



TEAMCENTER

Teamcenter Quality — Deployment and Administration

Teamcenter 2412

Unpublished work. © 2025 Siemens

This Documentation contains trade secrets or otherwise confidential information owned by Siemens Industry Software Inc. or its affiliates (collectively, "Siemens"), or its licensors. Access to and use of this Documentation is strictly limited as set forth in Customer's applicable agreement(s) with Siemens. This Documentation may not be copied, distributed, or otherwise disclosed by Customer without the express written permission of Siemens, and may not be used in any way not expressly authorized by Siemens.

This Documentation is for information and instruction purposes. Siemens reserves the right to make changes in specifications and other information contained in this Documentation without prior notice, and the reader should, in all cases, consult Siemens to determine whether any changes have been made.

No representation or other affirmation of fact contained in this Documentation shall be deemed to be a warranty or give rise to any liability of Siemens whatsoever.

If you have a signed license agreement with Siemens for the product with which this Documentation will be used, your use of this Documentation is subject to the scope of license and the software protection and security provisions of that agreement. If you do not have such a signed license agreement, your use is subject to the Siemens Universal Customer Agreement, which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/base/uca/>, as supplemented by the product specific terms which may be viewed at <https://www.sw.siemens.com/en-US/sw-terms/supplements/>.

SIEMENS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS DOCUMENTATION INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY. SIEMENS SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL OR PUNITIVE DAMAGES, LOST DATA OR PROFITS, EVEN IF SUCH DAMAGES WERE FORESEEABLE, ARISING OUT OF OR RELATED TO THIS DOCUMENTATION OR THE INFORMATION CONTAINED IN IT, EVEN IF SIEMENS HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TRADEMARKS: The trademarks, logos, and service marks (collectively, "Marks") used herein are the property of Siemens or other parties. No one is permitted to use these Marks without the prior written consent of Siemens or the owner of the Marks, as applicable. The use herein of third party Marks is not an attempt to indicate Siemens as a source of a product, but is intended to indicate a product from, or associated with, a particular third party. A list of Siemens' Marks may be viewed at: www.plm.automation.siemens.com/global/en/legal/trademarks.html. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

About Siemens Digital Industries Software

Siemens Digital Industries Software is a global leader in the growing field of product lifecycle management (PLM), manufacturing operations management (MOM), and electronic design automation (EDA) software, hardware, and services. Siemens works with more than 100,000 customers, leading the digitalization of their planning and manufacturing processes. At Siemens Digital Industries Software, we blur the boundaries between industry domains by integrating the virtual and physical, hardware and software, design and manufacturing worlds. With the rapid pace of innovation, digitalization is no longer tomorrow's idea. We take what the future promises tomorrow and make it real for our customers today. Where today meets tomorrow. Our culture encourages creativity, welcomes fresh thinking and focuses on growth, so our people, our business, and our customers can achieve their full potential.

Support Center: support.sw.siemens.com

Send Feedback on Documentation: support.sw.siemens.com/doc_feedback_form

Contents

Install Teamcenter Quality applications using Deployment Center

Install Quality Project Management and Quality Action Management using Deployment Center	1-1
Install FMEA using Deployment Center	1-2
Install Control Plan using Deployment Center	1-4
Install Problem Solving and Quality Issue using Deployment Center	1-6
Install Quality Audit using Deployment Center	1-8
Install Training and Qualification using Deployment Center	1-10

Install Teamcenter Quality applications using TEM

Install Quality Project Management and Quality Action Management using TEM	2-1
Install FMEA using TEM	2-2
Install Control Plan using TEM	2-4
Install Problem Solving and Quality Issue using TEM	2-6
Install Quality Audit using TEM	2-9
Install Training and Qualification using TEM	2-11

Managing the Quality Management workspace

About the Quality Management workspace	3-1
Enable the Quality Management workspace for Quality applications	3-2

Upgrade FMEA

Upgrading FMEA	4-1
----------------	-----

Configure FMEA

Configuring FMEA	5-1
Configure the creation of FMEA variants	5-1
About the configuration file for creating FMEA variants	5-1
Modifying the configuration file for creating FMEA variants	5-2
Configure the comparing and alignment of master and variant FMEAs	5-14
About the configuration file for comparing and aligning master and variant FMEAs	5-14
Modifying the configuration file for comparing and aligning master and variant FMEAs	5-14
Configure FMEA guideline objects	5-20
About FMEA guideline objects	5-20
View the guideline values for action priority	5-21
Edit the guideline values for action priority	5-21
Attach the configuration file to the guideline object	5-23
Configure the risk matrix	5-23

Modify the risk priority value in the risk matrix	5-23
Modifying the configuration file for the Net View Settings panel	5-24
Customize the FMEA formsheet	5-27
Customizing formsheets	5-27
FMEA formsheet elements	5-28
Customization tasks using the styling file	5-66
Customization tasks using the configuration file	5-75
Configure the company logo in the formsheet header	5-89
Customize the visual indicators for the Classification list of values in a Quality Characteristic for an FMEA	5-90
Set up and view audit logs for FMEA objects	5-91
Allow editing of occurrence and detection values for actions in an action group	5-91
Enable automatic carry over of occurrence and detection values from the initial state actions to the revision state actions	5-92
Specify the display of formsheets	5-92
Enable cascading of severity values in an FMEA	5-93
Enable a multi-line display of an FMEA object name	5-93
FMEA workflow handlers	5-93
QFMOFMEAREALIZATION-release-fmea-structure	5-93
QFMOFMEAREALIZATION-validate-for-class	5-94

Administer FMEA

Importing FMEA structures to Teamcenter	6-1
Import an FMEA structure	6-2
MSR FMEA elements and attribute definitions	6-3

Configure Quality Actions

Configuring Quality Actions	7-1
Set up dispatcher to trigger reminders and escalation workflows for quality actions	7-1
Dispatcher settings for triggering reminders and escalation	7-1
Start the dispatcher service to run the escalationmgmt translator	7-1
Start the escalationmgmt service to trigger reminders and escalation workflows for quality actions	7-2
Set up email notifications	7-3
View audit logs for modified quality actions	7-5
View the status of all quality actions	7-6
Set a preference to add external users to a quality action	7-7
Enable publishing and transferring ownership of actions and reminders to other sites	7-7

Configure Quality Project Management

Configuring Quality Project Management	8-1
View the context for quality actions in Quality Project Management	8-1

Configure the workflows for quality actions in Quality Project Management	8-1
Customize RYG rating behavior	8-2
About customizing RYG rating behavior in Quality Project Management	8-2
Customize the RYG rating behavior in Quality Project Management	8-4
View audit logs for modified RYG ratings	8-5
View the status of all quality projects and checklists	8-6
Enable publishing and transferring ownership of checklists and checklist questions to other sites	8-6

Configure Quality Audit

Specify the answer and the rating of a checklist based on the finding guideline	9-1
Set a preference for co-auditor's qualification	9-1
View audit logs for modified quality audit objects	9-2
Enable user access to the audits created from audit templates	9-2
Enable publishing and transferring ownership of audits and audit plans to other sites	9-4

Configure Problem Solving and Quality Issue

Disable the Due Date verification rule for associated quality actions and checklists	10-1
Configure the fonts used in reports	10-1
Display your company logo in Problem Solving and issue reports	10-2
Configure the duration assigned to the Proposed Due Date for a specific priority	10-5
Create the Problem Solving checklist master data	10-6
Update the transfer mode to use the custom objects when exporting objects	10-7
Configure reminders and trigger escalation workflows for Problem Solving	10-7
Migrate the root cause defects linked to corrective actions	10-10
Configure the Confidentiality Level of a Problem Solving process	10-11
Disable the risk calculation of a Problem Solving process	10-13
Define the type of contacts and locations that can be assigned to a Problem Solving process	10-13
Configure which pages appear in the Quality Management workspace header	10-14
Configure the Filters panel and Chart by list	10-16
Configure the types of users in the My Roles property of the Filters panel and the Chart by list	10-17
Allow External Users to view details of the assigned Problem Solving process	10-19
Business objects used to plan a Problem Solving process or an issue	10-20

Configure which business objects can be used to plan a Problem Solving process or an issue	10-20
View and set up audit logs	10-22
Configure the availability of related Problem Solving processes	10-23
Configure the display of a Microsoft Excel report	10-23
Configure Problem Solving to display localized content	10-23
Update the COTS workflow templates using the PSP-derive-problem-solving-items workflow handler	10-25

Configure Control Plan

Configure the properties displayed while comparing PMI for a modified balloon	11-1
Customize the style sheet used to print the Control Plan	11-2
Define the size and resolution of images added to Visual Quality Characteristics	11-3
Configure the name pattern for Inspection Definitions	11-4
Define the mapping between the properties of PMI balloons and Inspection Definitions	11-5
Define the mapping between the properties of PMI balloons and Quality Characteristics	11-6
Enable creating Quality Characteristics from parts with product and manufacturing information	11-10
Configure which rule file is used to generate, compare, and align a Control Plan with FMEA	11-10
Configure the default Tolerance Type for Variable Quality Characteristics	11-11
Configure the relation of a drawing with its attached IPXML file	11-11
Define which groups can view reference details of an Inspection Definition	11-11
Configure the mode of importing PMI balloons for a quality engineer	11-12
Customize the visual indicators for the Classification list of values in a Quality Characteristic	11-12

Configure Training and Qualification

Define the access privileges for Training and Qualification	12-1
Assign designated users to manage Training and Qualification data	12-4
Create the Training and Qualification checklist master data	12-4
Enable quality engineers to add reference documents to a qualification unit	12-5
Configuring preferences for custom Training and Qualification objects	12-6

1. Install Teamcenter Quality applications using Deployment Center

Install Quality Project Management and Quality Action Management using Deployment Center

Add the Quality Project Management application to your existing Teamcenter environment.

Prerequisites


Microservice framework is installed. For more information about the microservice framework, see *Microservices and the microservice framework* in *Teamcenter Installation Using Deployment Center* in the Teamcenter documentation.

Procedure

To install Quality Project Management and Quality Action Management using Deployment Center, do the following:

1. Log on to Deployment Center and select the environment to which you want to add Quality Project Management and Quality Action Management.

This procedure also assumes that you are familiar with Deployment Center. For more information about using Deployment Center, see *Deployment Center — Usage* in the Deployment Center documentation.

2. Go to the **Applications** task. Click **Add or Remove Selected Applications** .
3. In the **Available Applications** panel, use the web browser search to find the following applications, select the applications, and then click **Update Selected Applications**.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Base**
- **Teamcenter > Teamcenter > Program Planning**
- **Teamcenter > Teamcenter > Schedule Manager**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Action Management**

This template is mandatory for all Teamcenter Quality solutions.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Manager**

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Project Management(APQP)**

Deployment Center automatically selects any additional dependent applications.

4. In the **Selected Components** list of the **Components** task, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

When all components are fully configured, the **Deploy** task is enabled.

5. In the **Deploy** task, to generate deployment scripts, click **Generate Install Scripts**.

When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

6. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

For more information about running deployment scripts, see *Run the deployment scripts* in the Deployment Center documentation.

If you face any issues during the deployment, see *Troubleshoot the deployment script* in the Deployment Center documentation.

Install FMEA using Deployment Center

Add the FMEA application to your existing Teamcenter environment.

Prerequisites

Microservice framework is installed. For more information about the microservice framework, see *Microservices and the microservice framework* in *Teamcenter Installation Using Deployment Center* in the Teamcenter documentation.

Procedure

To install FMEA using Deployment Center, do the following:

1. Log on to Deployment Center and select the environment to which you want to add FMEA.

This procedure also assumes that you are familiar with Deployment Center. For more information about using Deployment Center, see *Deployment Center — Usage* in the Deployment Center documentation.

2. Go to the **Applications** task. Click **Add or Remove Selected Applications** .

3. In the **Available Applications** panel, use the web browser search to find the following applications, select the applications, and then click **Update Selected Applications**.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Base**
- **Teamcenter > Teamcenter > Teamcenter Foundation**
- **Teamcenter > Teamcenter > Active Content**
- **Teamcenter > Teamcenter > Active Workspace Base**
- **Teamcenter > Teamcenter > Extensions > Active Workspace User Management**

This template configures the Teamcenter installation for managing the admin objects, such as Group, Role and User, and its related behavior.

- **Teamcenter > Teamcenter > Extensions > Dispatcher**

This enables Teamcenter users to manage job distribution and execution.

- **Teamcenter > Teamcenter > Client Configuration**
- **Teamcenter > Teamcenter > Content and Document Management > Document Management Client Application**
- **Teamcenter > Teamcenter > Relationship Browser**

This template provides the ability to view relations in an FMEA.

- **Teamcenter > Teamcenter > Reporting**

This template provides the ability to view and generate reports in an FMEA.

- **Teamcenter > Teamcenter > Schedule Manager**

This template provides the ability to integrate Schedule Manager with FMEA.

- **Teamcenter > Teamcenter > Subscription**

This template provides the ability to use the subscriptions in FMEA.

- **Teamcenter > Teamcenter > Workflow**

This template provides the ability to use workflows in FMEA.

- **Teamcenter > Teamcenter > XRT Editor**

This template provides the ability to configure formsheets.

- **Teamcenter > Teamcenter > Teamcenter Quality > Failure Mode and Effects Analysis (FMEA)**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Action Management**

This template is mandatory for all Teamcenter Quality solutions.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Manager**

Deployment Center automatically selects any additional dependent applications.

4. In the **Selected Components** list of the **Components** task, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

When all components are fully configured, the **Deploy** task is enabled.

5. In the **Deploy** task, to generate deployment scripts, click **Generate Install Scripts**.

When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

6. Locate deployment scripts, copy each script to its target machine, and then run each script on its target machine.

For more information about running deployment scripts, see *Run the deployment scripts* in the Deployment Center documentation.

If you face any issues during the deployment, see *Troubleshoot the deployment script* in the Deployment Center documentation.

Install Control Plan using Deployment Center

Add the Control Plan application to your existing Teamcenter environment through a series of tasks from selecting the application and entering configuration parameters to generating and running deployment scripts.


Prerequisites

- Microservice framework is installed. For more information about the microservice framework, see *Microservices and the microservice framework* in *Teamcenter Installation Using Deployment Center* in the Teamcenter documentation.

- This procedure also assumes that you are familiar with Deployment Center. For more information about using Deployment Center, see *Deployment Center — Usage* in the Deployment Center documentation.

Procedure

To install Control Plan using Deployment Center, do the following:

1. Log on to Deployment Center and select the environment to which you want to add Control Plan.
2. Go to the **Applications** tab. Click **Add or Remove Selected Applications** .
3. In the **Available Applications** panel, use the web browser search to find the following applications, select the applications, and then click **Update Selected Applications**.

- **Teamcenter > Teamcenter > Active Content**

This template provides the ability to display the item structure.

- **Teamcenter > Teamcenter > Content and Document Management > Document Management Client Application**
- **Teamcenter > Teamcenter > Control and Inspection Plan Data Model**
- **Teamcenter > Teamcenter > Dimensional Planning and Validation**
- **Teamcenter > Teamcenter > Document Management**
- **Teamcenter > Teamcenter > Manufacturing Process Management > Customization for eM-Server Integration**
- **Teamcenter > Teamcenter > Manufacturing Process Management > Database Configuration for DPV**
- **Teamcenter > Teamcenter > Manufacturing Process Management > Manufacturing Characteristics Information**
- **Teamcenter > Teamcenter > Manufacturing Process Management > Work Instructions**
- **Teamcenter > Teamcenter > Reporting**

This template provides the ability to view and generate reports in a Control Plan.

- **Teamcenter > Teamcenter > Teamcenter Quality > Control and Inspection Plan**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Action Management**

This template is mandatory for all Teamcenter Quality solutions.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Base**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Manager**

Deployment Center automatically selects any additional dependent applications.

4. In the **Selected Components** list of the **Components** tab, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

When all components are fully configured, the **Deploy** tab is enabled.

5. In the **Deploy** tab, to generate deployment scripts, click **Generate Install Scripts**.

When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

6. Locate the deployment scripts, copy each script to its target machine, and then run each script on its target machine.

For more information about running deployment scripts, see *Run the deployment scripts* in the Deployment Center documentation.

If you face any issues during the deployment, see *Troubleshoot the deployment script* in the Deployment Center documentation.

Install Problem Solving and Quality Issue using Deployment Center


Add the Problem Solving and Quality Issue applications to your existing Teamcenter environment.

Prerequisites

- Microservice framework is installed. For more information about the microservice framework, see *Microservices and the microservice framework* in *Teamcenter Installation Using Deployment Center* in the Teamcenter documentation.
- This procedure also assumes that you are familiar with Deployment Center. For more information about using Deployment Center, see *Deployment Center — Usage* in the Deployment Center documentation.

Procedure

To install Problem Solving and Quality Issue using Deployment Center, do the following:

1. Log on to Deployment Center and select the environment to which you want to add Problem Solving and Quality Issue.
2. Go to the **Applications** tab. Click **Add or Remove Selected Applications** .
3. In the **Available Applications** panel, use the web browser search to find the following applications, select the applications, and then click **Update Selected Applications**.

- **Teamcenter > Teamcenter > Active Content**

This template provides the ability to display the item structure.

- **Teamcenter > Teamcenter > Enterprise Knowledge Foundation > Change Management**

- **Teamcenter > Teamcenter > Enterprise Knowledge Foundation > Issue Management**

- **Teamcenter > Teamcenter > Supplier Relationship Management > Vendor Management**

- **Teamcenter > Teamcenter > Quality Issue Management and Problem Solving base**

- **Teamcenter > Teamcenter > Retail Footwear and Apparel**

- **Teamcenter > Teamcenter > Softlines, Hardlines and Footwear**

- **Teamcenter > Teamcenter > Relationship Browser**

This template provides the ability to view the relationships between objects in a Problem Solving process.

- **Teamcenter > Teamcenter > Reporting**

This template provides the ability to view and generate reports in a Problem Solving process.

- **Teamcenter > Teamcenter > Schedule Manager**

This template provides the ability to integrate Schedule Manager with Problem Solving.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Action Management**

This template is mandatory for all Teamcenter Quality solutions.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Base**

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Issue Management and Problem Solving**

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Issue Management and Problem Solving search**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Management for Life Sciences Industry**

This template provides the ability to create Problem Solving processes and Non-Conformance issues for the Life Sciences industry in the web client.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Manager**
- **Teamcenter > Teamcenter > Teamcenter Quality > Supplier Quality Management**

To use **Supplier Quality Management**, you must select all features of **Quality Issue, Problem Solving, Vendor Management**, and **APQP**.

Deployment Center automatically selects any additional dependent applications.

4. In the **Selected Components** list of the **Components** tab, note any remaining components whose configuration status is not **100%**. Select each incomplete component, enter required parameters, and save component settings until all components in the environment show a configuration status of **100%**.

When all components are fully configured, the **Deploy** tab is enabled.

5. In the **Deploy** tab, to generate deployment scripts, click **Generate Install Scripts**.

When script generation is complete, note any special instructions in the **Deploy Instructions** panel.

6. Locate the deployment scripts, copy each script to its target machine, and then run each script on its target machine.

For more information about running deployment scripts, see *Run the deployment scripts* in the Deployment Center documentation.

If you face any issues during the deployment, see *Troubleshoot the deployment script* in the Deployment Center documentation.

Install Quality Audit using Deployment Center

Add the Control Plan application to your existing Teamcenter environment.

Prerequisites

Microservice framework is installed. For more information about the microservice framework, see *Microservices and the microservice framework* in *Teamcenter Installation Using Deployment Center* in the Teamcenter documentation.

Procedure

To install Quality Audit using Deployment Center, do the following:

1. Log on to Deployment Center and select the environment to which you want to add Quality Audit.

This procedure also assumes that you are familiar with Deployment Center. For more information about using Deployment Center, see *Deployment Center — Usage* in the Deployment Center documentation.

2. Select the following applications in the **Applications** task, and then click **Update Selected Applications**.

- **Teamcenter > Teamcenter > Enterprise Knowledge Foundation > Change Management**
- **Teamcenter > Teamcenter > Enterprise Knowledge Foundation > Issue Management**
- **Teamcenter > Teamcenter > Extensions > Program Planning Infrastructure**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Base**
- **Teamcenter > Teamcenter > Quality Issue Management and Problem Solving base**
- **Teamcenter > Teamcenter > Active Content**
- **Teamcenter > Teamcenter > Program Planning**
- **Teamcenter > Teamcenter > Schedule Manager**
- **Teamcenter > Teamcenter > Teamcenter Quality > Audit and Assessment**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Action Management**

This template is mandatory for all Teamcenter Quality solutions.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Issue Management and Problem Solving**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Manager**
- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Project Management(APQP)**

3. In the **Components** task, for each component, specify the parameters until the **Complete** column shows a configuration status of **100%**.
4. In the **Deploy** task, to generate deployment scripts, click **Generate Install Scripts**.
5. To deploy the components on the target machines, follow the instructions in the **Deploy Instructions** pane.

If you face any issues during the deployment, see *Troubleshoot the deployment script* in the Deployment Center documentation.

Install Training and Qualification using Deployment Center

Prerequisites

Microservice framework is installed. For more information about the microservice framework, see *Microservices and the microservice framework* in *Teamcenter Installation Using Deployment Center* in the Teamcenter documentation.

Procedure

To install Training and Qualification using Deployment Center, do the following:

1. Log on to Deployment Center.

This procedure also assumes that you are familiar with Deployment Center. For more information about using Deployment Center, see *Deployment Center — Usage* in the Teamcenter documentation.

2. Select the following applications in the **Applications** task, and then click **Update Selected Applications**.

- **Teamcenter > Teamcenter > Active Content**

This template provides the ability to display the item structure.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Action Management**

This template is mandatory for all Teamcenter Quality solutions.

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Base**

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Manager**

- **Teamcenter > Teamcenter > Teamcenter Quality > Quality Project Management (APQP)**

- **Teamcenter > Teamcenter > Teamcenter Quality > Training and Qualification**

3. In the **Components** task, for each component, specify the parameters until the **Complete** column shows a configuration status of **100%**.
4. In the **Deploy** task, to generate deployment scripts, click **Generate Install Scripts**.
5. To deploy the components on the target machines, follow the instructions in the **Deploy Instructions** pane.

If you face any issues during the deployment, see *Troubleshoot the deployment script* in the Deployment Center documentation.

2. Install Teamcenter Quality applications using TEM

Install Quality Project Management and Quality Action Management using TEM

Prerequisites

Teamcenter, including the microservice framework, are installed. For more information about the microservice framework, see *Installing Teamcenter microservices in Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM* in the Teamcenter documentation.

To install Quality Project Management and Quality Action Management using TEM, do the following:

1. Stop all Teamcenter services except FSC services before beginning installation.
2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.
6. In the **Feature Maintenance** panel, select **Add/Remove Features**.
7. Select the required features:
 - a. Expand **Base Install > Active Workspace > Server Extensions > Teamcenter Quality data model**.
 - b. Select the following server extensions features for Quality Project Management:

Feature name	Description
Quality Action Management data model	Adds support for quality actions. This feature is mandatory for all Teamcenter Quality solutions.
APQP Program Management data model	Adds support for Quality Project Management methodology to Teamcenter Quality.
Escalation Management Translator	Adds support for users or quality managers to receive alerts or e-mail notifications to remind them of their quality actions prior to the due date.

Tip:

To search for a feature by name, type the name or a partial name in the search box, and then click the search icon.

- c. Expand **Base Install > Active Workspace > Client > Teamcenter Quality**.
- d. Select the following client features for Quality Project Management:

Feature name	Description
Quality Action Management	Adds support for quality actions to the web client.
APQP Program Management	Adds support for Quality Project Management methodology to the web client.

- e. Expand **Base Install > Active Workspace > Client > Program Planning Execution Client**.
- f. Select the following client features for Quality Project Management:

Feature name	Description
Program Schedule Manager Client	Adds support for Program Planning, Schedule Manager, and Quality Project Management to the web client.

- 8. Continue the installation as described in *Install patches on a Teamcenter server in Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

Install FMEA using TEM

Prerequisites

Teamcenter, including the microservice framework, are installed. For more information about the microservice framework, see *Installing Teamcenter microservices in Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM* in the Teamcenter documentation.

To install FMEA using TEM, do the following:

1. Stop all Teamcenter services except FSC services before beginning installation.
2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.
6. In the **Feature Maintenance** panel, select **Add/Remove Features**.
7. Select the **Extensions > Teamcenter Quality platform > Quality Base** feature.
8. Select the required features:
 - a. Expand **Base Install > Active Workspace > Server Extensions > Teamcenter Quality data model**.
 - b. Select the following server extensions features for FMEA:

Feature name	Description
Quality Action Management	Adds support for quality actions. This feature is mandatory for all Teamcenter Quality solutions.
Quality Manager data model	Adds support for FMEA and Characteristics to the web client.
Failure Mode Effect and Analysis (FMEA) data model	Adds support for Failure Mode and Effects Analysis (FMEA).

Tip:

To search for a feature by name, type the name or a partial name in the search box, and then click the search icon.

- c. Ensure that you have selected the **Base Install > Active Workspace > Client > Workflow** feature.
- d. Expand **Base Install > Active Workspace > Client > Teamcenter Quality**.
- e. Select the following client features for FMEA:

Feature name	Description
Quality Action Management Client	Adds support for quality actions to the web client.
Quality Manager	Adds support for FMEA and Characteristics to the web client.
Failure Mode Effect and Analysis (FMEA)	Adds support for Failure Mode and Effects Analysis (FMEA) standards to the web client.

9. Continue the installation as described in *Install patches on a Teamcenter server in Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

Install Control Plan using TEM

Prerequisites

Teamcenter, including the microservice framework, are installed. For more information about the microservice framework, see *Installing Teamcenter microservices in Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM* in the Teamcenter documentation.

To install Control Plan using TEM, do the following:

1. Stop all Teamcenter services except FSC services before beginning installation.
2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.
6. In the **Feature Maintenance** panel, select **Add/Remove Features**.

