform the action "Zip & Download"
And I click the button "Zip & Download"
Then I see the button Download the zip file enabled

```

Scenario: Rate a Document

```

When I open the folder "Minium Tests"
And I open the "Rate It" function for the item "test.docx"
And I rate the item with "5 - Excellent" with the comment "Nice Document."
And I click the button "Submit"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Rate It" function for the item "test.docx"
And I can see a rating

```

Scenario: Move a Document

```

When I open the folder "Minium Tests"
And I create the folder "Test Move" if not exists
And I select the item "test.docx"
And I perform the action "Move"
And I click the button "Browse Content Server..."
And I select the "Test Move" as destiny to move the "test.docx"
And I click the button "Move"
And I click the button "OK"
Then I open the folder "Test Move"
And I see the item "test.docx"

```

Scenario Outline: Delete Documents

```

When I open the folder "Minium Tests"
And I select the item "<file>"
And I perform the action "Delete"
And I confirm the deletion
Then I can see that the file "<file>" no longer exists

```

Examples:

file	
test_copied.docx	
travel-approvals.docx	

Scenario: Create a Folder

```

When I open the folder "Minium Tests"
And I create the folder "Folder created by Minium" if not exists
Then I see the item "Folder created by Minium"

```

Scenario: Add Folder to Favorite

```

When I open the folder "Minium Tests"

```

And I open the "Add to Favorites" function for the item "Folder created by Minium"

Then I navigate to "Personal > Favorites"

And I see the item "Folder created by Minium" at My Favorites

Scenario: Remove folder from Favorite

When I navigate to "Personal > Favorites"

And I open the Edit/Organize option at My Favorites

And I select the item "Folder created by Minium" to remove from favorites

And I click the button "Submit"

Then I cannot see the item "Folder created by Minium" at My Favorites

Scenario Outline: Create two Folders

When I open the folder "Minium Tests"

And I create the folder "<folder>" if not exists

Then I see the item "<folder>"

Examples:

folder	
Folder created by Minium (1)	
Folder created by Minium (2)	

Scenario: Move a Folder

When I open the folder "Minium Tests"

And I select the item "Folder created by Minium (1)"

And I perform the action "Move"

And I click the button "Browse Content Server..."

And I select the "Folder created by Minium (2)" as destiny to move the "Folder created by Minium (1)"

And I click the button "Move"

And I click the button "OK"

Then I open the folder "Folder created by Minium (2)"

And I see the item "Folder created by Minium (1)"

Scenario: Delete data of this test

When I select the item "Minium Tests"

And I perform the action "Delete"

And I confirm the deletion

Then I cannot see the item "Minium Tests"

Reserve

Feature: Reserve Documents

Background:

Given I'm logged in with Admin
And I navigate to "Enterprise > Workspace"

Scenario: Create a new Document

Given I create the folder "Minium Tests" if not exists
When I open the folder "Minium Tests"
And I add an item of type "Document"
And I check the radio button "New"
And I fill the field "Name" with "test.docx"
And I click the button "Add"
Then I can see the file "test.docx" inside the folder "Minium Tests"

Scenario: Reserve a Document

When I open the folder "Minium Tests"
And I open the "Reserve" function for the item "test.docx"
And I select current user as Reserve By
And I click the button "Submit"
Then I navigate to "Personal > Reserved Items"
And I see the item "test.docx"

Scenario: Cannot delete a reserved document

When I open the folder "Minium Tests"
And I select the item "test.docx"
And I perform the action "Delete"
And I click the button "Delete"
Then I cannot delete the item

Scenario: Unreserve a Document

When I open the folder "Minium Tests"
And I open the "Unreserve" function for the item "test.docx"
And I click the button "Submit"
Then I navigate to "Personal > Reserved Items"
And I cannot see the item "test.docx"

Scenario: Delete data of this test

When I select the item "Minium Tests"
And I perform the action "Delete"
And I confirm the deletion
Then I cannot see the item "Minium Tests"

Search

Feature: Search

Background:

Given I'm logged in with Admin
And I navigate to "Enterprise > Workspace"

Scenario: Create data for this test

Given I create the folder "Minium Tests" if not exists
When I open the folder "Minium Tests"
And I create the folder "Folder for Searches" if not exists
And I open the folder "Folder for Searches"
And I add an item of type "Document"
And I set the file "Document_for_search.docx" to upload
And I click the button "Add"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder for Searches"
And I see the item "Document_for_search.docx"

Scenario: Facet Browsing for a Document

When I select "Document" from the "Content Type" filter
Then The filter returns the item "Document_for_search.docx"

Scenario: Content Filter for a Document

When I open the folder "Minium Tests > Folder for Searches"
And I search the text "search" at the Content Filter
Then The filter returns the item "Document_for_search.docx"

Scenario: Content Filter for a Document (No Results)

When I open the folder "Minium Tests > Folder for Searches"
And I search the text "search2" at the Content Filter
Then The filter not returns the item "Document_for_search.docx"

Scenario: Search for a Document

Given I wait that the document is indexed
When I perform the search:

Search	minium can	
Slices	From Here	
Object type	Everything	
Last modified	Modified in the past month	

Then The search returns the item "Document_for_search.docx"

Scenario: Advanced Search for a Document

When I navigate to "Tools > Search"
And I add the location "Enterprise" to the search
And I'm looking for the "Exact Phrase" with the text "minium can"
And I perform the advanced search
Then The search returns the item "Document_for_search.docx"

Scenario: Advanced Search for a Document (No Results)

When I navigate to "Tools > Search"
And I add the location "Enterprise" to the search
And I'm looking for the "Exact Phrase" with the text "minium can test"
And I perform the advanced search
Then The search not returns the item "Document_for_search.docx"

Scenario: Advanced Search for a Document (Category)

Given I have a category named "test-category-search"
When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder for Searches"
And I open the "Categories" function for the item "Document_for_search.docx"
And I add the category "test-category-search" to the item
And I navigate to "Tools > Search"
And I add the component "System Attributes" to the search
And I add the component "Categories..." to the search
And I select the category "test-category-search" to the search
And I remove the attribute "test-attribute" from the component category
And I remove the component "System Attributes" to the search
And I'm looking for the "All Words" with the text "Demo"