7. Expand **Extensions** and select the following features:
- **Teamcenter Quality platform > Quality Base**
 - **Teamcenter Quality platform > Control and Inspection Plan Data Model**
 - **Manufacturing Process Management > Customization for eM-Server Integration**
 - **Manufacturing Process Management > Database Configuration for DPV**
 - **Manufacturing Process Management > Manufacturing Characteristics Information**
 - **Manufacturing Process Management > Work Instructions**
8. Select the required features:
- a. Expand **Base Install > Active Workspace > Server Extensions**, and select the following features:

Feature name	Description
Active Content Structure	Adds support for Control Plan to display the item structure.
Active Workspace Document Management	Adds support to print a Control Plan.
DPV Active Workspace	Adds support to generate a DPV workbook from an operation of a Control Plan.
Reporting	Adds support to view and generate reports in a Control Plan.

- b. Expand **Base Install > Active Workspace > Server Extensions > Teamcenter Quality data model**, and select the following features:

Feature name	Description
Quality Action Management data model	Adds support for quality actions. This feature is mandatory for all Teamcenter Quality solutions.
Quality Manager data model	Adds support for Control and Inspection Plan and Characteristics.
Control and Inspection Plan	Adds support for Control Plan to Teamcenter Quality.
Quality Manager	Adds support for Characteristics.

- c. Expand **Base Install > Active Workspace > Client > Teamcenter Quality**, and select the following features:

Feature name	Description
Control and Inspection Plan Client	Adds support for Control Plan to the web client. Control and inspection planning allows you to manage critical characteristics of Failure Mode Effect Analysis (FMEA) and create a control plan that generates bill of process (BOP) elements.
Quality Action Management Client	Adds support for quality actions to the web client.
Quality Manager	Adds support for Control and Inspection Plan and Characteristics to the web client.

- d. Expand **Base Install > Active Workspace > Client** and select the following features:

Feature name	Description
Document Management Client	Adds support to print a Control Plan.
Reporting	Adds support to view and generate reports in a Control Plan.

9. Continue the installation as described in *Install patches on a Teamcenter server in Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

Install Problem Solving and Quality Issue using TEM

Prerequisites

1. Teamcenter, including the microservice framework, are installed. For more information about the microservice framework, see *Installing Teamcenter microservices in Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM* in the Teamcenter documentation.
2. (Optional) Retail Footwear and Apparel patch is installed if you want to integrate it with Problem Solving.

For more information about the installing the patch, see *Retail Footwear and Apparel — Deployment* in the Teamcenter documentation.

Procedure

To install Problem Solving and Quality Issue using TEM, do the following:

1. Stop all Teamcenter services *except* FSC services before beginning installation.
2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.
6. In the **Feature Maintenance** panel, select **Add/Remove Features**.
7. Expand **Extensions** → **Enterprise Knowledge Foundation** and select the following:
 - **Change Management**
 - **Issue Management**
8. Select the **Extensions > Supplier Relationship Management > Vendor Management** feature.
9. Expand **Extensions > Teamcenter Quality platform** and select the following:
 - **Quality Base**
 - **Quality Issue Management and Problem Solving base**
10. Select the required features:
 - a. Expand the **Base Install > Active Workspace > Server Extensions**, and select the following features:
 - **Active Workspace**
 - **Active Content Structure**
 - **Change Management**
 - **Relationship Viewer**
 - **Reporting**

- (Optional) **Vendor Management**
 - (Optional) **Retail Footwear and Apparel**
 - (Optional) **Softlines, Hardlines and Footwear**
- b. Expand **Base Install > Active Workspace > Server Extensions > Teamcenter Quality data model**, and select the following features:

Feature name	Description
Quality Action Management data model	Adds support for quality actions. This feature is mandatory for all Teamcenter Quality solutions.
Quality Manager data model	Adds Quality Manager to the web client.
Quality Issue Management and Problem Solving data model	Adds support for Problem Solving and Quality Issue to the web client.
Quality Issue Management and Problem Solving search	Adds the ability to search for Problem Solving objects in the web client.
Quality Management for Life Sciences Industry	Adds the ability to create Problem Solving processes and Non-Conformance issues for the Life Sciences industry in the web client.
Supplier Quality Management	Adds the ability to manage the Quality data of the vendors assigned to issues and Problem Solving processes in the web client. To use Supplier Quality Management , you must select all features of Quality Issue, Problem Solving, Vendor Management , and APQP .

Tip:

To search for a feature by name, type the name or a partial name in the search box, and then click the search icon.

- c. Expand **Base Install > Active Workspace > Client > Teamcenter Quality**, and select the following features:

Feature name	Description
Quality Action Management Client	Adds support for quality actions to the web client.
Quality Manager	Adds support for Problem Solving and Quality Issue to the web client.

Feature name	Description
Quality Issue Management and Problem Solving	Adds support for Problem Solving and Quality Issue to the web client.
Quality Management for Life Sciences Industry	Adds the ability to create Problem Solving processes and Non-Conformance issues for the Life Sciences industry in the web client.
Supplier Quality Management	Adds the ability to manage the Quality data of the vendors assigned to issues and Problem Solving processes in the web client. To use Supplier Quality Management , you must select all features of Quality Issue, Problem Solving, Vendor Management , and APQP .

- d. Expand **Base Install > Active Workspace > Client** and select the following features:

Feature name	Description
Reporting	Adds support to view and generate reports in a Problem Solving process.
Relationship Browser Client	Adds support to view the relationships between objects in a Problem Solving process.
(Optional) Schedule Manager	Adds support to integrate Schedule Manager with Problem Solving.
(Optional) Vendor Management	Adds support to add project, schedule, supplier, and customer information to the plan of a Problem Solving process or an issue.
(Optional) Retail Footwear and Apparel	Adds support to integrate Retail Footwear and Apparel with Problem Solving.

11. Continue the installation as described in *Install patches on a Teamcenter server in Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

Install Quality Audit using TEM

Prerequisites

Teamcenter, including the microservice framework, are installed. For more information about the microservice framework, see *Installing Teamcenter microservices in Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM* in the Teamcenter documentation.

To install Quality Audit using TEM, do the following:

1. Stop all Teamcenter services except FSC services before beginning installation.
2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.
6. In the **Feature Maintenance** panel, select **Add/Remove Features**.
7. Ensure that the following features are selected.
 - **Teamcenter Foundation** or **Server Manager** or **BMIDE 2tier** or **Teamcenter Rich Client**
 - **Active Workspace**
 - **Change Management**
 - **Program Planning Infrastructure**
 - **Program Planning for Active Workspace**
 - **Schedule Manager for Active Workspace**
 - **Program Change**
 - **Program Planning Event Change**
 - **Program Schedule Manager**
 - **Issue Management**
 - **Quality Issue Management and Problem Solving base**
 - **Quality base**
 - **Active Content Structure**
8. Select the required features.
 - a. Expand **Base Install > Active Workspace > Server Extensions > Teamcenter Quality data model**.

- b. Select the following server extensions features.

Feature name	Description
Quality Action Management data model	Adds support for quality actions. This feature is mandatory for all Teamcenter Quality solutions.
Quality Manager data model	Adds Quality Manager to the web client.
Quality Issue Management and Problem Solving data model	Adds support for Issue Manager and Problem Solving.
APQP Program Management data model	Adds support for Quality Project Management methodology.
Quality Audit	Adds support for quality auditing.

Tip:

To search for a feature by name, start typing the name in the search box, and click the search icon.

- c. Expand **Base Install > Active Workspace > Client > Teamcenter Quality**.
- d. Select the following client features for Quality Audit:

Feature name	Description
Quality Audit	Adds support for quality auditing to the web client.

9. Continue the installation as described in *Install patches on a Teamcenter server in Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

Install Training and Qualification using TEM

Prerequisites

Teamcenter, including the microservice framework, are installed. For more information about the microservice framework, see *Installing Teamcenter microservices in Teamcenter Installation on Windows Using TEM* and *Teamcenter Installation on Linux Using TEM* in the Teamcenter documentation.

To install Training and Qualification using TEM, do the following:

1. Stop all Teamcenter services except FSC services before beginning installation.

2. Launch Teamcenter Environment Manager (TEM) for the corporate server to which Server Extensions features are to be added.
3. In the **Maintenance** panel, select **Configuration Manager**.
4. In the **Configuration Maintenance** panel, select **Perform maintenance on an existing configuration**.
5. In the **Old Configuration** panel, select the corporate server configuration.
6. In the **Feature Maintenance** panel, select **Add/Remove Features**.
7. Expand **Extensions > Teamcenter Quality platform**, and select **Quality Base**.
8. Select the required features:
 - a. Expand **Base Install > Active Workspace > Server Extensions > Active Content Structure**.
 - b. Expand **Base Install > Active Workspace > Server Extensions > Teamcenter Quality data model**.
 - c. Select the following server extensions features for Training and Qualification:

Feature name	Description
Quality Action Management data model	Adds support for quality actions. This feature is mandatory for all Teamcenter Quality solutions.
Quality Manager data model	Adds Quality Manager to the web client.
Training and Qualification	Adds support for Training and Qualification.

Tip:

To search for a feature by name, type the name or a partial name in the search box, and then click the search icon.

- d. Expand **Base Install > Active Workspace > Client > Teamcenter Quality**.
 - e. Select the **Training and Qualification Client** feature.
9. Continue the installation as described in *Install patches on a Teamcenter server in Teamcenter Upgrade Using TEM* in the Teamcenter documentation.

3. Managing the Quality Management workspace

About the Quality Management workspace

The **Quality Management** workspace is a workspace dedicated to working with Teamcenter Quality applications. In this workspace, you can do the following:

- Access the dashboards for all Quality applications including Quality master data. You can also search and navigate using global search or pin any additional objects or saved searches as required.
- Easily manage and access all your reports or create new ones as required.
- Open frequently accessed objects saved as favorites.
- View the notifications for any pending tasks assigned to you.
- Access other applications such as Assistant, Discussions, Programs, Schedules, and so on.

You must **make the Quality Management workspace available for your organization's groups and roles**.

For more information about workspaces, see *Learn about workspaces* in *Active Workspace Customization* of the Teamcenter documentation.

The screenshot displays the Teamcenter Quality interface. At the top, there's a navigation bar with 'Tc Teamcenter Quality', a search icon, 'No Active ECN', a notification bell, and the SIEMENS logo. Below this, the 'Quality Applications' section features four tiles: 'QUALITY ACTIONS', 'QUALITY MASTER DATA', 'CONTROL PLAN AND INSPECTION PLAN', and 'FMEA'. The 'Report Dashboard' section shows a bar chart titled 'All Quality Actions' with 'Total Objects: 96'. The chart displays the following data:

Status	Object Count
Draft	78
Active	9
Completed	4
Template	4
Confirmed	1

The 'Favorites' section on the right lists items like 'Gimbal Spindle', 'Air Intake System De...', and 'Car engine overheati...'. The 'Inbox' section shows tasks such as 'Requestor: Assign S...' and 'Active Quality Action Active Stat...'.

Enable the Quality Management workspace for Quality applications

The **AUDIT AND ASSESSMENT** tile in the **Quality Management** workspace is dedicated to working with audits and assessments. For more information about workspaces, see *Configuration and Extensibility* in the Teamcenter documentation.

To make this workspace available at your site, do the following:

1. Use the `export_wsconfig` utility to export workspace definitions from the Teamcenter command prompt. For information about running the `export_wsconfig` utility, see `export_wsconfig` in *Active Workspace Customization* of the Teamcenter documentation.

In the following example, all existing workspace definitions are exported.

```
export_wsconfig -u=<username> -p=<password> -g=<groupname> -file="c:/exportedWorkspacesConfig.xml"
```

2. In the exported file, search for **TcQualityManagementWorkspace** to edit the available users of the **Quality Management** workspace.
3. Specify the organization groups and roles that can access the **Quality Management** workspace.

For example, to enable the workspace for the **dba** and **Engineering** groups and their **DBA** and **Designer** roles, specify the following:

```
<Workspace id="TcQualityManagementWorkspace">
  <WorkspaceMapping group="dba" role="DBA"/>
  <WorkspaceMapping group="Engineering" role="Designer"/>
</Workspace>
```

```
hosts x exportedWorkspacesConfig.xml x
1 <?xml version="1.0" encoding="UTF-8" standalone="no" ?>
2 <Import>
3
4 <Workspace id="TcQualityManagementWorkspace">
5   <WorkspaceMapping group="dba" role="DBA"/>
6   <WorkspaceMapping group="Engineering" role="Designer"/>
7 </Workspace>
```

4. Use the `import_wsconfig` utility to import the updated `TcQualityManagementWorkspace` workspace definition. For information about running the `import_wsconfig` utility, see `import_wsconfig` in *Active Workspace Customization* of the Teamcenter documentation.

In the following example, you import the `exportedWorkspacesConfig.xml` custom file, which contains the `TcQualityManagementWorkspace` workspace definition.

```
import_wsconfig -u=<username> -p=<password> -g=<groupname> -file="c:/
exportedWorkspacesConfig.xml"
```


4. Upgrade FMEA

Upgrading FMEA

Note:

You need not perform the migration utility steps if you are upgrading from Active Workspace 6.1 to Teamcenter 2412.

You must manually run the following utilities after upgrading from a previous version to Teamcenter 2412:

- **qfm0_migrate_quality_actions**

In Active Workspace 6.0, the FMEA data model was updated for action groups. Therefore, after upgrading from a previous version to Active Workspace 6.0 or above, you must run the **qfm0_migrate_quality_actions** utility so that action group objects are created for previous actions inside the FMEA.

- **qfm0_cascade_severity**

In Active Workspace 6.1, the cascading of severity was introduced for the FMEA data model. However, the cascading operation does not happen automatically on all FMEAs created in Active Workspace 6.0 or an earlier version. Therefore, after upgrading from a previous version to Active Workspace 6.1, you must run the **qfm0_cascade_severity** utility for the FMEA where you want to cascade the severity value. You can run this utility to update one or more FMEAs by providing their UIDs.

- **qfm0_update_sequence**

Earlier, the order of objects in the FMEA tree structure was based on the creation date. In Active Workspace 6.1, the FMEA data model has a sequence property for all FMEA objects. After upgrade, the sequence property for all objects created in 6.0 or earlier will have a default value of 10. When you run this utility, the sequence property for the objects created in 6.0 or earlier will be updated in incremental values of 10 based on their creation date. You can run this utility to update one or more FMEAs by providing their UIDs.

Note:

The instructions about how to use the utilities is embedded in the utilities themselves. To obtain help for the utilities, in the Teamcenter command prompt, enter: *utility-name -h*.

Updating existing FMEA guideline objects during upgrade

While upgrading FMEA to Teamcenter 2412, the following default guideline objects are installed:

- IATF 16949 - Process FMEA
- IATF 16949 - Design FMEA
- AIAG & VDA - Process FMEA
- AIAG & VDA - Design FMEA

To update these out-of-the-box guideline objects, you must navigate to the `TC_ROOT\install\qfm0qcfmea\data` folder and manually replace the following styling file and configuration files with the new ones:

- *FormSheetStyling.css*
- *AIAG_VDA_DesignFormSheetConfiguration.xml*
- *AIAG_VDA_ProcessFormSheetConfiguration.xml*
- *IATF_16949_DesignFormSheetConfiguration.xml*
- *IATF_16949_ProcessFormSheetConfiguration.xml*

5. Configure FMEA

Configuring FMEA

FMEA helps quality engineers to predict different failure modes for a product design or a process at the initial stage itself so that appropriate actions can be taken for preventing them.

After you install FMEA, you can configure FMEA to modify the default out of the box behavior:

- Set up rules to **create, compare, and align** FMEA variants by modifying the configuration files.
- Customize the default options for FMEA objects such as severity, occurrence, detection, and action priority values by **modifying the FMEA guideline files**.
- **Override the calculated risk priority value in the risk matrix by modifying the value** as required.
- **Customize the format and layout of FMEA formsheets**.
- Modify the out-of-the-box workflows to **release** and **validate** FMEAs.

Configure the creation of FMEA variants

About the configuration file for creating FMEA variants

When users create a master FMEA and use it to create a variant FMEA, the *FMEACreateVariantRules.xml* configuration file located in the `%Root%\install\qfm0qcfmea\data` folder specifies the default attributes that can be copied from the master FMEA to the variant FMEA. You can modify this file to specify any additional attributes that can be copied, or modify the default ones.

After modification, you must import the *FMEACreateVariantRules.xml* file into the *FMEACreateVariantRule* dataset. The **Qfm0CreateVariantConfigFile** preference value defines the Dataset name that contains this configuration file. While importing a modified version of the *FMEACreateVariantRules.xml* file, you must make sure that the value of the **Qfm0CreateVariantConfigFile** preference and the value of the `-d` attribute in the command that you enter to import the *FMEACreateVariantRules.xml* file is exactly the same as shown.

Value of `-d` attribute in the import command

```
import_file -u=infodba -p=${TC_USER_PASSWD_FILE} -g=dba  
-f=${TC_INSTALL_DIR}/qfm0qcfmea/data/FMEACreateVariantRules.xml  
-d=FMEACreateVariantRule -ref=Text -type=Text -de=r
```

Value of the Qfm0CreateVariantConfigFile preference

```
<preference name="Qfm0CreateVariantConfigFile" type="String" array="false"
  disabled="false" protectionScope="Site" envEnabled="false">
  <preference_description>Defines Dataset name which holds Create Variant
  Rule file,which will be used in create variant operation.
  </preference_description>
  <context name="Teamcenter">
    <value>FMEACreateVariantRule</value>
  </context>
</preference>
```

Modifying the configuration file for creating FMEA variants

The `FMEACreateVariantRules.xml` file that lets you configure the creation of variant FMEAs from the master FMEA has the following element tags:

- FMEAVariantRule
- BusinessObject
- ResetAttributes
- Attribute
- References
- Reference

A sample configuration file is as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE FMEAVariantRule SYSTEM "FMEACreateVariantDTD.dtd">
<FMEAVariantRule>
  <BusinessObject name="Qfm0FMEANode">
    <ResetAttributes>
      <Attribute attributePropName="qfm0Id"
copyFromSource="true"/>
    </ResetAttributes>
    <References>
      <Reference relRefPropName="HasParticipant"
attachTypeName="GroupMember" copyType="CopyAsRefType"
isTargetPrimary="true"/>
    </References>
  </BusinessObject>
  <BusinessObject name="Qfm0FMEAElement">
    <ResetAttributes>
```

```

        <Attribute attributePropName="qfm0Id"
copyFromSource="false" />
    </ResetAttributes>
</BusinessObject>
</FMEAVariantRule>

```

You can modify the tags in this configuration file and specify the attributes that can be copied from the master FMEA to the variant FMEA.

FMEAVariantRule

This is the root element tag. All other element tags are under the root node as the child.

BusinessObject

This element tag is used for pointing to any business object (BO) on which the given configuration is applied during variant creation.

There can be no or multiple `BusinessObject` element tags inside the `FMEAVariantRule` element tag.

The following table lists the attributes for this tag.

Attribute name	Description
name	This is a mandatory attribute for <code>BusinessObject</code> tag and its value must match the name of any BO.

ResetAttributes

This element tag is used for all the child `<Attribute>` element tags for the current BO. Therefore, the `<Attribute>` tags corresponding to all the properties or attributes of the current BO that need to be reset during the variant creation are entered under `<ResetAttributes>` as a child. The `<ResetAttributes>` tag is optional. There can be either one or no `<ResetAttributes>` element tag under the `<BusinessObject>` tag as a child.

Attribute

This element tag is used for resetting a single property or attribute of the parent BO during variant creation.

Many `<Attribute>` tags can be created under the `<ResetAttributes>` tag as children, but at least one is mandatory.

The following table lists the attributes for this tag.

Attribute name	Description
attributePropName	This is a mandatory attribute for the <Attribute> tag and its value must match that of any property name of the current BO.
copyFromSource	<p>The value of this attribute can be set to <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p> <p>If the value is <code>true</code>, for the given property, the value must be the same as that received in input.</p> <p>If the value is <code>false</code>, the value must be reset for the given property.</p>

When there are no `ResetAttributes` for **Customer** (`qfm0Customer`) and **Product Family** (`qfm0ProductFamily`), by default, these values are copied from the master FMEA to the variant FMEA for the BO `Qfm0FMEANode`.

Example:

The sample code for a scenario where the attributes are copied from the master FMEA to the variant FMEA is as follows:

```
<BusinessObject name="Qfm0FMEANode">
  <ResetAttributes>
    <Attribute attributePropName="qfm0Id"
copyFromSource="true"/>
  </ResetAttributes>
</BusinessObject>
```


The following graphic shows the master FMEA.

ID:	FMEA-000000009
Version Information:	2
Name:	FMEA M1  Edit Local...
Description:	
Type:	FMEA
FMEA Category:	Master
FMEA Type:	Design
FMEA Guidelines:	AIAG & VDA - Design FMEA
Release Status:	TCM Released
Date Released:	24-Sep-2021
Owner:	ed (ed)
Group ID:	demo
Last Modifying User:	ed (ed)
Latest:	True

▼ PLANNING AND PREPARATION

Customer Name:	Customer A
Company Name:	Company ABC
Product Family:	Engine
Platform:	

The following graphic shows the variant FMEA where the attributes are copied from the master FMEA.

ID:	FMEA-000000010
Version Information:	1
Name:	FMEA Variant V1  Edit Local...
Description:	
Type:	FMEA
FMEA Category:	Variant
FMEA Type:	Design
FMEA Guidelines:	AIAG & VDA - Design FMEA
Release Status:	
Date Released:	
Owner:	ed (ed)
Group ID:	demo
Last Modifying User:	ed (ed)
Latest:	True

▼ PLANNING AND PREPARATION

Customer Name:	Customer A
Company Name:	Company ABC
Product Family:	Engine
Platform:	

When the `ResetAttributes` are present for **Customer** (`qfm0Customer`) and **Product Family** (`qfm0ProductFamily`), then these values are not copied from the master FMEA to the variant FMEA and are reset for the BO `Qfm0FMEANode`.

Example:

The sample code for a scenario where the attributes of the master FMEA are not copied to the variant FMEA is as follows:


```
<BusinessObject name="Qfm0FMEANode">
  <ResetAttributes>
    <Attribute attributePropName="qfm0Id"
copyFromSource="true"/>
    <Attribute attributePropName="qfm0Customer"
copyFromSource="false"/>
  </ResetAttributes>
</BusinessObject>
```

```

    <Attribute attributePropName="qfm0ProductFamily"
copyFromSource="false" />
  </ResetAttributes>
</BusinessObject>

```


The following graphic shows the master FMEA.

ID:	FMEA-000000009
Version Information:	2
Name:	FMEA M1  Edit Local...
Description:	
Type:	FMEA
FMEA Category:	Master
FMEA Type:	Design
FMEA Guidelines:	AIAG & VDA - Design FMEA
Release Status:	TCM Released
Date Released:	24-Sep-2021
Owner:	ed (ed)
Group ID:	demo
Last Modifying User:	ed (ed)
Latest:	True

▼ **PLANNING AND PREPARATION**

Customer Name:	Customer A
Company Name:	Company ABC
Product Family:	Engine
Platform:	

The following graphic shows the variant FMEA where the attributes are not copied.

ID:	FMEA-000000011
Version Information:	1
Name:	FMEA Variant V2  Edit Local...
Description:	
Type:	FMEA
FMEA Category:	Variant
FMEA Type:	Design
FMEA Guidelines:	AIAG & VDA - Design FMEA
Release Status:	
Date Released:	
Owner:	ed (ed)
Group ID:	demo
Last Modifying User:	ed (ed)
Latest:	True

▼ PLANNING AND PREPARATION

Customer Name:	
Company Name:	Company ABC
Product Family:	

References

All child `<Reference>` element tags for the current BO are under this tag. So, the `<Reference>` tags corresponding to all those relations or reference properties of the current BO for which the copy type needs to be provided explicitly during variant creation are inserted under the `<References>` tag as a child.

The `<References>` tag is optional. There can be either one or no `<References>` element tag under the `<BusinessObject>` tag.

Reference

This element tag is used for defining the copy type for a single relation or reference property of the parent BO during variant creation.

Multiple `<Reference>` tags can be created under the `<References>` tag as child, but at least one is mandatory.

The following table lists the attributes for this tag.

Attribute name	Description
relRefPropName	<p>This is a mandatory attribute.</p> <p>The value of this attribute must be the name of the relation or the reference property.</p>
attachTypeName	<p>This is a mandatory attribute.</p> <p>The value of this attribute must be the name of the other (or second) BO involved in relation or name of the BO being referenced by the reference property.</p>
copyType	<p>The value of this attribute must be set to one of the following:</p> <ul style="list-style-type: none"> <li data-bbox="483 625 792 653">• CopyAsObjectType <p>Set this value if you want to copy the given relation or reference property as an object. The new object instance will be created corresponding to the relation or reference property and the newly created variant will point to this new object instance corresponding to the relation or reference property.</p> <ul style="list-style-type: none"> <li data-bbox="483 915 740 942">• CopyAsRefType <p>Set this value if you want to copy the given relation or reference property as a reference. The newly created variant will point to the existing object instance corresponding to the relation or reference property.</p> <ul style="list-style-type: none"> <li data-bbox="483 1136 721 1163">• DontCopyType <p>Set this value if you do not want to copy the given relation or reference property. The DontCopyType attribute is the default value for the copyType attribute.</p>
isTargetPrimary	<p>The value of this attribute can be set to either true or false. The default value is true.</p> <ul style="list-style-type: none"> <li data-bbox="483 1444 581 1472">• true <p>The BO name given in the <BusinessObject> tag will be the primary object and the BO name provided for the attachTypeName attribute will be the secondary object with respect to the relation or reference property.</p> <ul style="list-style-type: none"> <li data-bbox="483 1665 597 1692">• false <p>The BO name given in the <BusinessObject> tag will be the secondary object and the BO name provided for the attachTypeName attribute will be the primary object with respect to the relation or reference property.</p>

If you want to create a variant from a master FMEA and copy the **Cross Function Team** information from the master FMEA to the variant FMEA for BO `Qfm0FMEANode`, set the `copyType` as `CopyAsRefType`. The relation and BO information corresponding to the **Cross Function Team** is as follows:

Relation BO Name: HasParticipant

Primary Object: Qfm0FMEANode


Secondary Object: GroupMember

Example:

The sample code for a scenario where the cross-functional team information is copied from the master FMEA to the variant FMEA is as follows:

```
<BusinessObject name="Qfm0FMEANode">
  <References>
    <Reference relRefPropName="HasParticipant"
attachTypeName="GroupMember"
copyType="CopyAsRefType" isTargetPrimary="true"/>
  </References>
</BusinessObject>
```

Master FMEA

Name: FMEA M1 

Description:

Type: FMEA

FMEA Category: Master

FMEA Type: Design

FMEA Guidelines: AIAG & VDA - Design FMEA

Release Status: TCM Released

Date Released: 27-Sep-2021

Owner: ed (ed)

Group ID: demo




Last Modifying User: ed (ed)



Latest: True

▶ PLANNING AND PREPARATION


▶ MASTER TEMPLATE

▼ CROSS-FUNCTIONAL TEAM

Selection Mode Select All  Export To...  Paste  Add to

Name	Role	Group	E-Mail Address	Phone Number	Organization	
 4GD_Test (4gd_test)	4G Designer	4G Tester				

The following graphic shows an example of the variant FMEA with the **Cross Functional Team** information copied.

Name: FMEA M1 

Description:

Type: FMEA

FMEA Category: Variant

FMEA Type: Design

FMEA Guidelines: AIAG & VDA - Design FMEA

Release Status: TCM Released

Date Released: 27-Sep-2021

Owner: ed (ed)

Group ID: demo




Last Modifying User: ed (ed)



Latest: True

▶ PLANNING AND PREPARATION

▶ MASTER TEMPLATE

▼ CROSS-FUNCTIONAL TEAM

Selection Mode Select All  Export To...  Paste  Add to

Name	Role	Group	E-Mail Address	Phone Number	Organization	
 4GD_Test (4gd_test)	4G Designer	4G Tester				

If you do not want to copy the cross functional team information from the master FMEA to the variant FMEA, set the `copyType` as `DontCopyType`. The relation and BO information corresponding to the Cross Function Team is as follows:

Relation BO Name: HasParticipant

Primary Object: Qfm0FMEANode


Secondary Object: GroupMember

Example:

The sample code for a scenario where the cross-functional team information is not copied from the master FMEA to the variant FMEA is as follows:

```
<BusinessObject name="Qfm0FMEANode">
  <References>
  <Reference relRefPropName="HasParticipant"
attachTypeName="GroupMember"
      copyType=" DontCopyType" isTargetPrimary="true"/>
  </References>
</BusinessObject>
```

Master FMEA

Name: FMEA M1 

Description:

Type: FMEA

FMEA Category: Master

FMEA Type: Design

FMEA Guidelines: AIAG & VDA - Design FMEA

Release Status: TCM Released

Date Released: 27-Sep-2021

Owner: ed (ed)

Group ID: demo




Last Modifying User: ed (ed)



Latest: True

▶ PLANNING AND PREPARATION


▶ MASTER TEMPLATE

▼ CROSS-FUNCTIONAL TEAM

Selection Mode Select All   

Name	Role	Group	E-Mail Address	Phone Number	Organization	
 4GD_Test (4gd_test)	4G Designer	4G Tester				

This is an example of a variant FMEA with the **Cross Functional Team** information not copied from the master FMEA.

Name: FMEA Variant Without Team 

Description:

Type: FMEA

FMEA Category: Variant

FMEA Type: Design

FMEA Guidelines: AIAG & VDA - Design FMEA

Release Status:

Date Released:

Owner: ed (ed)

Group ID: demo




Last Modifying User: ed (ed)


Latest: True

▶ PLANNING AND PREPARATION

▶ MASTER TEMPLATE

▼ CROSS-FUNCTIONAL TEAM

Select All   

Name	Role	Group	E-Mail Address	Phone Number	Organization	

Configure the comparing and alignment of master and variant FMEAs

About the configuration file for comparing and aligning master and variant FMEAs

FMEA users can compare FMEAs and align them if required. The *CompareFMEAStructureRules.xml* configuration file located in the `%Root%\install\qfm0qcfmea\data` folder specifies the rules and properties while comparing the source and target FMEAs and aligning them. You can modify this file to specify any additional attributes that can be aligned, or modify the default ones.

After modification, you must import the *CompareFMEAStructureRules.xml* file into the *CompareFMEAStructureRules* dataset. The **Qfm0CompareFMEAStructureConfigFile** preference value defines the *Dataset* name that contains this configuration file. While importing a modified version of the *CompareFMEAStructureRules.xml* file, you must make sure that the value of the **Qfm0CompareFMEAStructureConfigFile** preference and the value of the `-d` attribute in the command that you enter to import the *CompareFMEAStructureRules.xml* file is exactly the same as shown.

Value of `-d` attribute

```
import_file -u=infodba -p=${TC_USER_PASSWD_FILE} -g=dba
-f=${TC_INSTALL_DIR}/qfm0qcfmea/data/CompareFMEAStructureRules.xml
-d=CompareFMEAStructureRules -ref=Text -type=Text -de=r
```

Preference value

```
<preference name="Qfm0CompareFMEAStructureConfigFile" type="String" array="false"
  disabled="false" protectionScope="Site" envEnabled="false">
  <preference_description>Defines Dataset name which holds rule file,
  which will be used compare FMEA structure.</preference_description>
  <context name="Teamcenter">
    <value>CompareFMEAStructureRules</value>
  </context>
</preference>
```

Modifying the configuration file for comparing and aligning master and variant FMEAs

The *CompareFMEAStructureRules.xml* file lets you configure the comparison of variant FMEAs and the master FMEA. After comparing, the alignment is based on the properties that are compared as defined in the configuration file. This file has the following element tags:

- *CompareFMEAStructureRules*
- *CompareFMEAStructureProperties*

- `BusinessObject`
- `MetaProperties`
- `Property`

A sample configuration file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CompareFMEAStructureRules SYSTEM
"CompareFMEAStructureRulesDTD.dtd" >
<CompareFMEAStructureRules>
  <CompareFMEAStructureProperties>
    <BusinessObject objectName="WorkspaceObject">
      <MetaProperties>
        <Property name="object_name" />
        <Property name="object_desc" />
      </MetaProperties>
    </BusinessObject>
  </CompareFMEAStructureProperties>
</CompareFMEAStructureRules>
```

You can modify the tags in this configuration file and configure the comparison of variant FMEAs and the master FMEA.

CompareFMEAStructureRules

This is the root element tag. All other element tags are added under it as child.

CompareFMEAStructureProperties

This element tag is used for holding all `<BusinessObject>` element tags as children under it.

This tag is optional. There can be either one or no `<CompareFMEAStructureProperties>` element tags as children under the `<CompareFMEAStructureRules>` element tag.

BusinessObject

This element tag is used for pointing to any business object (BO) on which the given configuration is applied during comparison.

There can be none or multiple `<BusinessObject>` element tags inside the `<CompareFMEAStructureProperties>` element tag as a child.

For any BO corresponding to a `<BusinessObject>` tag, the given configuration is applied to all the child BOs (or derived BO classes) which inherit the given BO class.

The following table lists the attributes for this tag.

Attribute name	Description
objectName	This is a mandatory attribute for <code>BusinessObject</code> element tag and its value should be the name of any BO.

MetaProperties

This element tag is used for holding `<Property>` element tags under it as child.

It is a mandatory element tag, although there can be multiple `<MetaProperties>` tags as well as children under the `<BusinessObject>` tag.

Property

This element tag points to a single property of the current BO in the context of a compare scenario.

At least one `<Property>` tag is mandatory although there can be multiple `<Property>` tags as well as children for the `<MetaProperties>` tag.

Relation or reference type of properties are not supported. Simple property types such as numbers, string, date, and boolean are supported.

The following table lists the attributes for this tag.

Attribute name	Description
name	This is a mandatory attribute for the <code><Property></code> element tag and its value must match the name of any property of the current BO.
compare	<p>The value of this attribute can be set to <code>true</code> or <code>false</code>. The default value is <code>false</code>.</p> <p>If the value is <code>true</code>, for the given property, the value must be the same as that received in input.</p> <p>This attribute is optional. It can have the value as either <code>true</code> or <code>false</code>. The default value is <code>true</code>.</p> <p>If set to <code>true</code>, the property is considered as one of the parameters during the comparison. If the value of the property is same for both FMEAs, the difference is not generated, but if the values are different, this is reflected in the compare results as the difference type Property Modified.</p> <p>If set to <code>false</code>, the property is not considered as one of the parameters during comparison. The compare results are not affected regardless of whether the value of the property is same or different in two FMEAs.</p>

Example

To compare the FMEAs based on the value of the name (`object_name`) and description (`object_desc`) property for all the elements present in the FMEA tree structure, except for the system element (`Qfm0SystemElement`) objects, specify the following in the configuration file:

```

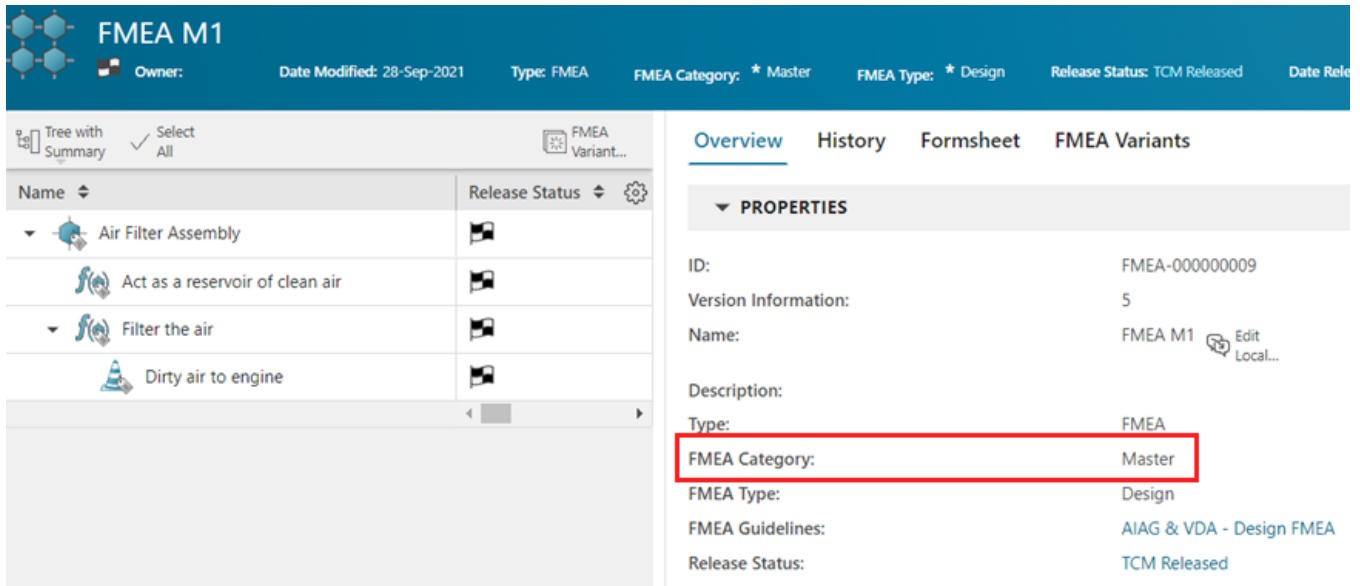
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE CompareFMEAStructureRules SYSTEM
"CompareFMEAStructureRulesDTD.dtd" >
<CompareFMEAStructureRules>
    <CompareFMEAStructureProperties>
        <BusinessObject objectName="WorkspaceObject">
            <MetaProperties>
                <Property name="object_name" />
                <Property name="object_desc" />
            </MetaProperties>
        </BusinessObject>
        <BusinessObject objectName="Qfm0SystemElement">
            <MetaProperties>
                <Property name="object_name" compare="false" />
                <Property name="object_desc" compare="false" />
            </MetaProperties>
        </BusinessObject>
    </CompareFMEAStructureProperties>
</CompareFMEAStructureRules>

```

The default value of the compare attribute for any `<Property>` element tag is `true`, so for this example, for all the `<Property>` element tags that do not include the compare attribute, the value for the compare attribute is automatically `true`.

For the first `<BusinessObject>` tag, the `objectName` is `WorkspaceObject`. Since all system element, function element, and failure element types of BOs are inherited from the `WorkspaceObject` BO, the property rules defined for the `WorkspaceObject` are applicable in all cases except those where a specific `<BusinessObject>` tag is present. In this example, the `<BusinessObject>` tag corresponds to `Qfm0SystemElement`, so for the `Qfm0SystemElement` objects, the rules defined for `Qfm0SystemElement` override the rules of `WorkspaceObject`.

The following graphic shows the master FMEA.



FMEA M1
 Owner: Date Modified: 28-Sep-2021 Type: FMEA FMEA Category: * Master FMEA Type: * Design Release Status: TCM Released Date Released:

Tree with Summary Select All FMEA Variant...

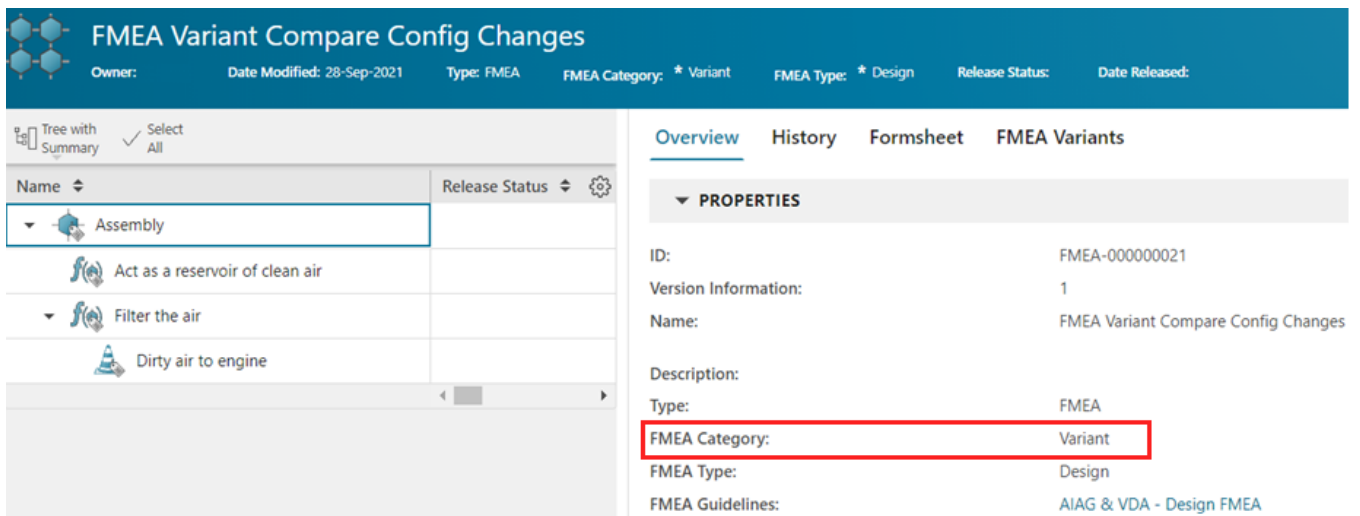
Name	Release Status
Air Filter Assembly	
Act as a reservoir of clean air	
Filter the air	
Dirty air to engine	

Overview History Formsheet FMEA Variants

PROPERTIES

ID: FMEA-00000009
 Version Information: 5
 Name: FMEA M1 Edit Local...
 Description:
 Type: FMEA
FMEA Category: Master
 FMEA Type: Design
 FMEA Guidelines: AIAG & VDA - Design FMEA
 Release Status: TCM Released

The following graphic shows the variant FMEA that is derived from the master FMEA.



FMEA Variant Compare Config Changes
 Owner: Date Modified: 28-Sep-2021 Type: FMEA FMEA Category: * Variant FMEA Type: * Design Release Status: Date Released:

Tree with Summary Select All

Name	Release Status
Assembly	
Act as a reservoir of clean air	
Filter the air	
Dirty air to engine	

Overview History Formsheet FMEA Variants

PROPERTIES

ID: FMEA-000000021
 Version Information: 1
 Name: FMEA Variant Compare Config Changes
 Description:
 Type: FMEA
FMEA Category: Variant
 FMEA Type: Design
 FMEA Guidelines: AIAG & VDA - Design FMEA

After variant creation, if you change the **Name** (object_name) of the system element to **Assembly** and the **Description** (object_desc) to **Assembly for air filter**, the compare result does not show any differences.

FMEA Variant Compare Config Changes
 Owner: ed (ed) Date Modified: 28-Sep-2021 Type: FMEA FMEA Category: * Variant FMEA Type: * Design Release Status: Date Released:

Tree with Summary Select All

Name	Release Status
Assembly	
Act as a reservoir of clean air	
Filter the air	
Dirty air to engine	

PROPERTIES

ID: SE-REP-00000025
 Version Information: 1
Name: Assembly Edit Local...
Description: Assembly for air filter Edit Local...
 Type: System Element Representation
 Context: FMEA Variant Compare Config Changes
 Release Status:

As there are no modifications to the function elements and failure elements, the difference summary does not show the updates in the compare results.

FMEA-Compare-000031/Difference summary of source and target FMEA
 Source FMEA: FMEA M1 Target FMEA: FMEA Variant Compare Config Changes

Tree Selection Mode Select All

Difference Type	Name	Context	Description

However, if you change the **Name** (object_name) of the function element to **Reservoir** and add **Dirty air is being passed to engine** as the **Description** (object_desc) to the failure element, the change is displayed in the difference summary.

FMEA Variant Compare Config Changes
 Owner: Date Modified: 28-Sep-2021 Type: FMEA FMEA Category: Variant FMEA Type: Design Release Status: Date Released:

Tree with Summary Select All

Name	Release Status
Assembly	
Reservoir	
Filter the air	
Dirty air to engine	

PROPERTIES

ID: FUN-REP-00000033
 Version Information: 1
Name: Reservoir Edit Local...
Description:

FMEA Variant Compare Config Changes
 Owner: ed (ed) Date Modified: 28-Sep-2021 Type: FMEA FMEA Category: Variant FMEA Type: Design Release Status: Date Released:

Tree with Summary Select All

Name	Release Status
Assembly	
Reservoir	
Filter the air	
Dirty air to engine	

PROPERTIES

ID: FAIL-REP-00000017
 Version Information: 1
 Name: Dirty air to engine Edit Local...
Description: Dirty air is being passed to engine
 Type: Failure Representation

The difference summary shows the updates as **Property Modified** in the comparison results.

FMEA-Compare-0000037/Difference summary of source and target FMEA
 Source FMEA: FMEA M1 Target FMEA: FMEA Variant Compare Config Changes

Tree Selection Mode Select All

Difference Type	Name	Context	Description	ID
Structure Modified	Assembly	FMEA Variant Compare Config Changes		SE-REP-000000025
Property Modified	Reservoir	FMEA Variant Compare Config Changes		FUN-REP-000000033
Structure Modified	Filter the air	FMEA Variant Compare Config Changes		FUN-REP-000000034
Property Modified	Dirty air to engine	FMEA Variant Compare Config Changes		FAIL-REP-000000017

Configure FMEA guideline objects

About FMEA guideline objects

The FMEA engineers create FMEA objects. While creating them, they can specify the failure analysis mode type as Design FMEA or Process FMEA. These FMEA objects have *guideline objects* associated with them. FMEA provides guideline objects based on the following industry standards:

- The AIAG & VDA industry standards published by the US-based Automotive Industry Action Group (AIAG) in collaboration with the German Association of the Automotive Industry (VDA) in June, 2019.
- The IATF 16949 standards originally created by the International Automotive Task Force that aims at harmonizing the different assessment and certification systems for the automotive sector.
- The AS13004 standards published by SAE International (formerly named Society of Automotive Engineers) in August 2017 that defines the application of process flow diagrams and process failure mode and effects analysis (PFMEA) throughout the lifecycle of a product.

- The AS13100 standards published by SAE International (formerly named Society of Automotive Engineers) in March 2021 that aims at quality management and system requirements for aero engine design and production organizations.


Teamcenter provides a default **FMEAGuideline.xml** file located in the `TC_ROOT\install\qfm0qcfmea\data` folder. The system uses the guideline objects as follows:

- To calculate the priority for a risk analysis. While performing a risk analysis, the FMEA engineer can provide occurrence values and detection values for current prevention control actions and detection control actions, respectively. Based on these values, the system determines the priority, such as low, medium, or high. Different organizations use different methods to prioritize risk. The default values are based on the industry standard guideline that you specify while creating the FMEA.
- To render **Formsheet** view. The *guideline objects* are linked to a **configuration file**. The configuration file is used to render the **Formsheet** view that displays the complete FMEA data, including system element representations, function representations, failure representations, and related actions.

When you install FMEA, the **create_fmea_guidelines** utility runs automatically to create the **FMEAGuideline.xml** configuration file. The guideline objects are created in a release status, and they cannot be modified or deleted. When you upgrade FMEA, if there are no guideline objects with a release status, the utility automatically creates the XML file. However, if the guideline objects exist, the XML file is not created.

If your company does not want to follow the default standards, or if you want to customize the **Formsheet** view, you can edit the **FMEAGuideline.xml** file as required. After changing the values, you must run the **create_fmea_guidelines** utility to update the values in the Teamcenter database. The new values that are then used to calculate the priority for risk analysis and render the **Formsheet** view.

View the guideline values for action priority

1. In Active Workspace, type the search criteria as **guidelines** in the global search, and click .

The search displays the **Design FMEA** and **Process FMEA** guideline objects.

2. Open the **Design FMEA** or the **Process FMEA** object.

The **Overview** tab displays the risk estimation method, creation date, released date, and release status.


3. Click the **Action Priority** tab.

The **ACTION PRIORITY** table displays the severity, occurrence, detection, and action priority values.

Edit the guideline values for action priority

1. Open the `TC_ROOT\install\qfm0qcfmea\data\FMEAGuideline.xml` file.




Attach the configuration file to the guideline object

1. Log on as a user with DBA privileges.
2. Search for the guideline object.
 - a. Type the search criteria as **guidelines** in the global search, and click .

The search displays the **Design FMEA** and **Process FMEA** guideline object.
 - b. Open the **Design FMEA** or the **Process FMEA** guideline object.

The **Overview** tab displays the risk estimation method, creation date, date released, and release status.
3. Add a configuration file.

You must delete the existing configuration file before you can add a new one.

 - a. In the **CONFIGURATION FILE** section, from the primary toolbar, click **Edit**  > **Delete**.
 - b. To add the configuration file, click **Add** , choose the XML file, and click **Add**.
4. To replace the configuration file, from the primary toolbar, click **Edit**  > **Replace**, choose the XML file, and click **Replace**.

Configure the risk matrix

Modify the risk priority value in the risk matrix

You can modify the risk priority value in the risk matrix and override the calculated values if required. The default colors associated with the risk priority values are: High → Red, Medium → Yellow, Low → Green. When you update the risk priority value, the colors of the cells in the risk matrix are also updated.

Procedure

1. Log on as dba and open the FMEA.
2. Click the **Overview** tab, and open the guideline object that is linked to the FMEA.
3. Click the **Risk Matrix** tab.
4. To change the risk priority value, double-click on the cell of the risk matrix that you want to modify.

You can select a cell from the **Severity Vs Occurrence**, **Severity Vs Detection**, or **Detection Vs Occurrence** matrix and change the risk priority value as required.

- In the message box, change the value and click **Save**.

Modifying the configuration file for the Net View Settings panel

Certain FMEA objects are displayed on the New View Setting panel, by default. You can customize this panel to add new objects or remove existing objects. You can make the required changes in the *NetViewConfiguration dataset* file, which contains information about new views and templates. This dataset is added in the NetViewConfiguration dataset preference.

The *NetViewConfiguration dataset* contains NetView Configuration xml file.

1. NetView

It contains information about views and templates.

```
<Netview> <Views> ..... </Views> <Template> ..... </Template> </
Netview>
```

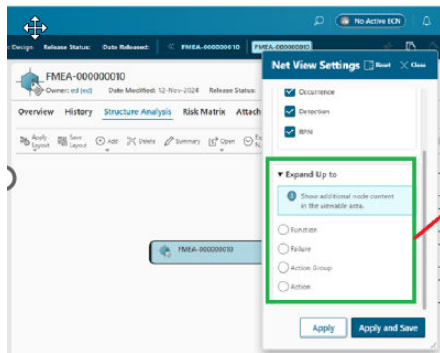
a. Views

It contains system, function, and failure net view configuration.