```

And I perform the advanced search
Then The search returns the item "Document_for_search.docx"
And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder for Searches"
And I remove the category "test-category-search" of the item
"Document_for_search.docx"

```

Scenario: Save a search

```

When I perform the search:
| Search          | minium          |
| Slices          | From Here      |
| Object type     | Everything      |
| Last modified  | Modified in the past month |
And I save the search with the name "test-save-search" at the folder
"Enterprise > Minium Tests"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I see the item "test-save-search"

```

Scenario: Retrieve a saved a search (No Results)

```

When I open the folder "Minium Tests"
And I open the item "test-save-search"
And I open the Advanced Search at the search results
And I add the location "Admin Home" to the search
And I perform the advanced search
Then The search not returns the item "Document_for_search.docx"

```

Scenario: Find Similar

```

When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder for Searches"
And I create the folder "Folder for Searches (2)" if not exists
And I open the folder "Folder for Searches (2)"
And I add an item of type "Document"
And I set the file "Document_for_search.docx" to upload
And I click the button "Add"
And I wait that the document is indexed
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder for Searches"
And I open the "Find Similar" function for the item
"Document_for_search.docx"
And The search returns an item "Document_for_search.docx" at the folder
"Folder for Searches (2)"

```

Scenario: Delete data of this test

```

When I select the item "Minium Tests"
And I perform the action "Delete"
And I confirm the deletion
Then I cannot see the item "Minium Tests"
And I remove a category named "test-category-search"
And I cannot see the item "test-category-search"

```

Security

Feature: Security

Background:

Given I'm logged in with Admin

Scenario Outline: Create groups for this test

Given I navigate to "Enterprise > Users & Groups"

And I delete if the group "<group_name>" exists

When At "Add Item", I select "Group"

And I fill the field "Group Name" with "<group_name>"

And I click the button "Submit"

Then I navigate to "Enterprise > Users & Groups"

And I should see the group "<group_name>"

Examples:

group_name	
test-group	
test-group-2	

Scenario Outline: Create users for this test

Given I navigate to "Enterprise > Users & Groups"

And I delete if the user "<user_name>" exists

When At "Add Item", I select "User"

And I fill the "Add New User" form with:

Log-in Name		<user_name>	
Department		<group_name>	
Password		123456	
Verify Password		123456	

And I click the button "Submit"

Then Login as "<user_name>" for the first time

And I change the password for "<user_name>" at the first login

Examples:

user_name		group_name	
test-user		test-group	
test-user-2		test-group-2	

Scenario: User cannot see the document

Given I navigate to "Enterprise > Workspace"

And I create the folder "Minium Tests" if not exists

When I open the folder "Minium Tests"

And I create the folder "Folder with Security" if not exists

And I open the folder "Folder with Security"

And I create a text document with the name "Document_with_Security.txt"

And I open the "Permissions" function for the item

"Document_with_Security.txt"

And I open the permissions for "Public Access"

And I remove the access "See"

And I save the permissions

And I close the permissions page

Then Login with the user "test-user"

And I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests > Folder with Security"

And I cannot see the item "Document_with_Security.txt"

Scenario: Grant "See" permission for the document to User

When I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests > Folder with Security"

And I open the "Permissions" function for the item

"Document_with_Security.txt"

And I assign access for the user "test-user"

And I set the access "See"

And I save the permissions

And I close the permissions page