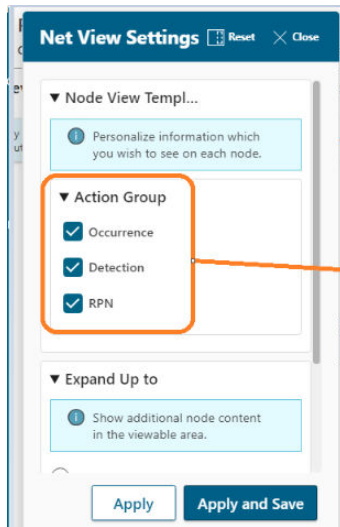
A. SystemNetView

It contains information about the system element view and its child nodes like function, failure, action group, and action.

```
<SystemNetView titleKey="tc_NetView_System"
template="System">
  <FunctionNode titleKey="tc_NetView_Function"
template="FunctionWithoutProp"/>
  <FailureNode titleKey="tc_NetView_Failure"
template="FailureWithoutProp"/>
  <ActionGroupNode titleKey="tc_NetView_ActionGroup"
template="ActionGroup" visiblePropertiesChoice="true"/>
  <ActionNode titleKey="tc_NetView_Action"
template="Action"/>
  ...
</SystemNetView>
```



```
<SystemNetView titleKey="tc_NetView_System" template="System">
  <FunctionNode titleKey="tc_NetView_Function"
  template="FunctionWithoutProp"/>
  <FailureNode titleKey="tc_NetView_Failure" template="FailureWithoutProp"/>
  <ActionGroupNode titleKey="tc_NetView_ActionGroup"
  template="ActionGroup" visiblePropertiesChoice="true"/>
  <ActionNode titleKey="tc_NetView_Action" template="Action"/>
</SystemNetView>
```



```
<SystemNetView titleKey="tc_NetView_System" template="System">
  <FunctionNode titleKey="tc_NetView_Function"
  template="FunctionWithoutProp"/>
  <FailureNode titleKey="tc_NetView_Failure" template="FailureWithoutProp"/>
  <ActionGroupNode titleKey="tc_NetView_ActionGroup"
  template="ActionGroup" visiblePropertiesChoice="true"/>
  <ActionNode titleKey="tc_NetView_Action" template="Action"/>
  ....
</SystemNetView>
```

i. **FunctionNode**

It contains template information, title key, and visible property choice configuration.

ii. **FailureNode**

It contains template information, title key, and visible property choice configuration.

iii. **ActionGroupNode**

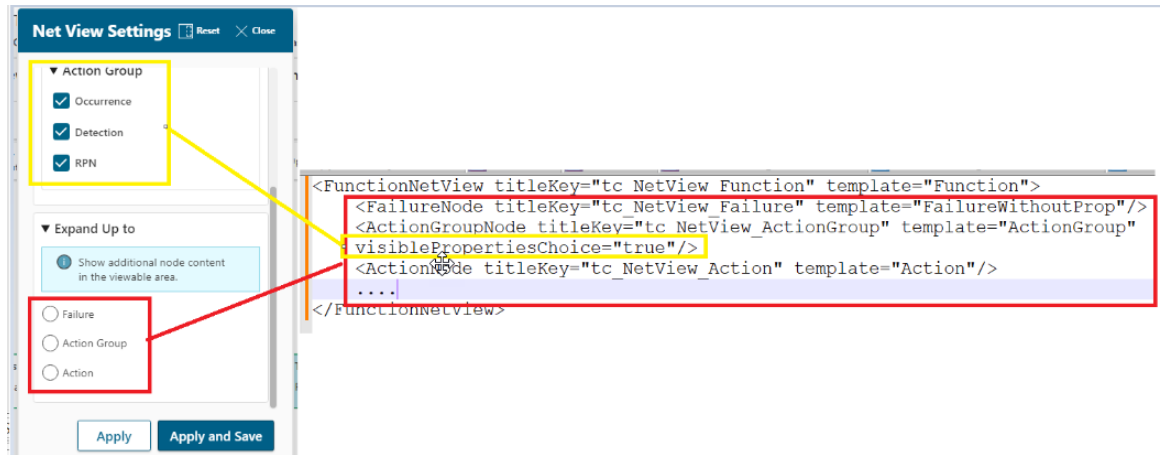
It contains template information, title key, and visible property choice configuration.

iv. **ActionNode**

It contains template information, title key, and visible property choice configuration.

B. FunctionNetView

It contains information about the function element view and its child nodes like failure, action group and action. The **FailureNode**, **ActionGroupNode**, and **ActionNode** come under the **FunctionNetView**.



C. FailureNetView

It contains information about the failure net view and its child nodes like action group and action. The **ActionGroupNode** and **ActionNode** come under the **FailureNetView**.

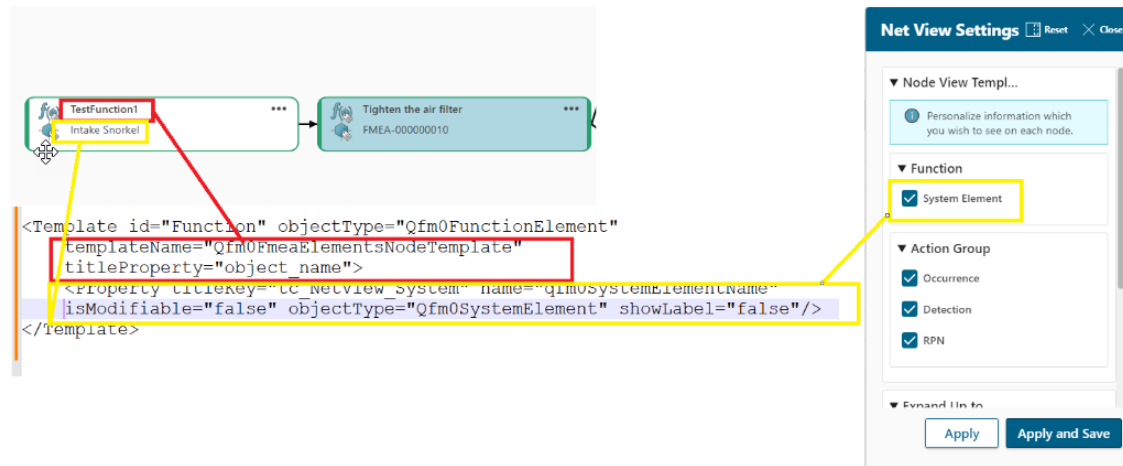


2. Templates

It contains list of template used in system, function, and failure net view.

a. Template

Template can be defined for Teamcenter object types. It contains information about template name from client viewModel.



A. Property

It contains configuration about property name which is shown in the node, show label, and is modifiable.

Customize the FMEA formsheet

Customizing formsheets

The complete FMEA data of the system element representation such as failure effects, failure causes, and all other related actions are displayed in the formsheet. This data can be viewed either in the **Formsheet** tab or in the formsheet report.

You can customize the styling and display of formsheet elements by modifying the following two files which are attached to the guideline objects linked to the FMEA:

- **Configuration file** used for configuring the formsheet elements such as adding or removing rows, columns, or tables.
- **Styling file** used for customizing the styling of formsheet elements such as the font size, color, alignment, orientation of the text, or the background color of cells.

After you specify the customizations in the styling file, you must include the class name of the styling file in the configuration file, and make sure that both the styling file and the configuration file are attached to the guideline object. Any custom styling which is defined in the styling file and configuration file is applied to both, the **Formsheet** tab view and the report.

FMEA formsheet elements

About formsheet elements

You can customize the layout of the **Formsheet** view by editing the configuration file that is attached to the FMEA guideline objects. These configuration files are XML rendering templates (XRT), also called style sheets. These XML files are stored in the Teamcenter database and are read as needed. Therefore, changes made to these rendering templates are reflected in the graphical user interface (GUI) without the need to build or deploy an application file.

Hydrogen Station Equipment Performance Device					
Exit Full Screen					
▼ FORMSHEET HEADER					
DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 1)					
Company Name	Siemens	Subject	Hydrogen Station Equipment Performance Device	DFMEA ID Number	FMEA-00000001
Engineering Location		DFMEA Start Date	07-May-2020 18:17	Design Responsibility	ed (ed)
Customer Name	Siemens	DFMEA Revision Date		Confidentiality Level	
Model Year/Platform	2019 /	Cross Functional Team			
STRUCTURE ANALYSIS (STEP 2)	FUNCTIONAL ANALYSIS (STEP 3)	FAILURE ANALYSIS (STEP 4)			
Focus Element	Focus Element Function and Requirement	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic
	Contain Hydrogen	Explosive atmosphere inside trailer	9	Loss of containment (minor leakage)	Component Leak

The XRT templates consist of the following elements:

Tip:

You must validate the positioning and usage of these elements manually. There is no schema for the formsheet configuration file.

Top elements

This element must be the overall element for the XML file.

<rendering>

This is the overall wrapper element for the panel's XML file.

Main elements

The main rendering of the **Formsheet** view typically consists of multiple headers. Typically, `<formsheetHeader>` contains row elements, `<objectSet>` contains **columnGroups** elements, and style contains **columnStyles** elements.

<code><formsheetHeader></code>	Displayed at the top of the rendered Formsheet view. This is used to organize information in the Formsheet header section.
<code><formsheetTableStyle></code>	Defines the default styling of the table column header and its contents.
<code><objectSet></code>	Displays the Formsheet table of properties from the related objects.
<code><style></code>	Adds the conditional style class to the table columns defined in <code><objectSet></code> .
<code><fsTableCommands></code>	(Optional) Defines the UI command ids to be returned along with column info. This information can be used in command condition for command visibility or disabling the command.

formsheetHeader elements

Groups and organizes your header or the property elements for the **Formsheet** header.

<code><row></code>	Groups multiple headers or property elements.
--------------------------	---

formsheetTableStyle elements

Defines the default styling class for the column group, the header, and the data.

<code><columnGroupStyle></code>	Defines the default style class for <code><columnGroup></code> .
<code><columnHeaderStyle></code>	Defines the default style class for the column header.
<code><columnDataStyle></code>	Defines the default style class for the column data.
<code><columnGroupDataStyle></code>	(Optional) Defines the default style class for the column group data.

objectSet elements

Groups and organizes your column elements.

<code><columnGroup></code>	Groups multiple column elements.
<code><column></code>	Creates a column property or DCP supported by Formsheet .

Style elements

Applies style classes conditionally.

<code><columnStyle></code>	Defines styling for the column data. This overrides the default styling for the specified column.
----------------------------------	---

Row elements

Displays header and property information.

<code><header></code>	Displays the header cell in the Formsheet header.
<code><property></code>	Displays a single property by name. This is the database name of the property and not the localized display name. You can choose a property or a dynamic compound property (DCP) supported by Formsheet on the FMEA node.

columnGroup elements

Creates a column in the **Formsheet** table for your property or DCP supported by **Formsheet**.

<code><column></code>	Creates a column property or DCP supported by Formsheet .
<code><dataCondition></code>	(Optional) Defines the data condition for the column group data.

columnStyle elements

Defines column specific styling.

<code><columnHeaderStyle></code>	Defines the column-specific styling class and the width of the column.
<code><template></code>	Defines the template styling or condition-based styling for the column data.
<code><dataCondition></code>	(Optional) Defines the data condition for the column group data.

template elements

Defines conditions for the template.

<code><if></code>	Defines the If condition for the style template.
<code><elseif></code>	Defines the Else if condition for the style template.
<code><else></code>	Defines the Else condition for the style template.

if else elements

Defines common elements for if else elements.

<code><conditions></code>	Groups multiple conditions. This element is not part of <code><else></code> .
<code><class></code>	Styling class for the defined condition.

conditions elements

Defines a condition with the operator used within a multiple expression.

<code><expression></code>	Defines the expression for the styling template.
---------------------------------	--

fsTableCommands elements

Defines formsheet table commands for column or column group.

<code><columnCommands></code>	Defines the column or column group command list.
-------------------------------------	--

columnCommands elements

Defines formsheet table commands for specific column or column group.

<code><command></code>	Defines the column or column group command.
------------------------------	---

dataCondition elements

Defines data condition for column or column group.

<code><value></code>	Defines the value for the condition.
----------------------------	--------------------------------------

command

Define command id for column or column group.

value elements

Defines the value for the condition.

rendering

This is the root element of a formsheet configuration file. All the formsheet elements are within this root element.

Attributes

None.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
  <rendering>
    < formsheetHeader logo="" >
      ...
    < /formsheetHeader>
    <objectSet source=".....">
      ...
    < /objectSet>
    <style>
      ...
    </style>
  </rendering>
```

additionalData

Displays additional information in columns in separate cells in the formsheet table and is optional.

Usage

Displays additional information in columns in the formsheet table in the form of a list or as plain text.

The `additionalData` element must be used as a child element of the `column` element.

Attributes

- **name:** (Mandatory) Specifies the database name of a property on the object. The source object of parent `column` element will be referred to as the source object for the `additionalData` element. You can add DCP properties in the name attribute. Examples: `name="GRM(Qfm0FunctionRequirement,WorkspaceObject).object_name"`
- **id:** (Mandatory) Specifies the unique ID for the additional data. The uniqueness of this ID is specific to `additionalData`, `columnGroup` and `column` elements.
- **titleKey:** (Optional) Specifies the key used for localization. It is used as the title for the additional data.
- **list:** (Optional) Use this attribute to display additional data in the form of a list, but skip this attribute if you want to display the data as plain text. Depending on the type of list you want, you can provide the value either as *ordered* or *unordered*.
- **marker:** (Optional) This attribute must be used only if the **list** attribute is present. It specifies the marker type of list. For example, if the list is an unordered list, `marker=disc` will have a bullet point before each list item.
- **sortByProperty:** (Optional) Sorts the data of **additionalData** using the defined property. The sorting is done in context of the source object.

Example:

```
<additionalData id="additional_FS_Requirement"
titleKey="tc_FS_Requirement"
name="GRM(Qfm0FunctionRequirement,WorkspaceObject).object_name"
list="unordered" marker="disc"
sortByProperty="relation_object.aqc0Order" />
```

All `Qam0QualityAction` will be sorted based on `qam0Sequence`. Data can also be sorted by using **relation_object** if the source column is rendered using `relation sortByProperty="relation_object.aqc0Order"`.

- **sortDirection:** (Optional) Specified the sorting direction of data as ascending or descending. The default value is ascending. The sort direction is considered only if the **sortByProperty** attribute is defined.
- **viewType:** (Optional) Lets you view the type for additional data. The supported types are **Text** and **Image**. The supported values are ascending or descending. The default value is **Text**. The **Image** value is used for rendering the attached image to dataset.

Example:

```
<additionalData id="additional_Sys_Ele_Images" titleKey="tc_FS_Images"
name="GRM(Aqc0Pictures,Image).awp0ThumbnailImageTicket "
list="unordered" marker="disc" viewType="Image" />
```

- **isModifiable:** (Optional) `additionalData` of column can be configured to allow or disallow modification. The default value is true. If user has the access to modify the property, and the value of `isModifiable` is set to false, then user cannot modify the property.

Example:

```
<additionalData id="additional_Sys_Ele_Images" titleKey="tc_FS_Images"
name="GRM(Aqc0Pictures,Image).awp0ThumbnailImageTicket "
list="unordered" marker="disc" viewType="Image" />
```

Examples

additionalData as a list

Suppose there is a column in the formsheet that displays the function name. Along with the function name, you want to show all the requirements and characteristics linked to this function object in the same cell itself in the form of an unordered list with bullets. To do this, use the `additionalData` element as follows:

```
<column id="col_Focus_Function_Ele"
titleKey="tc_Focus_Element_Function_Requirement"
sourceColumn="col_Focus_Sys_Ele.object"
name="REF(qfm0FunctionsElementsList,Qfm0FunctionElement).object_name">
    <additionalData id="additional_FS_Requirement"
titleKey="tc_FS_Requirement"
name="GRM(Qfm0FunctionRequirement,WorkspaceObject).object_name"
list="unordered" marker="disc" />
    <additionalData id="additional_FS_Characteristic"
titleKey="tc_FS_Characteristic"
name="GRM(Qfm0InspectionDefinition,Aqc0CharElementRevision).object_name"
list="unordered" marker="disc" />
</column>
```

FUNCTIONAL ANALYSIS (STEP 3) <<		
1. Next Higher Level Function <<	2. Focus Element Function and Requirement or Characteristic <<	3. Next Lower Level Function <<
	Supply Clean and compress the ambient air	Act as a reservoir of clean and compressed air
		Provide ambient air to air filter
		Compress the clean air
		Filter the air to provide clean air
		Requirement <ul style="list-style-type: none"> Filter the air to provide clean air
		Passage of clean air without leakage

To display the additionalData as an ordered list having a marker as decimal numbers such as 1,2,3, use the following:

```
<column id="col_Focus_Function_Ele"
titleKey="tc_Focus_Element_Function_Requirement"
sourceColumn="col_Focus_Sys_Ele.object"
name="REF(Qfm0FunctionsElementsList,Qfm0FunctionElement).object_name">
  <additionalData id="additional_FS_Requirement"
titleKey="tc_FS_Requirement"
name="GRM(Qfm0FunctionRequirement,WorkspaceObject).object_name"
list="ordered" marker="decimal" />
  <additionalData id="additional_FS_Characteristic"
titleKey="tc_FS_Characteristic"
name="GRM(Qfm0InspectionDefinition,Aqc0CharElementRevision).object_name"
list="ordered" marker="decimal" />
</column>
```

FUNCTIONAL ANALYSIS (STEP 3) <<		
1. Next Higher Level Function <<	2. Focus Element Function and Requirement or Characteristic <<	3. Next Lower Level Function <<
	Supply Clean and compress the ambient air	Act as a reservoir of clean and compressed air
		Provide ambient air to air filter
		Compress the clean air
		Filter the air to provide clean air
	Requirement <ul style="list-style-type: none"> Filter the air to provide clean air Characteristic <ul style="list-style-type: none"> Inspect-Thickness Inspect-Diameter 	Passage of clean air without leakage

additionalData as plain text

To display additional data as plain text, the `list` and `marker` attributes are not required.

In this example, the type of recommended action is shown as additional data where **Type** is the title and **Prevention Action** is the data.

```
<column id="col_Recommended_Action"
titleKey="tc_FS_Recommended_Actions"
sourceColumn="col_Failure_Cause.relation_object"
name="REF(qfm0PreventionOptActions,Qam0QualityAction).object_name,REF(
qfm0DetectionOptActions,Qam0QualityAction).object_name">
  <additionalData id="additional_FS_Action_Type"
titleKey="tc_FS_Action_Type" name="qam0QualityActionSubtype"/>
</column>
```

Recommended Actions <<
Revision State
CFD Analysis
Type <ul style="list-style-type: none"> Prevention Action
Extreme boundary condition testing
Type <ul style="list-style-type: none"> Detection Action

In the following example, the due date and time information is shown as additional data.

```
<column id="col_Responsibility_Target_Date"
titleKey="tc_Responsibility_Target_Date"
sourceColumn="col_Recommended_Action.object"
name="fnd0ResponsibleUser"> `
  <additionalData id="col_ Target_Date" name="gam0DueDate"/>
</column>
```

Recommended Actions <<	Responsibility and Target Completion Date <<
Extreme boundary condition testing	
Type <ul style="list-style-type: none"> Detection Action 	
Revision State	
Detection Action	19-May-2023 00:00
Type <ul style="list-style-type: none"> Detection Action 	<ul style="list-style-type: none"> ed (ed)

formsheetHeader

Specifies the content for the formsheet header.

Usage

This is typically at the top level of the formsheet, immediately after the rendering element.

Attributes

- **logo**: User can define dataset name where .svg file of logo is attached. If not defined the default dataset **QualityReportCompanyLogo** will be used for logo.

Example

```
<rendering>
<formsheetHeader logo=" " >
  <row>
    <header titleKey=.../>
    ...
  </row>
  <row>
    <property titleKey=.../>
    <header titleKey=.../>
    ...
  </row>
</formsheetHeader>
```

▼ FORMSHEET HEADER

DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)						
PLANNING AND PREPARATION (STEP 1)						
SIEMENS	Company Name	BMW	Subject	Air Intake System Design FMEA	DFMEA ID Number	FMEA-00000008 / 1
	Engineering Location	Pune	DFMEA Start Date	5/17/2023 5:51:41 AM	Design Responsibility	Ratnakar, Saurabh
	Customer Name	Sogefi	DFMEA Revision Date	5/17/2023 5:51:41 AM	Confidentiality Level	Business Use
	Model Year/Platform	2021-22 / Passenger Cars	Cross Functional Team			

row

Defines the formsheet row. It is a container for header and property cells.

Usage

Groups a header and property in a table row.

Attributes

None


Example

```
<rendering>
<formsheetHeader logo=" " >
  <row>
    <header titleKey=.../>
    ...
  </row>
```

```

<row>
  <property titleKey=.../>
  <header titleKey=.../>
  ...
</row>
</formsheetHeader>
</rendering>

```

▼ FORMSHEET HEADER						
	DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
	PLANNING AND PREPARATION (STEP 1)					
	Company Name	BMW	Subject	Air Intake System Design FMEA	DFMEA ID Number	FMEA-00000008 / 1
	Engineering Location	Pune	DFMEA Start Date	5/17/2023 5:51:41 AM	Design Responsibility	Ratnakar, Saurabh
	Customer Name	Sogefi	DFMEA Revision Date	5/17/2023 5:51:41 AM	Confidentiality Level	Business Use
Model Year/Platform	2021-22 / Passenger Cars	Cross Functional Team				

Additional information

The last child elements, that is, header and property, are expanded till the end of the row. Examples:

DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)

Cross Functional Team

header

Defines the header text in the formsheet header table.

Usage

Add any text to the header table.

Attributes

titleKey: (Optional) Specifies the key used for localization.

Example

```

<rendering>
<formsheetHeader logo=" " >
  <row>
    <header titleKey="tc_DFMEA_Fmea_Header"/>
  </row>
  <row>
    <header titleKey=.../>
  ...
</row>
<row>

```

```

        <property titleKey=.../>
        <header titleKey=.../>
    ...
    </row>
</formsheetHeader>
</rendering>

```

▼ FORMSHEET HEADER						
SIEMENS	DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
	PLANNING AND PREPARATION (STEP 1)					
	Company Name	BMW	Subject	Air Intake System Design FMEA	DFMEA ID Number	FMEA-00000008 / 1
	Engineering Location	Pune	DFMEA Start Date	5/17/2023 5:51:41 AM	Design Responsibility	Ratnakar, Saurabh
	Customer Name	Sogefi	DFMEA Revision Date	5/17/2023 5:51:41 AM	Confidentiality Level	Business Use
Model Year/Platform	2021-22 / Passenger Cars	Cross Functional Team				

Additional information

The header in the formsheet header uses 2 cells or two columns to display the information.

property

Displays property names and values from the FMEA object or the dynamic compound property.

Attributes

- **titleKey:** (Optional) Specifies the key used for localization.
- **name:** Specifies the database name of a property on the object. This is not the display name. It is not a required attribute. You can leave this empty if you specify the **titleKey** as a blank value.

You can add DCP properties in the **name** attribute, for example,
name="GRM(HasParticipant,GroupMember).user"

- **srcObject:** (Optional) Specifies the object from which the property needs to be rendered. Default value is fmea. fmea – Uses FMEA Node object for rendering property selected – Uses selected object for which formsheet is rendered. FMEA Node or system element.
- **isModifiable:** (Optional) Data of column group can be configured to allow or do not allow modification. Default value is true if not defined. If user has access to modify the property and in configuration is modifiable is false then user cannot modify the property.

Example

```

<rendering>
  <formsheetHeader logo="">
    <row>
      <header titleKey="tc_DFMEA_Fmea_Header"/>
    </row>

```

```

<row>
  <header titleKey="tc_Planning_AND_Preparation" />
</row>
<row>
  <property titleKey="tc_Company_Name"
    name="qfm0Companyname" />
  <property titleKey="tc_Subject" name="object_name" />
  <property titleKey="tc_DFMEA_ID_Number" name="qfm0Id" />
</row>
<row>
  <property titleKey="tc_Engineering_Location" name="" />
  <property titleKey="tc_DFMEA_Start_Date"
    name="creation_date" />
  <property titleKey="tc_DFMEA_Responsibility"
    name="owning_user" />
</row>
<row>
  <property titleKey="tc_Customer_Name"
    name="qfm0Customer" />
  <property titleKey="tc_DFMEA_Revision_Date" name="" />
  <property titleKey="tc_Confidentiality_level" name="" />
</row>
<row>
  <property titleKey="tc_ModelYear_Platform"
    name="CONCAT(qfm0ModelYear,/,qfm0Platform)" />
  <property titleKey="tc_Cross_Functional_Team"
    name="GRM(HasParticipant,GroupMember).user" />
</row>
</formsheetHeader>

```

Overview [Formsheet](#) Full Screen

FORMSHEET HEADER					
DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 5)					
Company Name	Siemens	Subject	Hydrogen Station Equipment Performance Device	DFMEA ID Number	FMEA-000000001
Engineering Location		DFMEA Start Date	13-Mar-2020 15:30	Design Responsibility	infodba (infodba)
Customer Name	Siemens	DFMEA Revision Date		Confidentiality Level	
Model Year/Platform	2019 /	Cross Functional Team			

formsheetTableStyle

Defines the default styling for the formsheet column group, header, and data styling.

Attributes

None.

Example

```
<rendering>
  ...
  <formsheetTableStyle>
    <columnGroupStyle cssClass=" " />
    <columnHeaderStyle
      cssClass="aw-qualityFmea-formsheetColumnHeaderText "
      width="200px" />
    <columnDataStyle cssClass=" " />
  </formsheetTableStyle>
</rendering>
```

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure (Lower Et)
	FMEA-00000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

columnGroupStyle

Defines the default styling for formsheet column groups.

Attributes

cssClass: (Mandatory) Specifies the **css** class for the column group.

Example

```
<rendering>
  ...
  <formsheetTableStyle>
    <columnGroupStyle cssClass=" " />
    <columnHeaderStyle
      cssClass="aw-qualityFmea-formsheetColumnHeaderText "
      width="200px" />
    <columnDataStyle cssClass=" " />
  </formsheetTableStyle>
</rendering>
```

Formsheet

Refresh Filter Print Formsheet Add Delete 25 Rows per Page Full Screen

► FORMSHEET HEADER

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure (Lower El)
	FMEA-00000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

columnHeaderStyle

Defines the default styling for formsheet column headers.

Attributes

- **cssClass:** (Mandatory) Specifies the **css** class for the column group.
- **width:** (Mandatory) Specifies the default width of the column.

Example

```
<rendering>
  ...
  <formsheetTableStyle>
    <columnGroupStyle cssClass=" " />
    <columnHeaderStyle
      cssClass="aw-qualityFmea-formsheetColumnHeaderText "
      width="200px" />
    <columnDataStyle cssClass=" " />
  </formsheetTableStyle>
</rendering>
```

Formsheet

Refresh Filter Print Formsheet Add Delete 25 Rows per Page Full Screen

► FORMSHEET HEADER

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure (Lower El)
	FMEA-00000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

command

Define command id for column or column group.

Attributes

- **commandId:** (Mandatory) Specifies the command id.

Example

```
<fsTableCommands>
  <columnCommands id="col_Focus_Sys_Ele">
    <command commandId="qfm0AddChildInFormsheet" />
  </columnCommands>
</fsTableCommands>
```

fsTableCommands

Define command id for column or column group for formsheet table.

Attributes

None.

Example

```
<fsTableCommands>
  <columnCommands id="col_Focus_Sys_Ele">
    <command commandId="qfm0AddChildInFormsheet" />
  </columnCommands>
</fsTableCommands>
```

columnDataStyle

Defines the default styling for the formsheet column data.

Attributes

cssClass: (Mandatory) Specifies the **css** class for the column group.

Example

```
<rendering>
  ...
  <formsheetTableStyle>
    <columnGroupStyle cssClass="" />
  </formsheetTableStyle>
</rendering>
```