Then Login with the user "test-user"

And I navigate to "Enterprise > Workspace"

And I open the folder "Minium Tests > Folder with Security"

And I see the item "Document_with_Security.txt"

```

Scenario: Grant "See Contents" permission for the document to User
  When I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "Permissions" function for the item
  "Document_with_Security.txt"
  And I open the permissions for "test-user"
  And I set the access "See Contents"
  And I save the permissions
  And I close the permissions page
  Then Login with the user "test-user"
  And I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I can download the item "Document_with_Security.txt"

Scenario: Grant "Modify" permission for the document to User
  When I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "Permissions" function for the item
  "Document_with_Security.txt"
  And I open the permissions for "test-user"
  And I set the access "Modify"
  And I save the permissions
  And I close the permissions page
  Then Login with the user "test-user"
  And I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "General" function for the item
  "Document_with_Security.txt"
  And I fill the field "Description" with "Test Description"
  And I click the button "Update"
  And I open the "General" function for the item
  "Document_with_Security.txt"
  And I see that the version description is "Test Description"

Scenario: Grant "Edit Attributes" permission for the document to User
  Given I have a category named "test-category-security"
  When I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "Permissions" function for the item
  "Document_with_Security.txt"
  And I open the permissions for "test-user"
  And I set the access "Edit Attributes"
  And I save the permissions
  And I close the permissions page
  Then Login with the user "test-user"
  And I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "Categories" function for the item
  "Document_with_Security.txt"
  And I add the category "test-category-security" to the item
  And I see that the item "Document_with_Security.txt" has the category
  "test-category-security"
  And I select and remove the category "test-category-security"

Scenario: Grant "Reserve" permission for the document to User
  When I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "Permissions" function for the item
  "Document_with_Security.txt"
  And I open the permissions for "test-user"
  And I set the access "Reserve"
  And I save the permissions
  And I close the permissions page
  Then Login with the user "test-user"
  And I navigate to "Enterprise > Workspace"
  And I open the folder "Minium Tests > Folder with Security"
  And I open the "Reserve" function for the item
  "Document_with_Security.txt"
  And I select current user as Reserve By
  And I click the button "Submit"
  And I navigate to "Personal > Reserved Items"
  And I see the item "Document_with_Security.txt"

```

```

And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Unreserve" function for the item
"Document_with_Security.txt"
And I click the button "Submit"
And I navigate to "Personal > Reserved Items"
And I cannot see the item "Document_with_Security.txt"

Scenario: Grant "Delete Versions" permission for the document to User
When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I open the permissions for "test-user"
And I set the access "Delete Versions"
And I save the permissions
And I close the permissions page
Then Login with the user "test-user"
And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Add Version" function for the item
"Document_with_Security.txt"
And I set the file "Document_with_Security_v2.txt" to upload
And I fill the field "Description" with "New Version of
Document_with_Security.txt"
And I click the button "Add Version"
And I open the "Versions" function for the item
"Document_with_Security.txt"
And I see the version "Document_with_Security_v2.txt"
And I open the "Delete" function for the version
"Document_with_Security_v2.txt"
And I click the button "Delete Version"
And I cannot see the version "Document_with_Security_v2.txt"