```

<columnHeaderStyle
  cssClass="aw-qualityFmea-formsheetColumnHeaderText "
  width="200px" />
<columnDataStyle cssClass=" " />
</formsheetTableStyle>
</rendering>

```

The screenshot shows a web-based FMEA formsheet interface. At the top, there are navigation icons (Refresh, Filter, Print, Add, Delete) and a 'Rows per Page' dropdown set to 25. Below the navigation is a table with the following structure:

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure C Lower Et
	FMEA-000000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

objectSet

Defines the formsheet table format by using properties. These properties can be from the selected object or from the related objects.

Attributes

source: (Mandatory) Specifies the data provider that returns the set of objects. The format for the attribute value is **dataProvider.type**, where the **dataProvider** is the name of a data provider, that is, the child class of **fndOBaseProvider**. The system recognizes the subtypes of the specified object type.

Currently, only a single data provider is supported as the source. The default **Qfm0FormSheetSourceProvider** returns the **System Element** list in the predefined format.

Example

```

<rendering>
  ...
  <objectSet
source="Qfm0FormSheetSourceProvider.Qfm0SystemElement">
    <columnGroup id=...>
      <column id=.../>
    </columnGroup>
  ...
</objectSet >
</rendering>

```

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure (Lower EL)
	FMEA-00000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

columnGroup

columnGroup

Groups the column elements.

Attributes

- **id:** (Mandatory) Specifies the unique ID for the column group. The uniqueness of this ID is specific to **columnGroup** and column elements.
- **titleKey:** (Mandatory) Specifies the key used for localization.
- **sourceColumn:** (Optional) Specifies the column to refer as the source object. The format of the value is `columnKey.objectType`, where `columnKey` is the value from the ID attribute of the source column and `objectType` means `object` or `relation_object` from the source column.

Example:

```
sourceColumn=" col_Failure_Cause.object "
```

OPTIMIZATION (STEP 6)										
DFMEA Prevention Action	DFMEA Detection Action	Responsible Person's Name	Target Completion Date	Status	Action Taken with Pointer to Evidence	Completion Date	Severity (S)	Occurrence (O)	Detection (D)	DFMEA AP
					Revision State QA0000000001 Type : Prevention Action Due Date : 04-Nov-2022 00:00 Completed by : ed (ed) Comment :	04-Nov-2022 13:39	10	10	10	High

You can skip `objectType` if you want to use the object from the source column. `sourceColumn=" col_Failure_Cause "` and `sourceColumn=" col_Failure_Cause.object "` are the same. To use the relation object from the source column as the source: `sourceColumn=" col_Failure_Cause.relation_object "`.

- **name** (Optional)

Specifies the database name of a property on the object. This is not the display name and is a required attribute. You can add DCP properties in the name attribute. Examples:
 name="object_name" name="REF(qfm0FunctionsElementsList,Qfm0FunctionElement).object_name"
 name="GRM(Qfm0Effect,WorkspaceObject).object_name" If the titleKey attribute is not provided, the localized value of the property is used as the column name. If the validation of the traversal fails, the column name is displayed as "sourceColumn + name" as the column name.

- **hidden:** (Optional) Column Group along with it's column will not be shown in formsheet.
- **showHeader:** (Optional) If false, In Formsheets columns under this column group will be shown without column group Header. Default value is "true" if not defined.
- **sortByProperty:** (Optional) Sort the Data of column group using defined property. The sorting will happen in context of source object. Example : <columnGroup id="col_Opt_Action_Grp" titleKey="tc_Optimization" sourceColumn="col_Failure_Cause.object" name="GRM(Qfm0OptimizationAction,Qam0QualityAction).object_name" sortByProperty="qam0Sequence"> All Qam0QualityAction will be sorted based on qam0Sequence. Data can be sorted by using relation_object also, If source column is rendered using relations. sortByProperty="relation_object.aqc0Order"
- **sortDirection:** (Optional) sorting direction of Data. Ascending or Descending. Default value is Ascending if not defined. Sort Direction will be considered only if sort By property is defined.
- **isModifiable:** (Optional) Data of column group can be configured to allow or do not allow modification. Default value is true if not defined. If user has access to modify the property and in configuration is modifiable is false then user cannot modify the property.

Example

```
<rendering>
  ...
  <objectSet
source="Qfm0FormSheetSourceProvider.Qfm0SystemElement">
  <columnGroup id="colGrp1" titleKey="tc_Structure_Analysis">
    <column id="col1" .../>
  </columnGroup>
  <columnGroup id="colGrp2" titleKey="tc_Failure_Analysis">
    <column id="col2" .../>
    <column id="col3" .../>
    <column id="col4" .../>
    <column id="col5" .../>
  </columnGroup>
  ...
</objectSet >
< /rendering>
```

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure (Lower Level)
	FMEA-00000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

dataCondition

Defines the data conditions for the columns or column groups.

Attributes

- **property:** (Mandatory) Specifies the **property** of object from the column or column group.
- **operation:** (Mandatory) Specifies the operation for the data condition. Supported types are:

range

contains

notContains

startsWith

endsWith

equals

notEquals

caseSensitiveEquals

caseSensitiveNotEquals

lt

gt

lte

gte

Example

```

<rendering>
    <column
id="col_Opt_Detection_Action" titleKey="tc_DFMEA_Opt_Detection_Action"
sourceColumn="col_Opt_Action_Grp.object"
name="REF(qam0DependentQualityActions,Qam0QualityAction).object_name"
sortByProperty="qam0Sequence">
        <dataCondition
property="qam0QualityActionSubtype" operation="equals">
            <value>Detection Action</value>
        </dataCondition>
        <dataCondition
property="qam0QualityActionStatus" operation="notEquals">
            <value>Completed</value>
        </dataCondition>
    </column>
</rendering>

```

The screenshot shows a 'Formsheet' window with a table titled 'OPTIMIZATION (STEP 6)'. The table has columns for DFMEA Prevention Action, DFMEA Detection Action, Responsible Person's Name, Target Completion Date, Status, Action Taken with Pointer to Evidence, Completion Date, Severity (S), Occurrence (O), Detection (D), and DFMEA AP. The table contains two rows of data. The first row shows a 'Prevention Action' with a status of 'Draft', a completion date of '04-Nov-2022 13:39', and a severity of '10'. The second row shows a 'Detection Action' with a status of 'Draft' and a severity of '10'. The 'DFMEA AP' column for both rows is 'High'.

DFMEA Prevention Action	DFMEA Detection Action	Responsible Person's Name	Target Completion Date	Status	Action Taken with Pointer to Evidence	Completion Date	Severity (S)	Occurrence (O)	Detection (D)	DFMEA AP
Prevention Action				Draft	QA0000000001 Type : Prevention Action Due Date : 04-Nov-2022 00:00	04-Nov-2022 13:39	10	10	10	High
	Detection Action			Draft	Completed by : ed (ed) Comment :					

columnGroupsDataStyle

Define the data for the column groups.

Attributes

- **cssClass:** (Mandatory) Specifies the **css** class for the column group

Example

```

<rendering>
    ...
    <formsheetTableStyle>
        <columnGroupStyle cssClass="" />
        <columnHeaderStyle
            cssClass="aw-qualityFmea-formsheetColumnHeaderText"
            width="200px" />
        <columnDataStyle cssClass="" />
    </formsheetTableStyle>

```

```

        <columnGroupDataStyle cssClass="aw-qualityFmea-
formsheetTextAlignCenter" />
    </formsheetTableStyle>
</rendering>

```

RISK ANALYSIS (STEP 5) <<				
Current Prevention Control (PC) of FC <<	Occurrence (O) of FC <<	Current Detection Controls (DC) of FC or FM <<	Detection (D) of FC/FM <<	DFMEA AP <<
Initial State				
Clamping Specification as per IS standard	4	Smoke Testing for Air Intake System	7	High

columnCommands

Define command id for column or column group for formsheet table.

Attributes

- **id:** (Mandatory) Specifies the column or column group id.

Example

```

<fsTableCommands>
  <columnCommands id="col_Focus_Sys_Ele">
    <command commandId="qfm0AddChildInFormsheet" />
  </columnCommands>
</fsTableCommands>

```

column

Displays columns in the formsheet table.

Attributes

- **id:** (Mandatory) Specifies the unique ID for the column. The uniqueness of this ID is specific to **columnGroup** and column elements.
- **titleKey:** (Optional) Specifies the key used for localization.

- **sourceColumn:** (Mandatory)

Specifies the column to refer as the source object. If the value is put as blank (null), the source is referred as the **objectSet** element's source data. The format of the value is **columnKey.objectType**, where **columnKey** is the value from the ID attribute of the source column and **objectType** means **object** or **relation_object** from the source column.

Examples:

- Value is null: **sourceColumn=""**
- Alternate source column: Define an alternate source by using **OR(columnKey,columnKey)**. If the first source column value is null, then the system takes the value from the second source column.

```
sourceColumn="OR(col12,col13)"
```

```
sourceColumn="OR(col12,col13).object"
```

DFMEA Prevention Action	DFMEA Detection Action	Responsible Person's Name	Target Completion Date	Status
Filter installation procedure, detection will be improved if included, if filter is installed once, and checked at annual manite		infodba (infodba)	13-Mar-2020 15:30	Completed
	Filter installation procedure, detection will be improved if included, if filter is installed once, and checked at annual manite	infodba (infodba)	13-Mar-2020 15:30	Completed

You can skip **objectType** if you want to use the object from the source column.

sourceColumn="col1" and **sourceColumn="col1.object"** are the same.

To use the relation object from the source column as the source:

sourceColumn="col6.relation_object"

- **name:** Specifies the database name of a property on the object. This is not the display name and is a required attribute. You can leave this empty if you specify the **titleKey** as a blank value.

You can add DCP properties in the **name** attribute. Examples:

```
name="object_name"
```

```
name="REF(qfm0FunctionsElementsList,Qfm0FunctionElement).object_name"
```

```
name="GRM(Qfm0Effect,WorkspaceObject).object_name"
```

If the **titleKey** attribute is not provided, the localized value of the property is used as the column name.

If the validation of the traversal fails, the column name is displayed as **"sourceColumn + name"** as the column name.

- **alternateDisplayCol:** (Optional) You can specify the next column ID to display as an alternative to the current column. Examples:

```
<column id="col12" ... alternateDisplayCol="col13" />
```

```
<column id="col13" ... />
```

The following displays the alternate display columns with the style applied to gray out or disable cells that are not used.

DFMEA Prevention Action	DFMEA Detection Action
Filter installation procedure, detection will be improved if included. If filter is installed once, and checked at annual manite	
	Filter installation procedure, detection will be improved if included. If filter is installed once, and checked at annual manite

Notes:

- Do not use the previous column as the alternate source column.
- If the second column does not have values to display, then the first column's next value is shown in that row.
- Recommended if both the column's **sourceColumn** is the same.
- **isModifiable:** (Optional) Data of column group can be configured to allow or do not allow modification. Default value is true if not defined. If user has access to modify the property and in configuration is modifiable is false then user cannot modify the property.
- **hidden:** (Optional) Column will not be shown in formsheet.

- **sortByProperty:** (Optional) Sort the Data of column using defined property. The sorting will happen in context of source object. Example : `<column id="col_Failure_Mode" titleKey="tc_Failure_Mode" sourceColumn="col_Focus_Function_Ele.object" name="REFBY(qfm0ParentElement,Qfm0FailureElement).object_name" sortByProperty="qfm0Sequence"/>` All Qam0QualityAction will be sorted based on qam0Sequence. Data can be sorted by using relation_object also, If source column is rendered using relations. `sortByProperty="relation_object.aqc0Order"`
- **sortDirection:** (Optional) sorting direction of Data. Ascending or Descending. Default value is Ascending if not defined. Sort Direction will be considered only if sort By property is defined.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
  ...
  <objectSet source="Qfm0FormSheetSourceProvider.Qfm0SystemElement">
    <columnGroup id="colGrp1" titleKey="tc_Structure_Analysis">
<column id="col1" titleKey="tc_DFMEA_Focus_Element" sourceColumn=" "
  name="object_name"/>
    </columnGroup>
    <columnGroup id="colGrp2" titleKey="tc_Functional_Analysis">
<column id="col2" titleKey="tc_Focus_Element_Function_Requirement"
  sourceColumn="col1.object"
name="REF(qfm0FunctionsElementsList,Qfm0FunctionElement).object_name" />
    </columnGroup>

    <columnGroup id="colGrp3" titleKey="tc_Failure_Analysis">
    <column id="col3" titleKey="tc_Failure_Effect" sourceColumn="col5"
name="GRM(Qfm0Effect,WorkspaceObject).object_name" />
      <column id="col4" titleKey="tc_Severity_of_Effect"
        sourceColumn="col3.relation_object" name="qfm0Severity"/>
      <column id="col5" titleKey="tc_Failure_Mode"
sourceColumn="col2"
name="REF(qfm0FailureElementsList,Qfm0FailureElement).object_name"/>
    <column id="col6" titleKey="tc_Failure_Cause" sourceColumn="col5"
      name="GRM(Qfm0Cause,WorkspaceObject).object_name"/>
    </columnGroup>
    ...
  </objectSet>
</rendering>
```

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)			FAILURE ANALYSIS (STEP 4)			
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure (Lower Level)
	FMEA-00000010			Act as a reservoir of clean air Requirement : New_Requirement_Edit Requirement : Air Filter Clogging	Wheel_Function	Water being sucked by intake snorkel	10	Air Filter Clogging	Water in A
								No Air to Engine	

Example for alternate display and source columns

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
    ...
    <objectSet
source="Qfm0FormSheetSourceProvider.Qfm0SystemElement">
        ...
<columnGroup id="colGrp5" titleKey="tc_Optimization">
    <column id="col12" titleKey="tc_DFMEA_Opt_Prev_Action"
        sourceColumn="col6.relation_object"

name="REF(qfm0PreventionOptActions,Qam0QualityAction).
        object_name" alternateDisplayCol="col13"/>
    <column id="col13"
titleKey="tc_DFMEA_Opt_Detection_Action"
        sourceColumn="col6.relation_object"

name="REF(qfm0DetectionOptActions,Qam0QualityAction).
        object_name"/>
    <column id="col14" titleKey="tc_Resp_person"
        sourceColumn="OR(col12,col13)"
        name="fnd0ResponsibleUser" />
    <column id="col15"
titleKey="tc_Target_Completion_Date"
        sourceColumn="OR(col12,col13)"

name="qam0DueDate"/>
    <column id="col16" titleKey="tc_Status"
        sourceColumn="OR(col12,col13)"
        name="qam0QualityActionStatus" />
    <column id="col17"
titleKey="tc_Action_Taken_with_Evidence"
        sourceColumn="OR(col12,col13)"

name="qam0Comment"/>
    <column id="col18" titleKey="tc_Completion_Date"
        sourceColumn="OR(col12,col13)"
        name="qam0CompletionDate" />

```

```

<column id="col19" titleKey="tc_Severity"
        sourceColumn="col5" name="qfm0MaxSeverity" />
<column id="col20" titleKey="tc_Occurrence"
        sourceColumn="col12" name="qam0Occurrence" />
<column id="col21" titleKey="tc_Detection"
        sourceColumn="col13" name="qam0Detection" />
<column id="col22" titleKey="tc_DFMEA_AP"
        sourceColumn="col6.relation_object"
        name="qfm0FinalActionPriority"/>
</columnGroup>
    . . .
</objectSet>
</rendering>

```

OPTIMIZATION (STEP 3)										
DFMEA Prevention Action	DFMEA Detection Action	Responsible Person's Name	Target Completion Date	Status	Action Taken with Pointer to Evidence	Completion Date	Severity (S)	Occurrence (O)	Detection (D)	DFMEA AP
Filter installation procedure, detection will be improved if included, if filter is installed once, and checked at annual maintenance		Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30	5	3		Low
	Filter installation procedure, detection will be improved if included, if filter is installed once, and checked at annual maintenance	Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30		4		Low
Filter installation procedure, detection will be improved if included, if filter is installed once, and checked at annual maintenance		Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30	5	3		Low
	Filter installation procedure, detection will be improved if included, if filter is installed once, and checked at annual maintenance	Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30		4		Low
Evaluate addition of redundant check valve also note that A/V is NC unless filling		Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30	5	2		Low
	Evaluate addition of redundant check valve also note that A/V is NC unless filling	Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30		4		Low
Consider redundant thermocouples (TT 4E), control system comparison, calibration plan		Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30	5	2		Low
	Consider redundant thermocouples (TT 4E), control system comparison, calibration plan	Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30		4		Low
Consider positive/pressure ventilation of the train, inert should prevent reaching 20, in a minor leak scenario. Update based		Infelise (Infelise)	13-Mar-2020 15:30	Completed		13-Mar-2020 15:30	5	2		Low

style

(Optional) Defines a style to the column data based on conditions.

You must define at least one **columnStyle** inside this element.

Attributes

None.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">

```

```

<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id=.../>
</columnGroup>
...
</objectSet>
<style>
    <columnStyle id="col_Opt_Detection">
    <columnHeaderStyle cssClass="aw-qualityFmea-formsheet
    ColumnHeaderVerticleText" width="70px"/>
<template>
    <if>
    <conditions>
    <condition operator="AND">
    <expression statement="col_Opt_Detection == null"/>
    <expression statement="col_Opt_Occurrence != null"/>
    </condition>
    </conditions>
    <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
    </if>
    </template>
    </columnStyle>
    ...
</style>
</rendering>

```

columnStyle

Groups condition elements for the defined column. You can define this element inside the style element. At least, one condition element should be defined in this element.

Attributes

id: (Mandatory) Specifies the unique ID for the column group. The uniqueness of this ID is specific to **columnGroup** and column elements.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">
<row>

```

```

<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id="col12".../>
</columnGroup>
...
</objectSet>
<style>
                <columnStyle id="col_Opt_Detection">
<columnHeaderStyle cssClass="aw-qualityFmea-formsheet
                ColumnHeaderVerticleText" width="70px"/>
<template>
                <if>
                <conditions>
                <condition operator="AND">
                <expression statement="col_Opt_Detection == null" />
                <expression statement="col_Opt_Occurrence != null"/>
                </condition>
                </conditions>
                <class cssClass="aw-qualityFmea-formsheetGreyCell"/>
                </if>
                </template>
                </columnStyle>                ...
</style>
</rendering>

```

template

Defines style template or condition based styling for the column data.

Attributes

None.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo=" ">
<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>

```

```

<objectSet source=...>
<columnGroup id=...>
<column id=.../>
</columnGroup>
...
</objectSet>
<style>
    <columnStyle id="col_Opt_Occurrence ">
<columnHeaderStyle cssClass="aw-qualityFmea-formsheet
    ColumnHeaderVerticleText" width="70px"/>
<template>
    <if>
    <conditions>
    <condition operator="AND">
    <expression statement=" col_Opt_Occurrence ==
null" />
    <expression statement=" col_Opt_Detection !=
null"/>
    </condition>
    </conditions>
    <class cssClass="aw-qualityFmea-formsheetGreyCell"/>
    </if>
    </template>
    </columnStyle>
    <columnStyle id="col_Opt_Detection">
<columnHeaderStyle cssClass="aw-qualityFmea-formsheet
    ColumnHeaderVerticleText" width="70px"/>
<template>
    <if>
    <conditions>
    <condition operator="AND">
    <expression statement="col_Opt_Detection == null" />
    <expression statement="col_Opt_Occurrence != null"/>
    </condition>
    </conditions>
    <class cssClass="aw-qualityFmea-formsheetGreyCell"/>
    </if>
    </template>
    </columnStyle>
    ...
</style>
</rendering>

```

Occurrence (O)	Detection (D)
3	
	4

if

Adds conditions to template styling.

If else condition is not defined, then the default style is applied to the column data.

Attributes

None.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">
<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id=.../>
</columnGroup>
...
</objectSet>
```

```

<style>
    <columnStyle id="col_Opt_Occurrence ">
    <columnHeaderStyle cssClass="aw-qualityFmea-formsheetColumn
    HeaderVerticleText" width="70px"/>
<template>
    <if>
    <conditions>
    <condition operator="AND">
    <expression statement=" col_Opt_Occurrence ==
null" />
    <expression statement=" col_Opt_Detection != null"/>
    </condition>
    </conditions>
    <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
    </if>
    </template>
</columnStyle>
    <columnStyle id="col_Opt_Detection">
    <columnHeaderStyle cssClass="aw-qualityFmea-formsheet
    ColumnHeaderVerticleText" width="70px"/>
<template>
    <if>
    <conditions>
    <condition operator="AND">
    <expression statement="col_Opt_Detection ==
null" />
    <expression statement="col_Opt_Occurrence !=
null"/>
    </condition>
    </conditions>
    <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
    </if>
    </template>
</columnStyle>
...
</style>
</rendering>

```

Occurrence (O)	Detection (D)
3	
	4

value

Define value for data condition.

Attributes

None.

Example

```
<rendering>
    <column
id="col_Opt_Detection_Action" titleKey="tc_DFMEA_Opt_Detection_Action"
sourceColumn="col_Opt_Action_Grp.object"
name="REF(qam0DependentQualityActions,Qam0QualityAction).object_name"
sortByProperty="qam0Sequence">
        <dataCondition property="qam0QualityActionSubtype"
operation="equals">
            <value>Detection Action</value>
        </dataCondition>
        <dataCondition
property="qam0QualityActionStatus" operation="notEquals">
            <value>Completed</value>
        </dataCondition>
    </column>
</rendering>
```

elseif

Adds conditions to template styling.

If not defined, then the default style is applied to the column data.

Attributes

None.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">
<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id=.../>
</columnGroup>
...
</objectSet>
<style>
                <columnStyle id="col_Opt_Occurrence ">
<columnHeaderStyle cssClass="aw-qualityFmea-formsheet
                ColumnHeaderVerticleText" width="70px"/>
<template>
                <if>
                ...
                </if>
                <elseif>
                <conditions>
                <condition operator="AND">
                <expression statement=" col_Opt_Occurrence ==
null" />
                <expression statement=" col_Opt_Detection != null"/>
                </condition>
                </conditions>
                <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
                </elseif>
                </template>
</columnStyle>

```

```

    ...
</style>
</rendering>

```

else

Adds conditions to template styling.

Only class can be defined in the **else** section.

Attributes

None.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">
<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id=.../>
</columnGroup>
...
</objectSet>
<style>
                <columnStyle id="col_Opt_Occurrence ">
                <columnHeaderStyle cssClass="aw-qualityFmea-formsheet
                ColumnHeaderVerticleText" width="70px"/>
<template>
                <if>
                ...
                </if>
                <elseif>
                ...
                </elseif >
                <else>
                <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
                </else>
                </template>
</columnStyle>

```

```

    ...
</style>
</rendering>

```

condition

Groups multiple conditions using this element.

Attributes

None.

Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">
<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id="col12".../>
</columnGroup>
...
</objectSet>
<style>
                <columnStyle id="col_Opt_Detection">
<columnHeaderStyle cssClass="aw-qualityFmea-formsheet
                ColumnHeaderVerticleText" width="70px"/>
<template>
                <if>
<conditions>
                <condition operator="AND">
                <expression statement="col_Opt_Detection ==
null" />
                <expression statement="col_Opt_Occurrence !=
null"/>
                </condition>
                </conditions>
                <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
                </if>
                </template>
</columnStyle>
                ...

```

```
</style>
</rendering>
```

expression

Define expression as a statement.

Attributes

statement: Define sub expression in string format.

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rendering SYSTEM "FormSheetConfigurationDTD.dtd" >
<rendering>
<formsheetHeader logo="">
<row>
<header titleKey=.../>
</row>
...
</formsheetHeader>
<objectSet source=...>
<columnGroup id=...>
<column id="col12".../>
</columnGroup>
...
</objectSet>
<style>
                <columnStyle id="col_Opt_Detection">
<columnHeaderStyle cssClass="aw-qualityFmea-formsheet
                ColumnHeaderVerticleText" width="70px"/>
<template>
                <if>
                <conditions>
                <condition operator="AND">
                <expression statement="col_Opt_Detection == null" />
                <expression statement="col_Opt_Occurrence != null"/>
                </condition>
                </conditions>
                <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
                </if>
                </template>
                </columnStyle>                ...
</style>
</rendering>
```

Customization tasks using the styling file

Configure custom styling using the styling file

The styling file works on the principles of cascading style sheets (CSS), which is used for web pages. To style the formsheet using the styling file, you must perform the following steps:

1. In the configuration file, specify the CSS class name of the target element using the **cssClass** attribute. The value of the **cssClass** attribute must be the name of a CSS class.

In the example, `green-color` is the CSS class name.

```
<columnDataStyle cssClass="green-color"/>
```

2. In the styling file, define the CSS class name, that was used in the configuration file, in the following format:

```
.<className> {
    <propertyName>: <value>;
}
```

Example:

```
.green-color {
    color: green;
}
```

Based on the definition in the styling file, the custom style is reflected on the target element.

For examples about the scope of styling, see:

- [Customize the styling for column headers, column group headers or cells](#)
- [Customize the styling for specific column headers](#)
- [Customize the styling of cells of a specific column](#)

3. Save and close the files.

For defining styling classes in styling file, basic knowledge about CSS (Cascading Styling Sheet) is required.

After you update the configuration file and the styling file, make sure that both the files are attached to the guideline object.

Customize the styling for column headers, column group headers, or cells

To define custom styling for column headers, column group headers, or cells, in the configuration file, add the **cssClass** attribute to the following elements within the **<formsheetTableStyle>** tag.

Align the text in all column-group headers to the center

To align the text in all column-group headers to the center, in the configuration file, define the **cssClass** attribute for the **<columnGroupStyle>** element as follows:

```
<formsheetTableStyle>
  <columnGroupStyle cssClass="aw-qualityFmea-
formsheetTextAlignCenter" />
</formsheetTableStyle>
```

Additionally, in the styling file, define the class **aw-qualityFmea-formsheetTextAlignCenter** as follows:

```
/* align text to center */
.aw-qualityFmea-formsheetTextAlignCenter{
  text-align: center !important;
}
```

The highlighted column group header text is center aligned.

Overview History Formsheet

FORMSHEET HEADER

DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)						
PLANNING AND PREPARATION (STEP 1)						
Company Name		Subject	Car	DFMEA ID Number	FMEA-00000015	
Engineering Location		DFMEA Start Date	23-Sep-2020 06:21	Design Responsibility	ed (ed)	
Customer Name		DFMEA Revision Date		Confidentiality Level		
Model Year/Platform		Cross Functional Team				

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)		
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function
	Break_36021939			Break_Function_16183504	
				Break_Function_53371504	
				Break_Function_13840504	
	Break 83070939			Break Function 98275504	

Align the text in all column headers to the center

To align the text in all column headers to the center, in the configuration file, define the **cssClass** attribute for the **<columnHeaderStyle>** element as follows:

```
<formsheetTableStyle>
  <columnHeaderStyle cssClass="aw-qualityFmea-
formsheetTextAlignCenter" width="200px" />
</formsheetTableStyle>
```

Additionally, in the styling file, define the class `aw-qualityFmea-formsheetTextAlignCenter` as follows:

```
/* align text to center */
.aw-qualityFmea-formsheetTextAlignCenter{
  text-align: center !important;
}
```

The highlighted column header text is aligned to the center.

The screenshot shows the 'Formsheet' tab in a software application. It displays a table for 'DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)'. The table is divided into sections: 'PLANNING AND PREPARATION (STEP 1)', 'STRUCTURE ANALYSIS (STEP 2)', and 'FUNCTIONAL ANALYSIS (STEP 3)'. The 'FUNCTIONAL ANALYSIS' section has three columns: '1. Next Higher Level', '2. Focus Element Function and Requirement or Characteristic', and '3. Next Lower Level Function'. The text in these column headers is centered, which is highlighted by a red box in the image.

DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 1)					
Company Name		Subject	Car	DFMEA ID Number	FMEA-00000015
Engineering Location		DFMEA Start Date	23-Sep-2020 06:21	Design Responsibility	ed (ed)
Customer Name		DFMEA Revision Date		Confidentiality Level	
Model Year/Platform		Cross Functional Team			
STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)		
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function
	Break_36021939			Break_Function_16183504	
				Break_Function_53371504	
				Break_Function_13840504	
	Break_83070939			Break_Function_98275504	
	Break_04600028			Break_Function_13840504	

Align the text in all data cells to the left

To align the text in all column headers to left, in the configuration file, define the `cssClass` attribute for the `columnDataStyle` element as follows:

```
<formsheetTableStyle>
  <columnDataStyle cssClass="aw-qualityFmea-formsheetTextAlignLeft" />
</formsheetTableStyle>
```

Additionally, in the styling file, define the class `aw-qualityFmea-formsheetTextAlignLeft` as follows:

```
/* align text to left */
.aw-qualityFmea-formsheetTextAlignLeft{
```

```

    text-align: left !important;
}
    
```

The highlighted column group header text is aligned to the left.

Overview History Formsheet



FORMSHEET HEADER					
DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 1)					
Company Name		Subject	Car	DFMEA ID Number	FMEA-000000015
Engineering Location		DFMEA Start Date	23-Sep-2020 06:21	Design Responsibility	ed (ed)
Customer Name		DFMEA Revision Date		Confidentiality Level	
Model Year/Platform		Cross Functional Team			
STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)		
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic	3. Next Lower Level Function
	Break_36021939			Break_Function_16183504	
				Break_Function_53371504	
				Break_Function_13840504	
	Break_83070939			Break_Function_98275504	
	Break_94600939			Clutch paddle	

Customize the styling for specific column headers

To define custom styling for column headers in the configuration file, add the **cssClass** attribute to the following elements within the `<formsheetTableStyle>` tag.

Align a specific column header text vertically

To align the text in a specific column header vertically, in the configuration file, define the **cssClass** attribute for the `<columnHeaderStyle>` tag inside the target column's `<columnStyle>` element as follows:

```

<style>
  <columnStyle id="col_Failure_Effect_Severity">
    <columnHeaderStyle cssClass="aw-qualityFmea-
formsheetColumnHeaderVerticleText" width="70px"/>
  </columnStyle>
</style>
    
```

Additionally, in the styling file, define the class `aw-qualityFmea-formsheetColumnHeaderVerticleText` as follows.

```

/* vertical header text*/
.aw-qualityFmea-formsheetColumnHeaderVerticleText span {
  transform: rotate(270deg);
  white-space: initial;
  display: block;
  margin-top: 20px;
  height: 60px;
  padding-top: 30px;
}

```

Overview History Formsheet

▼ FORMSHEET HEADER					
DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 1)					
Company Name		Subject		Car	
Engineering Location		DFMEA Start Date		23-Sep-2020 06:21	
Customer Name		DFMEA Revision Date			
Model Year/Platform		Cross Functional Team			
FAILURE ANALYSIS (STEP 4)					
Power Level Function	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Pre

- If **cssClass** is defined in the **<columnHeaderStyle>** element of a specific column's header, the common CSS style defined under the **<formsheetTableStyle>** for **<columnHeaderStyle>** is not applied.

In the following example, only the **aw-qualityFmea-formsheetColumnHeaderVerticleText** CSS class is applied to the **col_Failure_Effect_Severity** column header text.

```

<formsheetTableStyle>
  <columnHeaderStyle cssClass="bold-text" width="200px"/>
</formsheetTableStyle>

```

```

.....
.....
<style>
  <columnStyle id="col_Failure_Effect_Severity">
    <columnHeaderStyle cssClass="aw-qualityFmea-
formsheetColumnHeaderVerticleText" width="70px"/>
  </columnStyle>
</style>

```

- In the styling file, to apply multiple styles to column header text, all the CSS class definitions must be included in the **<columnHeaderStyle>** element.

In the example, both the CSS classes **bold-text** and **aw-qualityFmea-formsheetColumnHeaderVerticleText** are applied to the **col_Failure_Effect_Severity** column header text, which results in the column header being displayed in bold and being vertically oriented.

```

<style>
  <columnStyle id="col_Failure_Effect_Severity">
    <columnHeaderStyle cssClass=" bold-text aw-qualityFmea-
formsheetColumnHeaderVerticleText" width="70px"/>
  </columnStyle>
</style>

```

- In the configuration file, more than one CSS class can be specified in the **cssClass** attribute definition by inserting a single space between the two class names. The syntax for doing this is as follows:

```
cssClass="<class1> <class2> <class3> <class4>
```

Example:

```
<columnHeaderStyle cssClass=" bold-text green-color left-align"
width="70px" />
```

Customize the styling of cells of a specific column

To define custom styling for the data cells in any specific column, in the configuration file, add the **cssClass** attribute to various elements within the **<formsheetTableStyle>** tag.

Define the color of text in a column cell

To specify a color (*green* in this example) for the text in the data cells in a specific column, in the configuration file, define the **cssClass** attribute for the **<columnDataStyle>** tag inside the target column's **<columnStyle>** element as follows:

```

<style>
  <columnStyle id="col_Focus_Sys_Ele">
    <columnDataStyle cssClass="green-text" />

```

```

    </columnStyle>
</style>

```

Additionally, in the styling file, define the class **green-text** such that it specifies the color of the text as green.

```

.green-text {
  color: green;
}

```

Overview History Formsheet

▼ FORMSHEET HEADER

DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)				
PLANNING AND PREPARATION (STEP 1)				
Company Name		Subject	Car	DFMEA ID
Engineering Location		DFMEA Start Date	23-Sep-2020 06:21	Design Resp
Customer Name		DFMEA Revision Date		Confidentia
Model Year/Platform		Cross Functional Team		

STRUCTURE ANALYSIS (STEP 2)			FUNCTIONAL ANALYSIS (STEP 3)	
1. Next Higher Level	2. Focus Element	3. Next Lower Level	1. Next Higher Level Function	2. Focus Element Function and Requirement or Characteristic
	Break_36021939			Break_Function_16183504
				Break_Function_53371504
				Break_Function_13840504
	Break_83070939			Break_Function_98275504
	Break_94600939			Clutch paddle

- If **cssClass** is defined for a specific column's data cells using the **<columnDataStyle>** tag, then the common CSS style specified under **<formsheetTableStyle>** for **<columnDataStyle>** is not applied.

In the example, the **green-text** CSS class is applied for the header text of column **col_Focus_Sys_Ele** only.

```

<formsheetTableStyle>
  <columnDataStyle cssClass="bold-text" width="200px" />
</formsheetTableStyle>
.....
.....
<style>
  <columnStyle id="col_Focus_Sys_Ele">
    <columnDataStyle cssClass="green-text" width="70px" />

```

```

    </columnStyle>
</style>

```

- In the styling file, to apply multiple styles to text in the data cell, all the CSS class definitions must be included in the **<columnDataStyle>** element.

In the example, both the CSS classes **bold-text** and **green-text** are applied to the cell in the **col_Focus_Sys_Ele** column, which results in the column header being displayed in bold and green in color.

```

<style>
  <columnStyle id="col_Focus_Sys_Ele">
    <columnDataStyle cssClass=" bold-text green-text" width="70px"/>
  </columnStyle>
</style>

```

- In the configuration file, more than one CSS class can be specified in the **cssClass** attribute definition by inserting a single space between the two class names. The syntax for doing this is as follows:

```
cssClass="<class1> <class2> <class3> <class4>.....<classN>"
```

Example:

```

<columnDataStyle cssClass=" bold-text green-color left-align"
width="70px" />


```

Attach the formsheet styling file to the guideline object

The formsheet styling file can be used to specify the customized font size, color, alignment, and orientation of the text, and the background color of cells for the contents of the formsheet.

After you download and modify the styling file, you can attach the formsheet styling file to the guideline object, or use the **create_fmea_guideline** utility to attach the formsheet styling file to the guideline object.

Attach the formsheet styling file to the guideline object

1. Log on as a user with DBA privileges.
2. Search for the guideline object.
 - a. Type the search criteria as **guidelines** in the global search, and click .


The search displays the **Design FMEA** and **Process FMEA** guideline objects.

- b. Open the **Design FMEA** or the **Process FMEA** object.

The **Overview** tab displays the creation date, date released, and release status.

3. Add the styling file.

You must delete the existing styling file before you can add a new one.

- a. In the **STYLING FILE** section, from the primary toolbar, click **Edit**  > **Delete**.


The existing styling file is deleted.

- b. To add a new styling file, in the **STYLING FILE** section, from the primary toolbar, click **Add** .

- c. Choose the styling file, and click **Add**.

Note:

The styling file must have a .css extension.

- d. To replace the configuration file, from the primary toolbar, click **Edit**  > **Replace**, choose the XML file, and click **Replace**.

Use the `create_fmea_guideline` utility to attach the formsheet styling file to the guideline object

1. Open the *FMEAGuideline.xml* file for the required guideline object element tag.
2. Make sure that the value of `formSheetStylingFile` attribute is set to specify the name of the styling file.

```
<FMEAGuideline name="D-FMEA Guidelines"
  type="Design" formSheetConfigFile="DesignFormSheetConfiguration.xml"
  formSheetStylingFile="StylingFile.css">
```

3. Run the `create_fmea_guideline` utility.

This creates a new guideline object or updates an existing guideline object and also attaches the specified styling file to it.


Note:

The styling file must have a .css extension.


To add or replace a styling file from the UI, the user must have the appropriate access rights as per the access control list (ACL). Users with DBA group can add or replace the styling file.

Customization tasks using the configuration file

Download the configuration file to customize the formsheet

1. Log on as a user with DBA privileges.
2. Search for the guideline object.
 - a. Type the search criteria as **guidelines** in the global search, and click .

The search displays the **Design FMEA** and **Process FMEA** guideline objects.

 - b. Open the **Design FMEA** or the **Process FMEA** object.
3. In the **CONFIGURATION FILE** section, click **Download File** .

Add a header row and property elements in the formsheet header

1. **Download the configuration file to customize the formsheet.**
2. Open the configuration file and go to the **formsheetHeader** section.

The form sheet header is rendered as follows:

Hydrogen Station Equipment Performance Device					
Exit Full Screen					
FORMSHEET HEADER					
DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 1)					
Company Name	Siemens	Subject	Hydrogen Station Equipment Performance Device	DFMEA ID Number	FMEA-00000001
Engineering Location		DFMEA Start Date	07-May-2020 18:17	Design Responsibility	ed (ed)
Customer Name	Siemens	DFMEA Revision Date		Confidentiality Level	
Model Year/Platform	2019 /	Cross Functional Team			

3. Edit the formsheet header.

```
<formsheetHeader logo=" " >
  <row>
    <header titleKey="tc_PFMEA_Fmea_Header" />
  </row>
  <row>
    <header titleKey="tc_Planning_AND_Preparation" />
  </row>
</formsheetHeader>
```

Attribute	Description
header	Specifies the text for the header table.
titleKey	(Optional) Specifies the key used for localization.
row	Groups the header and properties in a table row. This is a container for the header and property cells when defined for the formsheet row.

Note:

logo as an element is not supported.

- Edit the first **header** element and specify the appropriate text for the header.

Example: `<header titleKey="My Text" />`

In the above example, `<header titleKey="tc_PFMEA_Fmea_Header"/>` resolves to the following text in **en_US** by default:

```
<key id="tc_DFMEA_Fmea_Header">DESIGN FAILURE MODE AND EFFECTS ANALYSIS
(DSIGN FMEA)</key>
```

The localization strings are defined in the `TC_ROOT\lang\textserver\en_US\qfm0qcfmea_text_locale.xml` file.

- Edit the second **header** element and specify the appropriate text for the header.

Example: `<header titleKey="My Text" />`

In the above example, `<header titleKey="tc_Planning_AND_Preparation"/>` resolves to the following text in **en_US**:

```
<key id="tc_Planning_AND_Preparation">PLANNING AND PREPARATION (STEP
1)</key>
```

- Edit the properties and specify values as appropriate. Example: `<property titleKey="My Text" />`

You can add DCP properties in the **name** attribute, for example, `name="GRM(HasParticipant,GroupMember).user"`

- Save the configuration file and **attach it to the guideline object**.
- Open the **Formsheet** and view the changes.

Add a column to the formsheet

1. **Download the configuration file to customize the formsheet.**
2. Open the configuration file and go to the **columnGroup** section.

The columns are rendered as follows:

Hydrogen Station Equipment Performance Device						
FORMSHEET HEADER						
STRUCTURE ANALYSIS (STEP 2)	FUNCTIONAL ANALYSIS (STEP 3)	FAILURE ANALYSIS (STEP 4)				
Focus Element	Focus Element Function and Requirement	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC
HySTEP						
	Contain Hydrogen	Explosive atmosphere inside trailer	9	Loss of containment (minor leakage)	Component Leak	Rated components, acceptance testing, checks
	Unidirectional Hydrogen Passage from Nozzle	Leak to atmosphere	9	Hydrogen flow back through receptacle	Check valve failure	SAE J2600 H70 receptacle

In the above rendering, the first column **STRUCTURE ANALYSIS (STEP 2)** is defined by the first column group, that is, `<columnGroup id="colGrp1" titleKey="tc_Structure_Analysis">` in the **columnGroup** section. Similarly, the second column, **FUNCTIONAL ANALYSIS (STEP 3)** is defined by the second column group, that is, `<columnGroup id="colGrp2" titleKey="tc_Functional_Analysis">`. The other columns are similarly defined by the other columns in the **columnGroup** section.

3. To add a column, for example, **Initial RPN** in the **Risk ANALYSIS (STEP 5)** column:

```
<columnGroup id="colGrp4" titleKey="tc_Risk_Analysis">
  <column id="col7" titleKey="tc_Current_Pre_Ctrl_Action"
    sourceColumn="col6.relation_object"
    name="REF(qfm0PreventionCtrlActions,Qam0QualityAction)
    .object_name"/>
  <column id="col8" titleKey="tc_Occurrence_FC"
    sourceColumn="col7.object" name="qam0Occurrence"/>
  <column id="col9" titleKey="tc_Current_Detection_Ctrl_Action"
    sourceColumn="col6.relation_object"
    name="REF(qfm0DetectionCtrlActions,Qam0QualityAction)
    .object_name"/>
  <column id="col10" titleKey="tc_Detection_FC_FM"
```

```

        sourceColumn="col9.object"    name="qam0Detection" />
    <column id="col11" titleKey="tc_DFMEA_AP"
        sourceColumn="col6.relation_object"
        name="qfm0ActionPriority" />
    <column id="col50" titleKey="Initial RPN"
        sourceColumn="col6.relation_object" name="qfm0InitialRPN"/>
</columnGroup>

```

In the above example, **titleKey** specifies the name of the column and **name="qfm0InitialRPN"** specifies the database name of the property for which you want the values to be displayed.

Risk ANALYSIS (STEP 5)						
Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP	Initial RPN	DFMEA Prevention Action
Rated components, acceptance testing, checks	4	Operator inspection, hydrogen sensors	4	High	144	Consider positive/passive ventilation of the trailer. Vent should prevent reaching LFL in a minor leak scenario. Update based on
SAE J2600 H70 receptacle	3	Operator inspection	4	Low	108	Evaluate addition of redundant check valve also note that AV5 is NC unless filling.
Thermocouple, dispenser credentials	2	Operator inspection	5	Medium	90	Consider redundant thermocouples (T 46), control system comparison, calibration plan

- Save the configuration file and **attach it to the guideline object**.
- Open the **Formsheet** and view the changes.

Hide or display columns or column groups in formsheets and formsheet reports

The `hidden` attribute can be applied to hide `<column>` or `<columnGroup>` elements in the formsheet. This attribute can have following values:

- `true`

The column or column group is hidden or not displayed in the **Formsheet** and **Formsheet Report**. All other columns and their values, irrespective of whether they depend on the column that is hidden or not, are not impacted and are displayed.

- false

The column or column group is not hidden and is displayed in the **Formsheet** and **Formsheet Report**. This is the default value of the attribute.

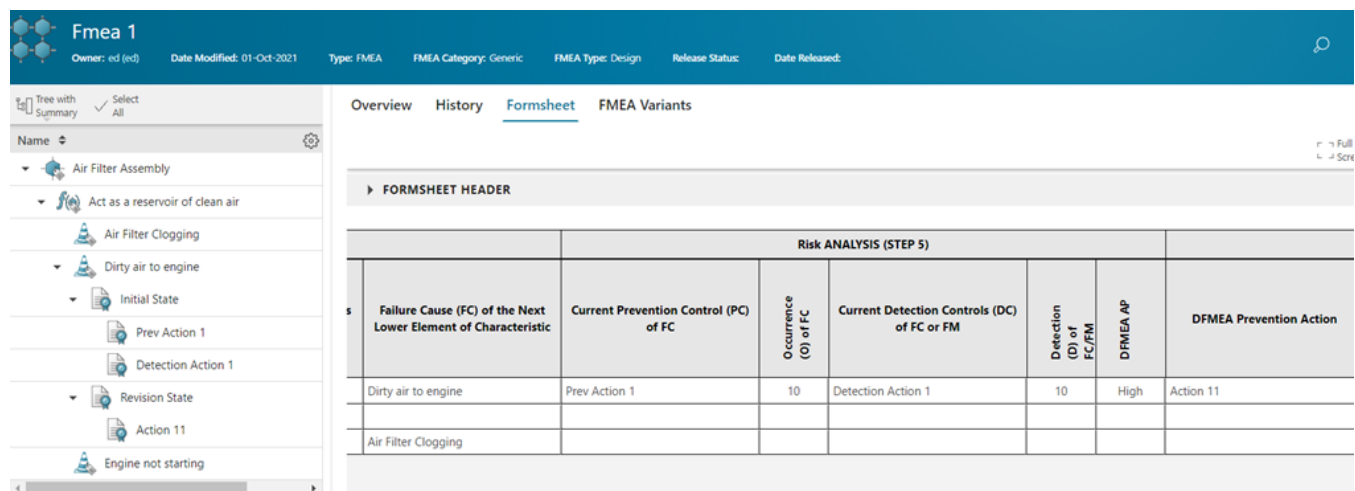
The `hidden` attribute is optional. If you do not include this attribute on any `<column>` or `<columnGroup>`, then the default value is taken as false, and the corresponding column or column group is displayed.

Hide a column in the Formsheet and Formsheet Report

The following code shows examples where the column is hidden in the **Formsheet** or **Formsheet Report**.

```
<columnGroup id="colGrp4" titleKey="tc_Risk_Analysis">
  <column id="col_Ctrl_Action_Grp"
titleKey="tc_Control_Action_Group" hidden="true"
sourceColumn="col_Failure_Cause.object"
name="GRM(Qfm0ControlAction,Qam0QualityAction).object_name" />
</columnGroup>
```

Formsheet does not display the **Control Action Group** column.



The screenshot shows the Fmea 1 interface with a tree view on the left and a formsheet table on the right. The table is titled "Risk ANALYSIS (STEP 5)" and has the following columns:

Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP	DFMEA Prevention Action
Dirty air to engine	Prev Action 1	10	Detection Action 1	10	High	Action 11
Air Filter Clogging						

Formsheet Report does not display the Control Action Group

Cross Functional Team							
E ANALYSIS (STEP 4)			Risk ANALYSIS (STEP 5)				
(S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP
10	Air Filter Clogging	Dirty air to engine	Prev Action 1	10	Detection Action 1	10	High
10	Dirty air to engine						
	Engine not starting	Air Filter Clogging					

Display a column in the Formsheet and Formsheet Report

The following code shows examples where the column is displayed in the **Formsheet** or **Formsheet Report**.

The `hidden` attribute is set to false

```
<columnGroup id="colGrp4" titleKey="tc_Risk_Analysis">
  <column id="col_Ctrl_Action_Grp"
  titleKey="tc_Control_Action_Group" hidden="false"
  sourceColumn="col_Failure_Cause.object"
  name="GRM(Qfm0ControlAction,Qam0QualityAction).object_name"/>
</columnGroup>
```

OR

The `hidden` attribute is missing

```
<columnGroup id="colGrp4" titleKey="tc_Risk_Analysis">
  <column
  id="col_Ctrl_Action_Grp" titleKey="tc_Control_Action_Group"
  sourceColumn="col_Failure_Cause.object"
  name="GRM(Qfm0ControlAction,Qam0QualityAction).object_name"/>
</columnGroup>
```

The **Control Action Group** column is displayed in the **Formsheet**.

The screenshot shows the 'Fmea 1' interface with a tree view on the left and a 'Formsheets' report on the right. The report is titled 'FORMSHEET HEADER' and 'FAILURE ANALYSIS (STEP 4)'. The table below shows the data for this report, with the 'Control Action Group' column highlighted in red.

Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Control Action Group	Current Prevention Control of FC
Air Filter Clogging	Dirty air to engine	Initial State	Prev Action 1
Dirty air to engine			
Engine not starting	Air Filter Clogging		

The **Control Action Group** column is displayed in the **Formsheet Report**.

PLANNING AND PREPARATION (STEP 1)									
Subject					Fmea 1				
DFMEA Start Date					01-Oct-2021 11:08				
DFMEA Revision Date									
Cross Functional Team									
FAILURE ANALYSIS (STEP 4)			Risk ANALYSIS (STEP 5)						
(S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Control Action Group	Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP	DFMEA Prevention Action
10	Air Filter Clogging	Dirty air to engine	Initial State	Prev Action 1	10	Detection Action 1	10	High	Action 11
10	Dirty air to engine								
	Engine not starting	Air Filter Clogging							

Hide a column group in the Formsheet and Formsheet Report

If the hidden attribute is set to `true` for any column group, then neither that column group nor its child columns are displayed in the **Formsheet** and **Formsheet Report**.

```
<columnGroup id="colGrp5" titleKey="tc_Optimization" hidden="true">
  <column id="col_Opt_Action_Grp" titleKey="tc_Opt_Action_Group"
  hidden="true" sourceColumn="col_Failure_Cause.object"
  name="GRM(Qfm0OptimizationAction,Qam0QualityAction).object_name"/>
  <column
  id="col_Opt_Prev_Action" titleKey="tc_DFMEA_Opt_Prev_Action"
  sourceColumn="col_Opt_Action_Grp.object"
  name="REF(qam0PreventiveActions,Qam0QualityAction).object_name"
  alternateDisplayCol="col_Opt_Detection_Action"/>
</columnGroup>
```

Optimization Analysis (Step 5) is hidden in the **Formsheet**.

FORMSHEET HEADER							
FAILURE ANALYSIS (STEP 4)			Risk ANALYSIS (STEP 5)				
(S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP
	Air Filter Clogging	Dirty air to engine	Prev Action 1	10	Detection Action 1	10	High
	Dirty air to engine						
	Engine not starting	Air Filter Clogging					

Optimization Analysis (Step 5) is hidden in the **Formsheet Report**.

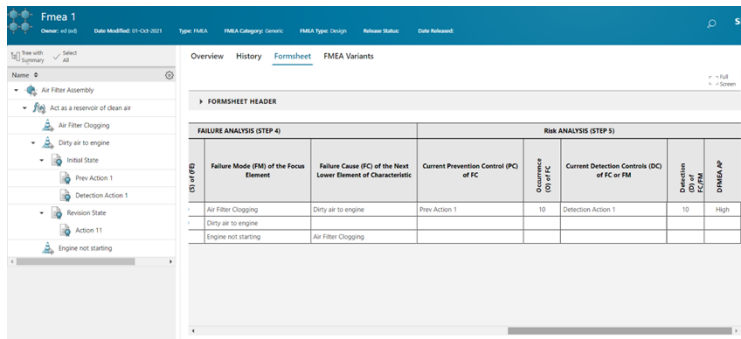
Cross Functional Team							
FAILURE ANALYSIS (STEP 4)			Risk ANALYSIS (STEP 5)				
(S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP
10	Air Filter Clogging	Dirty air to engine	Prev Action 1	10	Detection Action 1	10	High
10	Dirty air to engine						
	Engine not starting	Air Filter Clogging					

Hide column groups in the Formsheets and Formsheets Report

For a column group, if the `hidden` attribute is not set (`false` by default), but for all the child columns, the `hidden` attribute is set to `true`, then neither the column group nor its child columns are displayed.