Scenario: Grant "Delete" permission for the document to User
When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I open the permissions for "test-user"
And I set the access "Delete"
And I save the permissions
And I close the permissions page
Then Login with the user "test-user"
And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I select the item "Document_with_Security.txt"
And I perform the action "Delete"
And I confirm the deletion
And I can see that the file "Document_with_Security.txt" no longer exists

Scenario: Grant "Edit Permissions" permission for the document to User
When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I create a text document with the name "Document_with_Security.txt"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I open the permissions for "Public Access"
And I remove the access "See"
And I save the permissions
And I assign access for the user "test-user"
And I set the access "Edit Permissions"
And I save the permissions
And I close the permissions page
Then Login with the user "test-user"
And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I open the permissions for "test-user"

Scenario: User cannot create document at Folder with Security

```

```

When Login with the user "test-user"
And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
Then The user cannot create a document at Folder with Security

```

Scenario: User is able to create a document at Folder with Security

```

When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Permissions" function for the item "Folder with Security"
And I open the permissions for "Public Access"
And I remove the access "See"
And I save the permissions
And I assign access for the group "test-group"
And I set the access "See Contents"
And I set the access "Add Items"
And I save the permissions
And I close the permissions page
Then Login with the user "test-user"
And I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I create a text document with the name "Test_add.txt"
And I see the item "Test_add.txt"

```

Scenario: The creation of permissions are propagated to sub-items

```

When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Permissions" function for the item "Folder with Security"
And I assign access for the group "test-group-2"
And I set the access "See Contents"
And I select apply the permissions to "This Item & Sub-Items"
And I save the permissions
And I close the permissions page
And I confirm the Apply Permissions to Sub-Items
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item "Test_add.txt"
And I see the permissions for "test-group-2" has the access "See Contents"
And I close the permissions page
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I see the permissions for "test-group-2" has the access "See Contents"
And I close the permissions page

```

Scenario: The deletion of permissions are propagated to sub-items

```

When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Permissions" function for the item "Folder with Security"
And I open the permissions for "test-group-2"
And I select apply the permissions to "This Item & Sub-Items"
And I remove the permission
And I accept the operation
And I close the permissions page
And I confirm the Apply Permissions to Sub-Items
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item "Test_add.txt"
And I cannot see the permissions for "test-group-2"
And I close the permissions page
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I cannot see the permissions for "test-group-2"
And I close the permissions page

```

Scenario: Update the owner of the document

```

When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I update the document owner user to "test-user"
And I close the permissions page
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"

```

```

And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I see that the document owner user is "test-user"

```

```

Scenario: Update the owner group of the document
When I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I update the document owner group to "test-group-2"
And I close the permissions page
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests > Folder with Security"
And I open the "Permissions" function for the item
"Document_with_Security.txt"
And I see that the document owner group is "test-group-2"

```

```

Scenario: Delete data of this test
When I navigate to "Enterprise > Workspace"
And I select the item "Minium Tests"
And I perform the action "Delete"
And I confirm the deletion
And I remove a category named "test-category-security"
Then I navigate to "Enterprise > Workspace"
And I cannot see the item "Minium Tests"
And I navigate to "Admin > Content Server Administration"
And Select "Open the Categories Volume"
And I cannot see the item "test-category-security"

```

```

Scenario Outline: Delete users of this test
When I navigate to "Enterprise > Users & Groups"
And I delete the user "<user_name>"
Then I navigate to "Enterprise > Users & Groups"
And I should not see the user "<user_name>"

```

Examples:

```

| user_name |
| test-user |
| test-user-2 |

```

```

Scenario Outline: Delete groups of this test
When I navigate to "Enterprise > Users & Groups"
And I delete the group "<group_name>"
Then I navigate to "Enterprise > Users & Groups"
And I should not see the group "<group_name>"

```

Examples:

```

| group_name |
| test-group |
| test-group-2 |

```

Version

Feature: Version

Background:

Given I'm logged in with Admin
And I navigate to "Enterprise > Workspace"

Scenario: Create a new Document

Given I create the folder "Minium Tests" if not exists
When I open the folder "Minium Tests"
And I add an item of type "Document"
And I check the radio button "New"
And I fill the field "Name" with "test.docx"
And I click the button "Add"
Then I can see the file "test.docx" inside the folder "Minium Tests"

Scenario: Add a version to a Document

When I open the folder "Minium Tests"
And I open the "Add Version" function for the item "test.docx"
And I set the file "travel-approvals.docx" to upload
And I fill the field "Description" with "New Version of test.docx"
And I click the button "Add Version"
Then I open the "Versions" function for the item "test.docx"
And I see the version "travel-approvals.docx"

Scenario: Update version description

When I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the "General" function for the version "travel-approvals.docx"
And I fill the field "Description" with "New Version of test.docx Updated"
And I click the button "Update"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the "General" function for the version "travel-approvals.docx"
And I see that the version description is "New Version of test.docx

Updated"

Scenario: Lock version

When I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the "Lock" function for the version "travel-approvals.docx"
And I click the button "Lock Version"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the functions of the version "travel-approvals.docx"
And I can see that the version "travel-approvals.docx" is locked

Scenario: Try to delete a locked version

When I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the functions of the version "travel-approvals.docx"
Then I cannot see the function "Delete" to the version

Scenario: Cannot delete a document with a locked version

When I open the folder "Minium Tests"
And I select the item "test.docx"
And I perform the action "Delete"
And I click the button "Delete"
Then I cannot delete the item

Scenario: Unlock version

When I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the "Unlock" function for the version "travel-approvals.docx"
And I click the button "Unlock Version"
Then I navigate to "Enterprise > Workspace"
And I open the folder "Minium Tests"

And I open the "Versions" function for the item "test.docx"
And I open the functions of the version "travel-approvals.docx"
And I can see that the version "travel-approvals.docx" is not locked

Scenario: Add another version to a Document

When I open the folder "Minium Tests"
And I open the "Add Version" function for the item "test.docx"
And I set the file "travel-approvals-v2.docx" to upload
And I fill the field "Description" with "New Version of test.docx"
And I click the button "Add Version"
Then I open the "Versions" function for the item "test.docx"
And I see the version "travel-approvals-v2.docx"

Scenario: Purge versions

When I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I set "2" to purge all versions except the most recent
And I click the button "Purge Versions"
And I accept the operation
Then I open the "Versions" function for the item "test.docx"
And I can see "2" versions available

Scenario: Remove a version from Document

When I open the folder "Minium Tests"
And I open the "Versions" function for the item "test.docx"
And I open the "Delete" function for the version "travel-approvals.docx"
And I click the button "Delete Version"
Then I cannot see the version "travel-approvals.docx"

Scenario: Delete data of this test

When I select the item "Minium Tests"
And I perform the action "Delete"
And I confirm the deletion
Then I cannot see the item "Minium Tests"

Steps Index

Given definitions

- I create the folder "<parameter>" if not exists
- I delete if the group "<parameter>" exists
- I delete if the user "<parameter>" exists
- I have a category named "<parameter>"
- I wait that the document is indexed
- I'm at the login page
- I'm at the main page
- I'm logged in with Admin

When definitions

- At "Add Item", I select "(User|Group)"
- I accept the operation
- I add an item of type "<parameter>"
- I add the category "<parameter>" to the item
- I add the component "<parameter>" to the search
- I add the location "<parameter>" to the search
- I apply the categories to Sub-Items of the folder "<parameter>"
- I assign access for the group "<parameter>"
- I assign access for the user "<parameter>"
- I check the radio button "<parameter>"
- I choose to add a new "Text: Field" Attribute
- I click in add item and select "<parameter>"
- I click the button "<parameter>"
- I close the permissions page
- I confirm the Apply Permissions to Sub-Items
- I confirm the deletion
- I create a text document with the name "<parameter>"
- I delete the group "<parameter>"