```
<columnGroup id="colGrp5" titleKey="tc_Optimization">
  <column id="col_Opt_Action_Grp" titleKey="tc_Opt_Action_Group"
  hidden="true" sourceColumn="col_Failure_Cause.object"
  name="GRM(Qfm0OptimizationAction,Qam0QualityAction).object_name"/>
  <column
  id="col_Opt_Prev_Action" titleKey="tc_DFMEA_Opt_Prev_Action"
  hidden="true" sourceColumn="col_Opt_Action_Grp.object"
  name="REF(qam0PreventiveActions,Qam0QualityAction).object_name"
  alternateDisplayCol="col_Opt_Detection_Action"/>
</columnGroup>
```

Optimization (Step 6) and all its child columns are not displayed in the **Formsheet**.



Optimization (Step 6) and all its child columns are not displayed in the **Formsheet Report**.

DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)								
PLANNING AND PREPARATION (STEP 1)								
Fmea 1		DFMEA ID Number		FMEA-00000001				
01-Oct-2021 11:08		Design Responsibility		ed				
		Confidentiality Level						
FAILURE ANALYSIS (STEP 4)				Risk ANALYSIS (STEP 5)				
Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC	Occurrence (O) of FC	Current Detection Controls (DC) of FC or FM	Detection (D) of FC/FM	DFMEA AP
Engine not starting	10	Air Filter Clogging	Dirty air to engine	Prev Action 1	10	Detection Action 1	10	High
Air Filter Clogging	10	Dirty air to engine						
		Engine not starting	Air Filter Clogging					

Remove a column from the formsheet

1. Download the configuration file to customize the formsheet.
2. Open the configuration file and go to the **columnGroup** section.

The columns are rendered as follows:

Hydrogen Station Equipment Performance Device						
FORMSHEET HEADER						
STRUCTURE ANALYSIS (STEP 2)	FUNCTIONAL ANALYSIS (STEP 3)	FAILURE ANALYSIS (STEP 4)				
Focus Element	Focus Element Function and Requirement	Failure Effects (FE)	Severity (S) of (FE)	Failure Mode (FM) of the Focus Element	Failure Cause (FC) of the Next Lower Element of Characteristic	Current Prevention Control (PC) of FC
HySTEP						
	Contain Hydrogen	Explosive atmosphere inside trailer	9	Loss of containment (minor leakage)	Component Leak	Rated components, acceptance testing, checks
	Unidirectional Hydrogen Passage from Nozzle	Leak to atmosphere	9	Hydrogen flow back through receptacle	Check valve failure	SAE J2600 H70 receptacle

In the above rendering the first column, **STRUCTURE ANALYSIS (STEP 2)** is defined by the first column group, that is, `<columnGroup id="colGrp1" titleKey="tc_Structure_Analysis">` in the **columnGroup** section. Similarly, the second column, **FUNCTIONAL ANALYSIS (STEP 3)** is defined by the second column group, that is, `<columnGroup id="colGrp2" titleKey="tc_Functional_Analysis">`. The other columns are similarly defined by the other columns in the **columnGroup** section.

3. You can remove a column or remove or hide the column ID. In the following example, there is only one column ID. Therefore, you must remove or hide the entire complete column group.

```
<columnGroup id="colGrp1" titleKey="tc_Structure_Analysis">
  <column id="col1" titleKey="tc_PFMEA_Focus_Element"
sourceColumn=" "
      name="object_name" />
</columnGroup>
```

4. In the above example, the **col1** column ID is referred to by the **sourceColumn** in the second column group as follows:

```
<columnGroup id="colGrp2" titleKey="tc_Functional_Analysis">
  <column id="col2" titleKey="tc_PFMEA_Function_Process"
    sourceColumn="col1.object"
    name="REF(qfm0FunctionsElementsList,Qfm0FunctionElement)
    .object_name"/>
</columnGroup>
```

In such cases, you must change the value of the **sourceColumn**. You can enter a blank value as follows.

Example:

```
<columnGroup id="colGrp2" titleKey="tc_Functional_Analysis">
  <column id="col2" titleKey="tc_PFMEA_Function_Process"
    sourceColumn=""
    name="REF(qfm0FunctionsElementsList,Qfm0FunctionElement)
    .object_name"/>
</columnGroup>
```

5. Save the configuration file and **attach it to the guideline object**.
6. Open the **Formsheet** and view the changes.

Concatenate data for properties in the formsheet header

1. **Download the configuration file to customize the formsheet**.
2. Open the configuration file and go to the **formsheetHeader section**.

Example:

```
<formsheetHeader logo="">
  <row>
    <header titleKey="tc_PFMEA_Fmea_Header"/>
  </row>
  <row>
    <header titleKey="tc_Planning_AND_Preparation"/>
  </row>
  <row>
    <property titleKey="tc_Company_Name" name="qfm0CompanyName"/>
    <property titleKey="tc_Subject" name="object_name"/>
    <property titleKey="tc_PFMEA_ID_Number" name="qfm0Id"/>
  </row>
  <row>
    <property titleKey="tc_Engineering_Location" name=""/>
    <property titleKey="tc_PFMEA_Start_Date" name="creation_date"/>
    <property titleKey="tc_PFMEA_Responsibility"
```

```

name="owning_user" />
</row>
<row>
  <property titleKey="tc_Customer_Name" name="qfm0Customer" />
  <property titleKey="tc_PFMEA_Revision_Date" name="" />
  <property titleKey="tc_Confidentiality_level" name="" />
</row>
<row>
  <property titleKey="tc_ModelYear_Platform"
    name="CONCAT(qfm0ModelYear,/,qfm0Platform)" />
  <property titleKey="tc_Cross_Functional_Team"
    name="GRM(HasParticipant,GroupMember).user" />
</row>
</formsheetHeader>

```

The form sheet header is rendered as follows:

Hydrogen Station Equipment Performance Device					
FORMSHEET HEADER					
DESIGN FAILURE MODE AND EFFECTS ANALYSIS (DESIGN FMEA)					
PLANNING AND PREPARATION (STEP 1)					
Company Name	Siemens	Subject	Hydrogen Station Equipment Performance Device	DFMEA ID Number	FMEA-00000001
Engineering Location		DFMEA Start Date	07-May-2020 18:17	Design Responsibility	ed (ed)
Customer Name	Siemens	DFMEA Revision Date		Confidentiality Level	
Model Year/Platform	2019 /	Cross Functional Team			

- (Optional) To concatenate two properties to be displayed, specify the properties as `name="CONCAT(PropertyName,I,PropertyName)"`.

Example (displaying the model year and the platform):

```

<row>
  <property titleKey="tc_ModelYear_Platform"
    name="CONCAT(qfm0ModelYear,/,qfm0Platform)" />
  <property titleKey="tc_Cross_Functional_Team"
    name="GRM(HasParticipant,GroupMember).user" />
</row>

```

- Save the configuration file and **attach it to the guideline object**.
- Open the **Formsheet** and view the changes.

Override the default styling for the column data

- Download the configuration file to customize the formsheet.**

- Open the configuration file and go to the **FormSheet Table Default Styling** section.

This defines the default CSS class and width of each column in the formsheet table.

```
<!-- FormSheet Table Default Styling
// This section is used to define default css class and width of
//      column in formsheet table.
//
// DTD: <!ELEMENT formsheetTableStyle (columnGroupStyle,
//      columnHeaderStyle,columnDataStyle)>
// DTD: <!ELEMENT columnGroupStyle EMPTY>
// DTD: <!ATTLIST columnGroupStyle cssClass CDATA #REQUIRED>
// DTD: <!ELEMENT columnHeaderStyle EMPTY>
// DTD: <!ATTLIST columnHeaderStyle cssClass CDATA #REQUIRED
// DTD:      width NMTOKEN #REQUIRED>
// DTD: <!ELEMENT columnDataStyle EMPTY>
// DTD: <!ATTLIST columnDataStyle cssClass CDATA #REQUIRED>
//
-->
```

You can override the default styling by providing conditional styling for individual columns and column data in the **User defined styling** section.

```
<!-- User defined styling, conditional styling for individual
      column and column data
//
// DTD: <!ELEMENT style (columnStyle)+>
// DTD: <!ELEMENT columnStyle (columnHeaderStyle?,template?)>
// DTD: <!ATTLIST columnStyle id IDREF #REQUIRED>
// DTD: <!ELEMENT columnHeaderStyle EMPTY>
// DTD: <!ATTLIST columnHeaderStyle cssClass CDATA #REQUIRED
// DTD:      width NMTOKEN #REQUIRED>
// DTD: <!ELEMENT template (if,elseif?,else?)>
// DTD: <!ELEMENT if (conditions,class)>
// DTD: <!ELEMENT elseif (conditions,class)>
// DTD: <!ELEMENT else (class)>
// DTD: <!ELEMENT conditions (condition)+>
// DTD: <!ELEMENT condition (expression)+>
// DTD: <!ATTLIST condition operator NMTOKEN #REQUIRED>
// DTD: <!ELEMENT expression EMPTY>
// DTD: <!ATTLIST expression statement CDATA #REQUIRED>
// DTD: <!ELEMENT class EMPTY>
// DTD: <!ATTLIST class cssClass CDATA #REQUIRED>
//
// Example :
      <columnStyle id="ColPrevOptAction" >
        <template>
          <if>
```

```

        <conditions>
          <condition operator="AND">
            <expression statement="ColPrevOptAction == null" />
            <expression statement="ColDetectionOptAction !=
null"/>
          </condition>
        </conditions>
        <class cssClass="aw-qualityFmea-formsheetGreyCell"/>
      </if>
      <elseif>
        <conditions>
          <condition operator="AND">
            <expression statement=
"ColPrevOptAction == null" />
            <expression statement=
"ColDetectionOptAction != null"/>
          </condition>
        </conditions>
        <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
      </elseif>
      <else>
        <class cssClass="aw-qualityFmea-
formsheetGreyCell"/>
      </else>
    </template>
    <columnHeaderStyle cssClass="aw-qualityFmea-formsheetColumn
HeaderText" width="200px"/>
  </columnStyle>
  //
  -->

```

3. To override the default style and to use vertical text for some of the columns, use the **if** elements.

```

<columnStyle id="col19">
  <columnHeaderStyle cssClass="aw-qualityFmea-
formsheetColumnHeader
  VerticleText" width="70px"/>
</columnStyle>
<columnStyle id="col20">
  <columnHeaderStyle cssClass="aw-qualityFmea-
formsheetColumnHeader
  VerticleText" width="70px"/>
<template>
  <if>
    <conditions>
      <condition operator="AND">
        <expression statement="col20 == null" />
        <expression statement="col21 != null"/>

```

```

        </condition>
    </conditions>
    <class cssClass="aw-qualityFmea-formsheetGreyCell" />
</if>
</template>
</columnStyle>

```

Occurrence (O)	Detection (D)
3	
	4

4. Save the configuration file and **attach it to the guideline object**.
5. Open the **Formsheet** and view the changes.

Configure the company logo in the formsheet header

You can configure the company logo displayed in the formsheet header by adding the company logo file to the **data** folder and then updating the configuration file.

Procedure

1. Copy the image file to the *TR\install\qfm0qcfmea\data* folder.
2. Import the image to the reference dataset using the following command:

```

import_file -u=user_name -pf=${TC_USER_PASSWD_FILE} -g=dba
-f=${TC_INSTALL_DIR}/qfm0qcfmea/data/SiemensLogo.svg
-d=NameForNewLogo -ref=Image -type=Image -de=r

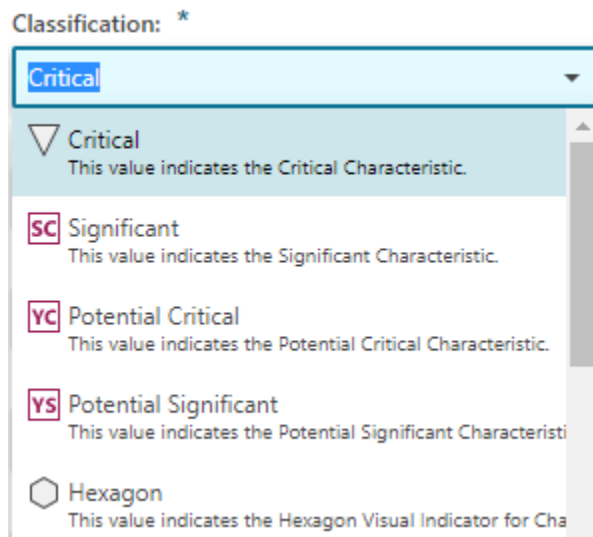
```

3. Add the **NameForNewLogo** from the command in step 2 to the formsheet configuration as follows:

```
<formsheetHeader logo="NameForNewLogo ">
```

Customize the visual indicators for the Classification list of values in a Quality Characteristic for an FMEA

You can customize the visual indicators that appear in the values of the **Classification** list in a Quality Characteristic for an FMEA. This allows a quality engineer to identify the criticality of a specific classification when they are working with a Quality Characteristic.



Procedure

1. In BMIDE, add the custom values of the **Classification** list in the following format:

```
iconName = `indicator'+<name of the value that appears in the Classification list>
```

In this example: `iconName = indicatorCriticalCharacteristic`

The internal name of the **Classification** list is **Qc0Criticality**.

2. Use *configuration builder* to register indicators in *Configuration and Extensibility* in the Teamcenter documentation.

Set up and view audit logs for FMEA objects

Set up audit logs for FMEA objects

To enable FMEA Engineers to view audit logs, you must ensure that you have configured **Audit Manager** using the Business Modeler IDE and set the following preferences:

- Set the `TC_audit_manager_version` preference to **3**.
- Set the `AWC_show_audit_logs` preference to **true**.

View audit logs for FMEA objects

As a dba user, you can view audit logs for system elements, functions, failures, and quality actions in the FMEA master data. The audit logs can help to troubleshoot any errors that might occur.

1. On the **HOME** page, click the **QUALITY MASTER DATA** tile and change the workspace to **Default**.
2. In the **Systems Elements, Functions, Failures, or Quality Actions** tab, select the object for which you want to view the audit logs and click the **Audit Logs** tab.

The **General Logs** table displays the logged date, the event type name, user ID, group name, and other details.

The **Workflow Logs** table displays the workflow-related actions that have been performed for the FMEA object.

3. (Optional)

To view additional information, select an event and click **Information** ⓘ > **All Properties** tab.

The **Information** panel displays the old values and the new values of the fields that are modified.

Allow editing of occurrence and detection values for actions in an action group

The `Fmea_Edit_OD_Value_On_ActionGroup` preference let you specify if users can edit the occurrence and detection values for the actions in an action group.

When you set the value of the `Fmea_Edit_OD_Value_On_ActionGroup` preference to **false**, users can edit the occurrence and detection values for any prevention or detection actions in the action group.

When you set the value of the **Fmea_Edit_OD_Value_On_ActionGroup** preference to **true**, users can edit the occurrence value if at least one prevention action is specified in the action group and detection value if at least one detection action is specified in the action group respectively.

For example, if an action group consists of prevention actions only, and the **Fmea_Edit_OD_Value_On_ActionGroup** preference is set to **true**, then users can edit the occurrence value only. Since no detection actions are specified, users cannot edit the detection value. To be able to edit both the occurrence and detection values as required, users must add a detection action to the action group.

Enable automatic carry over of occurrence and detection values from the initial state actions to the revision state actions

The **Fmea_Carryover_OD_From_Initial_Action** preference lets you specify if the occurrence and detection values from the initial state actions are automatically carried over to the revision state actions.

When you set the **Fmea_Carryover_OD_From_Initial_Action** preference value to **true**, the occurrence and detection values of the initial state actions are automatically carried over to the revision state actions.

When you set the **Fmea_Carryover_OD_From_Initial_Action** preference value to **false**, the occurrence and detection values of the initial state actions are not carried over to the revision state actions, and the values are set to **10**, which is the default value. Users can edit the values for the revision state actions as required.

Specify the display of formsheets

Enable the display of formsheet in the lite mode

You can set the **SE_Formsheet_lite_view_mode** preference to **true** to display the formsheet of the focus system element in a new window. In the lite mode, the formsheet displays the functions, failures, and actions of the focus system element. The information about the children of the focus system element is not included.

The default value of this preference is **false**.

The **Lite** mode is available only when users open the formsheet using the **Open in Formsheet** right-click menu option in the **Tree** view.

Show the header row in the formsheet

The header row in the formsheet displays information such as the company name, company logo, the start and revision date of the FMEA, and the cross functional team. You can set the **FMEA_Header_Always_Collapsed** preference to **false**, so that the formsheet header row is displayed.

The default value of this preference is **true**, and the formsheet header row is collapsed.

Enable cascading of severity values in an FMEA

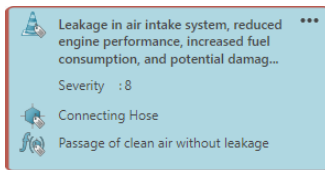
The `FMEA_cascaded_severity` preference lets you specify if the severity values of the highest failure in the FMEA structure is cascaded in the FMEA structure.

When you set the preference to `true`, users can edit the severity value of the highest failure in the FMEA structure, the updated severity value is cascaded to all sublevels in the failure nets.

When you set the preference to `false`, users cannot modify the severity values of the individual failures on the sublevels and the severity value from the higher levels will not overwrite them.

Enable a multi-line display of an FMEA object name

By default, an FMEA object's name is displayed in a single line within a node. This is because the default value of the `Qfm0FMEA_NetView_MultilineEllipsis` preference is `false`. Users can hover over an FMEA object's name to view its complete name.



Set the `Qfm0FMEA_NetView_MultilineEllipsis` preference to `true` to enable multi-line text for displaying an FMEA object's name.

Note:

The multi-line ellipsis preference value is saved when you save the net view settings. When you retrieve the graph with the saved net view settings, the saved preference value is restored.

FMEA workflow handlers

QFM0FMEAREALIZATION-release-fmea-structure

Description

This workflow action handler releases the Failure Mode and Effects Analysis (FMEA) structure which is under the given FMEA root node.

Syntax

```
QFM0FMEAREALIZATION-release-fmea-structure [-retain_release_date=true/false][-exclude_released=true/false]
```

Arguments

-retain_release_date

Indicates whether to keep the released date or not while adding the release status on the FMEA object.

-exclude_released

Does not release an already released FMEA object.

Placement

Requires no specific placement.

Restrictions

None.

Note:

For more information about workflow handlers, see *Workflow Handlers*.

QFM0FMEAREALIZATION-validate-for-class

Description

This workflow rule handler validates the type of object submitted to workflow.

Syntax

QFM0FMEAREALIZATION-validate-for-class -className=[*BusinessObject class name*]

Arguments

-className

Business object class name.

Placement

Place on the **Start** action of a **Do** task.

Restrictions

None.

Note:

For more information about workflow handlers, see *Workflow Handlers*.

6. Administer FMEA

Importing FMEA structures to Teamcenter

You can **import an FMEA structure in the MSR-XML format into Teamcenter**. There are various tags used in the MSR format. However, Teamcenter supports the following tags which when included in the MSR file are imported successfully:

There are various tags used in the MSR format. For a successful import, you must ensure that the following tags are included in the FMEA MSR file.

- COMPANIES{XE "COMPANIES"\b}
- FM-STRUCTURES{XE " FM-STRUCTURES"\b}
- FM-STRUCTURE{XE "FM-STRUCTURE"\b}(FMEA Node)
- FM-STRUCTURE-ROOT{XE"FM-STRUCTURE-ROOT"\b}
- FM-STRUCTURE-ELEMENTS{XE"FM-STRUCTURE-ELEMENTS"\b}
- FM-STRUCTURE-ELEMENT{XE"FM-STRUCTURE-ELEMENTS"\b}
- FM-SE-DECOMPOSITION{XE"FM-SE-DECOMPOSITION"\b}
- FM-SE-FUNCTIONS{XE"FM-SE-FUNCTIONS"\b}
- FM-SE-CHARACTERISTICS{XE"FM-SE-CHARACTERISTICS"\b}
- FM-FUNCTION-TYPES{XE"FM-FUNCTION-TYPES"\b}
- FM-FUNCTION-TYPE{XE"FM-FUNCTION-TYPE"\b}
- FM-FUNCTIONS{XE"FM-FUNCTIONS"\b}
- FM-FUNCTION{XE"FM-FUNCTION"\b}
- FM-USERDEFINED-ATTRIBUTE-REFS{XE"FM-USERDEFINED-ATTRIBUTE-REFS"\b}
- FM-PREREQUISITES{XE"FM-PREREQUISITES"\b}
- FM- FAULT-REFS{XE"FM-FAULT-REFS"\b}
- FM- CHARACTERISTICS{XE"FM-CHARACTERISTICS"\b}

- FM- CHARACTERISTIC{XE"FM-CHARACTERISTIC"\b}
- FM- FAULT-TYPES{XE"FM-FAULT-TYPES"\b}
- FM- FAULT-TYPE{XE"FM-FAULT-TYPE"\b}
- FM- FAULTS{XE"FM-FAULTS"\b}
- FM-FAULT{XE"FM-FAULT"\b}
- FM-CAUSES{XE"FM-CAUSES"\b}
- FM-SIGNIFICANCE{XE"FM-SIGNIFICANCE"\b}
- FM-TASK-HISTORY{XE"FM-TASK-HISTORY"\b}
- FM-TASK-SETS{XE"FM-TASK-SETS"\b}
- FM-TASK-SET{XE"FM-TASK-SET"\b}
- FM-DETECTION-TASKS{XE"FM-DETECTION-TASKS"\b}
- FM-OCCURRENCE-TASKS{XE"FM-OCCURRENCE-TASKS"\b}
- FM-ACTIONS{XE"FM-ACTIONS"\b}
- FM-ACTION{XE"FM-ACTION"\b}

Note:

For detailed information about the XML schema of the tags and the mapping of the element attributes to the objects in Teamcenter, see *MSR FMEA elements and attribute definitions*.

You can also include additional tags in the MSR file. To do that, you must update the **MsrFmeaToPlmxml** dataset. For more information about the MSR FMEA attributes that can be included in the MSR file, other than the mandatory ones, navigate to <http://www.msr-wg.de/medoc/> and refer to the **Downloads** section.


The **MsrFmeaToPlmxml** dataset is the default dataset that contains the mapping file which is required during the import. If you want to use a different dataset other than the default, you must modify the **Qfm0MSRtoPLMXMLMappingFile** preference to specify the name of the dataset which holds the mapping file (xslt transform), which is used to transform the MSR FMEA XML file to the PLMXML format.

Import an FMEA structure

You can import an FMEA structure in MSR-XML format into Teamcenter.

For a successful import, you must ensure that the appropriate tags are included in the FMEA MSR file. For more information about the format of the tags in the MSR format, see [Importing FMEA structures to Teamcenter](#).

Procedure

1. On the **Home** page, click **FOLDERS**.
2. From the primary toolbar, click **New**  **> Import FMEA**.
3. In the **Import FMEA** panel, click **Choose File** and browse to select the file that you want to import.
4. From the **FMEA Guidelines** list, select the required guideline.
5. Click **Import**.

The FMEA is imported in the specified folder.

MSR FMEA elements and attribute definitions

COMPANIES{XE "COMPANIES"\b}

This can be used to describe the companies involved in the project. The company responsible for the generation or maintenance of the file or some parts of the file is determined by a <TEAM-MEMBER-REF> within <ADMIN-DATA>. In simple cases there is only one company.

XML Schema

```
<!ELEMENT COMPANIES (COMPANY+)>
```

Sub elements	FMEA entity attribute for import/export
COMPANY	FMEA preparing company

Element Attribute

```
<!ATTLIST COMPANIES
    S                CDATA        #IMPLIED
    SI               CDATA        #IMPLIED
    SYSCOND          CDATA        #IMPLIED
    T                CDATA        #IMPLIED
    VIEW            CDATA        #IMPLIED
>
```

FM-STRUCTURES{XE "FM-STRUCTURES"\b}

XML Schema

```
<!ELEMENT FM-STRUCTURES (FM-STRUCTURE*)>
```

Sub elements	FMEA entity attribute for import/export
FM-STRUCTURE	FMEA Node

Element Attribute

```
<!ATTLIST FM-STRUCTURES
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND          CDATA        #IMPLIED
  T                CDATA        #IMPLIED
  VIEW             CDATA        #IMPLIED
>
```

FM-STRUCTURE{XE "FM-STRUCTURE"\b}(FMEA Node)

XML Schema

```
<!ELEMENT FM-STRUCTURE ( LONG-NAME , SHORT-NAME? , DESC? , CATEGORY? ,
ADMIN-DATA? , ANNOTATIONS? , FM-STRUCTURE-OWNER? , FM-VARIANT-REFS? ,
FM-INTERFACE? , FM-STRUCTURE-ROOT? , FM-ORPHAN-HOME? , USER-COVER-
SHEETS? , SDGS? , MISC-DATA? ) >
```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	FMEA Node Name if length is < 128 else FMEA Node Desc.
SHORT-NAME	FMEA Node Name if LONG-NAME length is > 128
FM-STRUCTURE-ROOT	Root System Element Reference

Element Attribute

```
<!ATTLIST FM-STRUCTURE
  F-ID-CLASS       NMTOKEN    #FIXED    "FM-STRUCTURE"
  ID               ID          #IMPLIED
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND          CDATA        #IMPLIED
  T                CDATA        #IMPLIED
  VIEW             CDATA        #IMPLIED
>
```

FM-STRUCTURE-ROOT{XE"FM-STRUCTURE-ROOT"\b}

XML Schema

```
<!ELEMENT FM-STRUCTURE-ROOT ( FM-STRUCTURE-ELEMENT-REF ) >
```

Sub elements	FMEA entity attribute for import/export
FM-STRUCTURE-ELEMENT-REF	Reference of Root System Element

Element Attribute

```

<!ATTLIST FM-STRUCTURE-ROOT
  S                CDATA      #IMPLIED
  SI               CDATA      #IMPLIED
  SYSCOND         CDATA      #IMPLIED
  T               CDATA      #IMPLIED
  VIEW            CDATA      #IMPLIED
>

```

FM-STRUCTURE-ELEMENTS{XE"FM-STRUCTURE-ELEMENTS"\b}

XML Schema

```

<!ELEMENT FM-STRUCTURE-ELEMENTS ( FM-STRUCTURE-ELEMENT* ) >

```

Sub elements	FMEA entity attribute for import/export
FM-STRUCTURE-ELEMENT	System element representation

Element Attribute

```

<!ATTLIST FM-STRUCTURE-ELEMENTS
  S                CDATA      #IMPLIED
  SI               CDATA      #IMPLIED
  SYSCOND         CDATA      #IMPLIED
  T               CDATA      #IMPLIED
  VIEW            CDATA      #IMPLIED
>

```

FM-STRUCTURE-ELEMENT{XE"FM-STRUCTURE-ELEMENT"\b}

XML Schema