- I delete the user "<parameter>"
- I edit the group "<parameter>"
- I edit the user "<parameter>"
- I enter the correct administration password
- I enter the correct administrator credentials
- I enter the wrong administration password

- I enter the wrong administrator credentials
- I fill the "<parameter>" form with:
- I fill the field "<parameter>" with "<parameter>"
- I logout
- I navigate to "<parameter>"
- I open "<parameter>" with the type "<parameter>"
- I open the "<parameter>" function for the item "<parameter>"
- I open the Advanced Search at the search results
- I open the Edit/Organize option at My Favorites
- I open the folder "<parameter>"
- I open the item "<parameter>"
- I open the permissions for "<parameter>"
- I perform the action "<parameter>"
- I perform the advanced search
- I perform the search:
- I rate the item with "<parameter>" with the comment "<parameter>"
- I remove the access "<parameter>"
- I remove the attribute "<parameter>" from the component category
- I remove the category "<parameter>" of the item "<parameter>"
- I remove the component "<parameter>" to the search
- I remove the permission
- I remove the user "<parameter>" from the group
- I save the permissions
- I save the search with the name "<parameter>" at the folder "<parameter>"
- I search for the users that startsWith "<parameter>"
- I search the text "<parameter>" at the Content Filter
- I select "<parameter>" from the "<parameter>" filter
- I select apply the permissions to "<parameter>"
- I select current user as Reserve By
- I select the "<parameter>" as destiny to move the "<parameter>"
- I select the category "<parameter>" to the search
- I select the current folder as destiny to copy the "<parameter>"
- I select the item "<parameter>"
- I select the item "<parameter>" to remove from favorites
- I select the user "<parameter>" to Add to group
- I set "<parameter>" to purge all versions except the most recent

- I set the "<parameter>" as destiny name
- I set the access "<parameter>"
- I set the file "<parameter>" to upload
- I submit the selection of the users
- I update the document owner group to "<parameter>"
- I update the document owner user to "<parameter>"
- I update the text to "<parameter>"
- I'm looking for the "<parameter>" with the text "<parameter>"
- Select "<parameter>"
- Type the Content Server administration password

Then definitions

- I can download the item "<parameter>"
- I can see "<parameter>" versions available
- I can see a rating
- I can see that the file "<parameter>" no longer exists
- I can see that the version "<parameter>" is locked
- I can see that the version "<parameter>" is not locked
- I can see the file "<parameter>" inside the folder "<parameter>"
- I cannot delete the item
- I cannot see the function "<parameter>" to the version
- I cannot see the item "<parameter>"
- I cannot see the item "<parameter>" at My Favorites
- I cannot see the permissions for "<parameter>"
- I cannot see the version "<parameter>"
- I change the password for "<parameter>" at the first login
- I open the "<parameter>" function for the version "<parameter>"
- I open the category "<parameter>" and see the attribute "<parameter>"
- I open the functions of the version "<parameter>"
- I open the groups of the user "<parameter>"
- I remove a category named "<parameter>"
- I see that the document owner group is "<parameter>"
- I see that the document owner user is "<parameter>"
- I see that the item "<parameter>" has the category "<parameter>"
- I see that the text contains "<parameter>"
- I see that the version description is "<parameter>"
- I see the button Download the zip file enabled

- I see the item "<parameter>"
- I see the item "<parameter>" at My Favorites
- I see the permissions for "<parameter>" has the access "<parameter>"
- I see the version "<parameter>"
- I select and remove the category "<parameter>"
- I should be asked to type the Content Server administration password
- I should be redirected to the main page
- I should not see the group "<parameter>"
- I should not see the user "<parameter>"
- I should see a warning message at signin
- I should see an authentication error
- I should see the "<parameter>" with the value "<parameter>" at the popup "<parameter>"
- I should see the Content Server Administration page
- I should see the group "<parameter>"
- I should see the user "<parameter>"
- I view the user "<parameter>"
- I'm no longer authenticated
- Login as "<parameter>" for the first time
- Login with the user "<parameter>"
- The filter not returns the item "<parameter>"
- The filter returns the item "<parameter>"
- The search not returns the item "<parameter>"
- The search returns an item "<parameter>" at the folder "<parameter>"
- The search returns the item "<parameter>"
- The user belongs to the group "<parameter>"
- The user cannot create a document at Folder with Security
- The user not belongs to the group "<parameter>"