```

<!ELEMENT FM-STRUCTURE-ELEMENT ( LONG-NAME , SHORT-NAME? , DESC? ,
CATEGORY? , ADMIN-DATA? , ANNOTATIONS? , FM-USERDEFINED-ATTRIBUTE-
REFS? , FM-LINK-DESTINATIONS? , PART-NUMBER? , FM-SE-DECOMPOSITION? ,
FM-SE-FUNCTIONS? , FM-SE-CHARACTERISTICS? , FM-MODULE-REF? , FM-
MACHINE-REFS? , TEAM-MEMBER-REF? , SDGS? , MISC-DATA? ) >

```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	System Element Name if length is < 128 else System Element Description
SHORT-NAME	System Element Name if LONG-NAME length is > 128
FM-SE-DECOMPOSITION	Child System Element representation
FM-SE-FUNCTIONS	Function references
FM-SE-CHARACTERISTICS	Inspection Definition references. Function Elements will be created with same name.

Element Attribute

```

<!ATTLIST FM-STRUCTURE-ELEMENT
  F-ID-CLASS      NMTOKEN   #FIXED   "FM-STRUCTURE-ELEMENT"

```

ID	ID	#IMPLIED
S	CDATA	#IMPLIED
SI	CDATA	#IMPLIED
SYSCOND	CDATA	#IMPLIED
T	CDATA	#IMPLIED
VIEW	CDATA	#IMPLIED

>

FM-SE-DECOMPOSITION{XE"FM-SE-DECOMPOSITION"b**XML Schema**

```
<!ELEMENT FM-SE-DECOMPOSITION
  (
    FM-STRUCTURE-ELEMENT-REF*
  )
>
```

Sub elements	FMEA entity attribute for import/export
FM-STRUCTURE-ELEMENT-REF	Child System Element Representation references.

Element Attribute

```
<!ATTLIST FM-SE-DECOMPOSITION
  S          CDATA      #IMPLIED
  SI         CDATA      #IMPLIED
  SYSCOND    CDATA      #IMPLIED
  T          CDATA      #IMPLIED
  VIEW       CDATA      #IMPLIED
>
```

>

FM-SE-FUNCTIONS{XE"FM-SE-FUNCTIONS"b**XML Schema**

```
<!ELEMENT FM-SE-FUNCTIONS
  (
    FM-FUNCTION-REF*
  )
>
```

Sub elements	FMEA entity attribute for import/export
FM-FUNCTION-REF	Function Element Representation references.

Element Attribute

```
<!ATTLIST FM-SE-FUNCTIONS
  S          CDATA      #IMPLIED
  SI         CDATA      #IMPLIED
  SYSCOND    CDATA      #IMPLIED
  T          CDATA      #IMPLIED
>
```

```

        VIEW                CDATA        #IMPLIED
    >

```

FM-SE-CHARACTERISTICS{XE"FM-SE-CHARACTERISTICS"\b

XML Schema

```

<!ELEMENT FM-SE-CHARACTERISTICS
  (
    FM-CHARACTERISTIC-REF*
  )
>

```

Sub elements	FMEA entity attribute for import/export
FM-CHARACTERISTIC-REF	Inspection Definition references.

Element Attribute

```

<!ATTLIST FM-SE-CHARACTERISTICS
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND          CDATA        #IMPLIED
  T                CDATA        #IMPLIED
  VIEW             CDATA        #IMPLIED
>

```

FM-FUNCTION-TYPES{XE"FM-FUNCTION-TYPES"\b

List of function specifications.

XML Schema

```

<!ELEMENT FM-FUNCTION-TYPES
  (
    FM-FUNCTION-TYPE*
  )
>

```

Sub elements	FMEA entity attribute for import/export
FM-FUNCTION-TYPE	Function specification.

Element Attribute

```

<!ATTLIST FM-FUNCTION-TYPES
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND          CDATA        #IMPLIED
  T                CDATA        #IMPLIED
>

```

```

        VIEW                CDATA        #IMPLIED
    >

```

FM-FUNCTION-TYPE{XE"FM-FUNCTION-TYPE"\b

XML Schema

```

<!ELEMENT FM-FUNCTION-TYPE
    (
        LONG-NAME ,
        SHORT-NAME? ,
        DESC? ,
        CATEGORY? ,
        ADMIN-DATA? ,
        ANNOTATIONS? ,
        FM-FUNCTION-TYPE-DECOMPOSITION? ,
        SDGS? ,
        MISC-DATA?
    )
>

```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	Function Specification Name if length is < 128 else Function Specification Description.
SHORT-NAME	Function Specification Name if LONG-NAME length is > 128

Element Attribute

```

<!ATTLIST FM-FUNCTION-TYPE
    F-ID-CLASS          NMTOKEN    #FIXED    "FM-FUNCTION-TYPE"
    ID                  ID          #IMPLIED
    S                   CDATA      #IMPLIED
    SI                  CDATA      #IMPLIED
    SYSCOND             CDATA      #IMPLIED
    T                   CDATA      #IMPLIED
    VIEW               CDATA      #IMPLIED
>

```

FM-FUNCTIONS{XE"FM-FUNCTIONS"\b

List of function elements.

XML Schema

```

<!ELEMENT FM-FUNCTIONS
    (
        FM-FUNCTION*
    )
>

```

Sub elements	FMEA entity attribute for import/export
FM-FUNCTION	Function element

Element Attribute

```
<!ATTLIST FM-FUNCTIONS
  S                CDATA      #IMPLIED
  SI               CDATA      #IMPLIED
  SYSCOND          CDATA      #IMPLIED
  T                CDATA      #IMPLIED
  VIEW             CDATA      #IMPLIED
>
```

FM-FUNCTION{XE"FM-FUNCTION"\b

XML Schema

```
<!ELEMENT FM-FUNCTION
  (
    LONG-NAME ,
    SHORT-NAME? ,
    DESC? ,
    CATEGORY? ,
    ADMIN-DATA? ,
    ANNOTATIONS? ,
    FM-USERDEFINED-ATTRIBUTE-REFS? ,
    FM-LINK-DESTINATIONS? ,
    FM-FUNCTION-TYPE-REF? ,
    FM-PREREQUISITES? ,
    FM-FAULT-REFS? ,
    FM-CLASSIFICATION? ,
    TEAM-MEMBER-REF? ,
    SDGS? ,
    MISC-DATA?
  )
>
```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	Function Element Name if length is < 128 else Function Element Description
SHORT-NAME	Function Element Name if LONG-NAME length is > 128
FM-USERDEFINED-ATTRIBUTE-REFS	Function Requirement references. Where FM-USERDEFINED-ATTRIBUTE/ CATEGORY is FunctionRequirement. Item type created as reference.
FM-FUNCTION-TYPE-REF	Function Specification reference

Sub elements	FMEA entity attribute for import/export
FM-PREREQUISITES	Next Lower Level Function Element References. In case of Inspection Definition, it's Function Element will be added.
FM-FAULT-REFS	Failure Element references

Element Attribute

```
<!ATTLIST FM-FUNCTION
  F-ID-CLASS          NMTOKEN   #FIXED   "FM-FUNCTION"
  ID                  ID           #IMPLIED
  S                   CDATA       #IMPLIED
  SI                  CDATA       #IMPLIED
  SYSCOND             CDATA       #IMPLIED
  T                   CDATA       #IMPLIED
  VIEW                CDATA       #IMPLIED
>
```

FM-USERDEFINED-ATTRIBUTE-REFS{XE"FM-USERDEFINED-ATTRIBUTE-REFS"\b}

List of function elements.

XML Schema

```
<!ELEMENT FM-USERDEFINED-ATTRIBUTE-REFS
  (
    FM-FM-USERDEFINED-ATTRIBUTE-REFS+
  )
>
```

Sub elements	FMEA entity attribute for import/export
FM-USERDEFINED-ATTRIBUTE-REF	User defined object references

Currently only CATEGORY with FunctionRequirement is supported. Item is created as reference.

Element Attribute

```
<!ATTLIST FM-USERDEFINED-ATTRIBUTE-REFS
  S                   CDATA       #IMPLIED
  SI                  CDATA       #IMPLIED
  SYSCOND             CDATA       #IMPLIED
  T                   CDATA       #IMPLIED
  VIEW                CDATA       #IMPLIED
>
```

FM-PREREQUISITES{XE"FM-PREREQUISITES"\b}

XML Schema

```
<!ELEMENT FM-PREREQUISITES
  (
```

```

    FM-FUNCTION-REF |
    FM-CHARACTERISTIC-REF
  ) *
>

```

Sub elements	FMEA entity attribute for import/export
FM-FUNCTION-REF	Next Lower Level Function Element References.
FM-CHARACTERISTIC-REF	Reference of Inspection Definition. Parent Function of Inspection Definition will be added as Next Lower Level Function Element.

Element Attribute

```

<<!ATTLIST FM-PREREQUISITES
    S                CDATA        #IMPLIED
    SI               CDATA        #IMPLIED
    SYSCOND          CDATA        #IMPLIED
    T                CDATA        #IMPLIED
    VIEW             CDATA        #IMPLIED
>

```

FM- FAULT-REFS{XE"FM-FAULT-REFS"\b}

XML Schema

```

<!ELEMENT FM-FAULT-REFS
  (
    FM-FAULT-REF*
  )
>

```

Sub elements	FMEA entity attribute for import/export
FM-FAULT-REF	Failure Element References.

Element Attribute

```

<!ATTLIST FM-FAULT-REFS
    S                CDATA        #IMPLIED
    SI               CDATA        #IMPLIED
    SYSCOND          CDATA        #IMPLIED
    T                CDATA        #IMPLIED
    VIEW             CDATA        #IMPLIED
>

```

FM- CHARACTERISTICS{XE"FM-CHARACTERISTICS"\b}

XML Schema

```

<!ELEMENT FM-CHARACTERISTICS
  (
    FM-CHARACTERISTICS*
  )

```

```
)
>
```

Sub elements	FMEA entity attribute for import/export
FM-CHARACTERISTIC	Function Element and Inspection Definition.

Element Attribute

```
<!ATTLIST FM-CHARACTERISTICS
    S                CDATA        #IMPLIED
    SI               CDATA        #IMPLIED
    SYSCOND          CDATA        #IMPLIED
    T                CDATA        #IMPLIED
    VIEW            CDATA        #IMPLIED
>
```

FM- CHARACTERISTIC{XE"FM-CHARACTERISTIC"b}

For every Inspection Definition one function element is created.

XML Schema

```
<!ELEMENT FM-CHARACTERISTIC
(
    LONG-NAME ,
    SHORT-NAME? ,
    DESC? ,
    CATEGORY? ,
    ADMIN-DATA? ,
    ANNOTATIONS? ,
    FM-USERDEFINED-ATTRIBUTE-REFS? ,
    FM-LINK-DESTINATIONS? ,
    FM-CHARACTERISTIC-TYPE-REF? ,
    FM-PREREQUISITES? ,
    FM-FAULT-REFS? ,
    FM-CLASSIFICATION? ,
    PRMS? ,
    FM-CONTROL-METHODS? ,
    FM-REACTIONS? ,
    FM-TEST-SAMPLE-REFS? ,
    TEAM-MEMBER-REF? ,
    FIGURE? ,
    SDGS? ,
    MISC-DATA?
)
>
```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	Function Element/ Inspection Definition Name if length is < 128 else Function Element/ Inspection Definition Description.
SHORT-NAME	Function Element/ Inspection Definition Name if LONG-NAME length is > 128
FM-USERDEFINED-ATTRIBUTE-REFS	Function Requirement references. Where FM-USERDEFINED-ATTRIBUTE/ CATEGORY is FunctionRequirement. Item type created as reference.
FM-PREREQUISITES	Next Lower Level Function Element References. In case of Inspection Definition, it's Function Element will be added.
FM-FAULT-REFS	Failure Element references

Element Attribute

```

<!ATTLIST FM-CHARACTERISTIC
  F-ID-CLASS          NMTOKEN      #FIXED    "FM-CHARACTERISTIC "
  ID                  ID              #IMPLIED
  S                   CDATA         #IMPLIED
  SI                  CDATA         #IMPLIED
  SYSCOND             CDATA         #IMPLIED
  T                   CDATA         #IMPLIED
  VIEW                CDATA         #IMPLIED

```

FM- FAULT-TYPES{XE"FM-FAULT-TYPES"\b}

List of Failure Specification.

XML Schema

```

<!ELEMENT FM-FAULT-TYPES
  (
    FM-FAULT-TYPES*
  )

```

Sub elements	FMEA entity attribute for import/export
FM-FAULT-TYPE	Failure Specification.

Element Attribute

```

<!ATTLIST FM-FAULT-TYPES
  S                   CDATA         #IMPLIED
  SI                  CDATA         #IMPLIED
  SYSCOND             CDATA         #IMPLIED
  T                   CDATA         #IMPLIED

```

```

        VIEW                CDATA        #IMPLIED
    >

```

FM- FAULT-TYPE{XE"FM-FAULT-TYPE"\b}

Failure Specification.

XML Schema

```

<!ELEMENT FM-FAULT-TYPE
    (
        LONG-NAME ,
        SHORT-NAME? ,
        DESC? ,
        CATEGORY? ,
        ADMIN-DATA? ,
        ANNOTATIONS? ,
        FM-FAULT-TYPE-DECOMPOSITION? ,
        SDGS? ,
        MISC-DATA?
    )
>

```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	Failure Specification Name if length is < 128 else Failure Specification Description.
SHORT-NAME	Failure Specification Name if LONG-NAME length is > 128

Element Attribute

```

<!ATTLIST FM-FAULT-TYPE
    F-ID-CLASS          NMTOKEN    #FIXED    "FM-FAULT-TYPE"
    ID                  ID          #IMPLIED
    S                   CDATA      #IMPLIED
    SI                  CDATA      #IMPLIED
    SYSCOND             CDATA      #IMPLIED
    T                   CDATA      #IMPLIED
    VIEW                CDATA      #IMPLIED
>

```

FM- FAULTS{XE"FM-FAULTS"\b}

List of Failure Elements.

XML Schema

```

<!ELEMENT FM-FAULTS
    (
        FM-FAULT*
    )
>

```

```
)
>
```

Sub elements	FMEA entity attribute for import/export
FM- FAULTS	Failure Element.

Element Attribute

```
<!ATTLIST FM-FAULTS
    S                CDATA        #IMPLIED
    SI               CDATA        #IMPLIED
    SYSCOND          CDATA        #IMPLIED
    T                CDATA        #IMPLIED
    VIEW             CDATA        #IMPLIED
>
```

FM-FAULT{XE"FM-FAULT"\b}

Failure element.

XML Schema

```
<!ELEMENT FM-FAULT
(
    LONG-NAME ,
    SHORT-NAME? ,
    DESC? ,
    CATEGORY? ,
    ADMIN-DATA? ,
    ANNOTATIONS? ,
    FM-USERDEFINED-ATTRIBUTE-REFS? ,
    FM-LINK-DESTINATIONS? ,
    FM-FAULT-TYPE-REF? ,
    FM-CLASSIFICATION? ,
    FM-CAUSES? ,
    FM-SIGNIFICANCE? ,
    FM-TASK-HISTORY? ,
    SDGS? ,
    MISC-DATA?
)
>
```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	Failure Element Name if length is < 128 else Failure Element Description.
SHORT-NAME	Failure Element Name if LONG-NAME length is > 128
FM-FAULT-TYPE-REF	Failure specification reference.

Sub elements	FMEA entity attribute for import/export
FM-CAUSES	Failure cause reference.
FM-SIGNIFICANCE-REFS	Severity for Failure Causes attached to Failure Mode.
FM-TASK-HISTORY	Current control and optimization actions, If this Failure Element is attached as Failure Cause.

Element Attribute

```

<!ATTLIST FM-FAULT
  F-ID-CLASS          NMTOKEN    #FIXED    "FM-FAULT "
  F-NAMESPACE        NMTOKENS   #FIXED    "FM-TASK-SET FM-TASK-
SETS"
  ID                  ID           #IMPLIED
  S                   CDATA        #IMPLIED
  SI                  CDATA        #IMPLIED
  SYSCOND            CDATA        #IMPLIED
  T                   CDATA        #IMPLIED
  VIEW               CDATA        #IMPLIED
>

```

FM-CAUSES{XE"FM-CAUSES"b}

List of Failure Specification.

XML Schema

```

<!ELEMENT FM-CAUSES
  (
    FM-CAUSES*
  )
>

```

Sub elements	FMEA entity attribute for import/export
FM-FAULT	Failure element.

Element Attribute

```

<!ATTLIST FM-CAUSES
  S                   CDATA        #IMPLIED
  SI                  CDATA        #IMPLIED
  SYSCOND            CDATA        #IMPLIED
  T                   CDATA        #IMPLIED
  VIEW               CDATA        #IMPLIED
>

```

FM-SIGNIFICANCE{XE"FM-SIGNIFICANCE"b}

XML Schema

```

<!ELEMENT FM-SIGNIFICANCE
  (

```

```

RISK-PRIORITY-FACTOR ,
DESC? ,
ANNOTATIONS?
)
>

```

Sub elements	FMEA entity attribute for import/export
RISK-PRIORITY-FACTOR	Severity of Failure Effect.

Element Attribute

```

<!ATTLIST FM-SIGNIFICANCE
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND         CDATA        #IMPLIED
  T               CDATA        #IMPLIED
  VIEW            CDATA        #IMPLIED

```

FM-TASK-HISTORY{XE"FM-TASK-HISTORY"\b}

XML Schema

```

<!ELEMENT FM-TASK-HISTORY
  (
    FM-TASK-SETS*
  )
>

```

Sub elements	FMEA entity attribute for import/export
FM-TASK-SETS	Quality Action reference defined in first Task sets are added as current control Actions and remaining are added as Optimization Actions.

Element Attribute

```

<!ATTLIST FM-TASK-HISTORY
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND         CDATA        #IMPLIED
  T               CDATA        #IMPLIED
  VIEW            CDATA        #IMPLIED

```

FM-TASK-SETS{XE"FM-TASK-SETS"\b}

Quality Action reference defined in first Task sets are added as current control Actions and remaining are added as Optimization Actions.

XML Schema

```

<!ELEMENT FM-TASK-SETS
  (

```

```

LONG-NAME ,
SHORT-NAME? ,
DESC? ,
CATEGORY? ,
ADMIN-DATA? ,
ANNOTATIONS? ,
DATE-1? ,
FM-TASK-SET* ,
SDGS? ,
MISC-DATA?
)
>

```

Sub elements	FMEA entity attribute for import/export
FM-TASK-SET	Quality Action reference defined in Task set are added as prevention and detection Actions.

Element Attribute

```

<!ATTLIST FM-TASK-SETS
  F-ID-CLASS          NMTOKEN   #FIXED   "FM-TASK-SETS "
  ID                  ID          #IMPLIED
  S                   CDATA      #IMPLIED
  SI                  CDATA      #IMPLIED
  SYSCOND             CDATA      #IMPLIED
  T                   CDATA      #IMPLIED
  VIEW                CDATA      #IMPLIED
>

```

FM-TASK-SET{XE"FM-TASK-SET"\b}

XML Schema

```

<!ELEMENT FM-TASK-SET
(
  LONG-NAME ,
  SHORT-NAME? ,
  DESC? ,
  CATEGORY? ,
  ADMIN-DATA? ,
  ANNOTATIONS? ,
  FM-DETECTION-TASKS? ,
  FM-OCCURRENCE-TASKS? ,
  FM-EXTERNAL-ACTIONS-DETECT? ,
  FM-EXTERNAL-ACTIONS-OCCURRENCE? ,
  SDGS? ,
  MISC-DATA?
)
>

```

Sub elements	FMEA entity attribute for import/export
FM-DETECTION-TASKS	Detection Action reference.
FM-OCCURENCE-TASKS	Prevention Action reference.

Element Attribute

```

<!ATTLIST FM-TASK-SET
  F-ID-CLASS          NMTOKEN    #FIXED    "FM-TASK-SET"
  ID                  ID            #IMPLIED
  S                   CDATA       #IMPLIED
  SI                  CDATA       #IMPLIED
  SYSCOND             CDATA       #IMPLIED
  T                   CDATA       #IMPLIED
  VIEW                CDATA       #IMPLIED
>

```

FM-DETECTION-TASKS{XE"FM-DETECTION-TASKS"\b}

Detection action reference.

XML Schema

```

<!ELEMENT FM-DETECTION-TASKS
  (
    RISK-PRIORITY-FACTOR? ,
    FM-ACTION-REF*
  )
>

```

Sub elements	FMEA entity attribute for import/export
RISK-PRIORITY-FACTOR	Detection rating of action.
FM-ACTION-REF	Quality action references.

Element Attribute

```

<!ATTLIST FM-DETECTION-TASKS
  S                   CDATA       #IMPLIED
  SI                  CDATA       #IMPLIED
  SYSCOND             CDATA       #IMPLIED
  T                   CDATA       #IMPLIED
  VIEW                CDATA       #IMPLIED
>

```

FM-OCCURRENCE-TASKS{XE"FM-OCCURRENCE-TASKS"\b}

Prevention action reference.

XML Schema

```

<!ELEMENT FM-OCCURRENCE-TASKS
  (
    RISK-PRIORITY-FACTOR? ,
    FM-ACTION-REF*
  )
>

```

Sub elements	FMEA entity attribute for import/export
RISK-PRIORITY-FACTOR	Occurrence rating of action.
FM-ACTION-REF	Quality action references.

Element Attribute

```

<!ATTLIST FM-OCCURRENCE-TASKS
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND         CDATA        #IMPLIED
  T               CDATA        #IMPLIED
  VIEW            CDATA        #IMPLIED
>

```

FM-ACTIONS{XE"FM-ACTIONS"\b}

List of quality actions.

XML Schema

```

<!ELEMENT FM-ACTIONS
  (
    FM-ACTION*
  )
>

```

Sub elements	FMEA entity attribute for import/export
FM-ACTION	Quality action.

Element Attribute

```

<!ATTLIST FM-ACTIONS
  S                CDATA        #IMPLIED
  SI               CDATA        #IMPLIED
  SYSCOND         CDATA        #IMPLIED
  T               CDATA        #IMPLIED
  VIEW            CDATA        #IMPLIED
>

```

FM-ACTION{XE"FM-ACTION"\b}

XML Schema

```

<!ELEMENT FM-ACTION
  (

```

```

LONG-NAME ,
SHORT-NAME? ,
DESC? ,
CATEGORY? ,
ADMIN-DATA? ,
ANNOTATIONS? ,
FM-USERDEFINED-ATTRIBUTE-REFS? ,
FM-LINK-DESTINATIONS? ,
FM-ACTION-TYPE-REF? ,
TEAM-MEMBER-REF? ,
FM-TASK-SCHEDULE? ,
FM-ACTION-STATE? ,
REMARK? ,
FM-HISTORY-STATES? ,
SDGS? ,
MISC-DATA?
)
>

```

Sub elements	FMEA entity attribute for import/export
LONG-NAME	Quality Action Name if length is < 128 else Quality Action Description.
SHORT-NAME	Quality Action Name if LONG-NAME length is > 128
CATEGORY	Quality Action Subtype.
FM-ACTION-STATE	Quality Action Status.

Element Attribute

```

<!ATTLIST FM-ACTION
  F-ID-CLASS          NMTOKEN   #FIXED   "FM-ACTION"
  ID                  ID           #IMPLIED
  S                   CDATA      #IMPLIED
  SI                  CDATA      #IMPLIED
  SYSCOND             CDATA      #IMPLIED
  T                   CDATA      #IMPLIED
  VIEW                CDATA      #IMPLIED
>

```


7. Configure Quality Actions

Configuring Quality Actions

You can configure quality actions that are created to assign and track the status of all quality-related tasks.

You can do the following:

- **Set up dispatcher for sending reminders and triggering escalation workflows.**
- **View audit logs for modified quality actions.**
- **View the status and reports of all quality actions.**

Set up dispatcher to trigger reminders and escalation workflows for quality actions

Dispatcher settings for triggering reminders and escalation

You can **set up the Teamcenter Dispatcher component to create dispatcher jobs that trigger reminders and escalation for Teamcenter objects.**

To do this, you must first ensure that the **Escalation Management Translator** is installed. Subsequently, the first dispatcher request is generated from **escalation_request** script. The first request is in the **INITIAL** state until the dispatcher component is activated. Once running, the dispatcher starts processing the **escalationmgmt** request.

In addition to installing the translator, you must set the following preferences:

- To trigger the first request, set the **AWC_escalation_mgmt_start_time** preference.
- To specify the time interval after which the next request must be run, set the **AWC_escalation_mgmt_sleep_time** preference.
- To transfer all your profiles and preferences from a legacy version to a newer version, you must set the value of **AWC_Rem0ReadReminderProfilesFromPreference** as **True**. By default, this preference value is **False**, and the profiles are transferred based on the reminder business object.

Start the dispatcher service to run the escalationmgmt translator

Before you start the **escalationmgmt** service, you must start the dispatcher service for ensure that the **escalationmgmt** translator is running.

Procedure

1. Open Teamcenter command prompt.
2. Browse to *DR [Dispatcher Root]* folder under *TC_ROOT*.
3. Start the Scheduler from *TC_ROOT\DR\Scheduler\bin – runscheduler.bat*.
4. Start the Module from *TC_ROOT\DR\Module\bin – runmodule.bat*. Check that the **escalationmgmt** translator is listed in the module prompt.
5. Start the Dispatcher Client from *TC_ROOT\DR\DispatcherClient\bin – runDispatcherClient.bat*.

Start the escalationmgmt service to trigger reminders and escalation workflows for quality actions

You can set up the Teamcenter Dispatcher component to create dispatcher jobs which trigger the reminders and escalation workflows for quality actions.

Prerequisites

The **Escalation Management Translator** feature, available for Teamcenter 13.1 and later releases, is installed and the **escalationmgmt** translator is running.

Procedure

1. Check the status of the dispatcher request.
 - a. Open the Teamcenter Rich Client.
 - b. Click on **My Teamcenter**.
 - c. From the **File** menu select **Translation**.
 - d. Select **Administrator Console – ALL**.

The **Dispatcher Request Administration Console** displays the status of the **escalationmgmt** service.

- e. To view more details related to the dispatcher request, click on the **Properties** tab.
2. Restart the **escalationmgmt** service.
 - a. Open the Teamcenter command prompt.

- b. Navigate to the `TC_ROOT` folder.
- c. Run the following request:

```
escalation_request.exe/sh -u=<user> -p=<password>
```

Note:

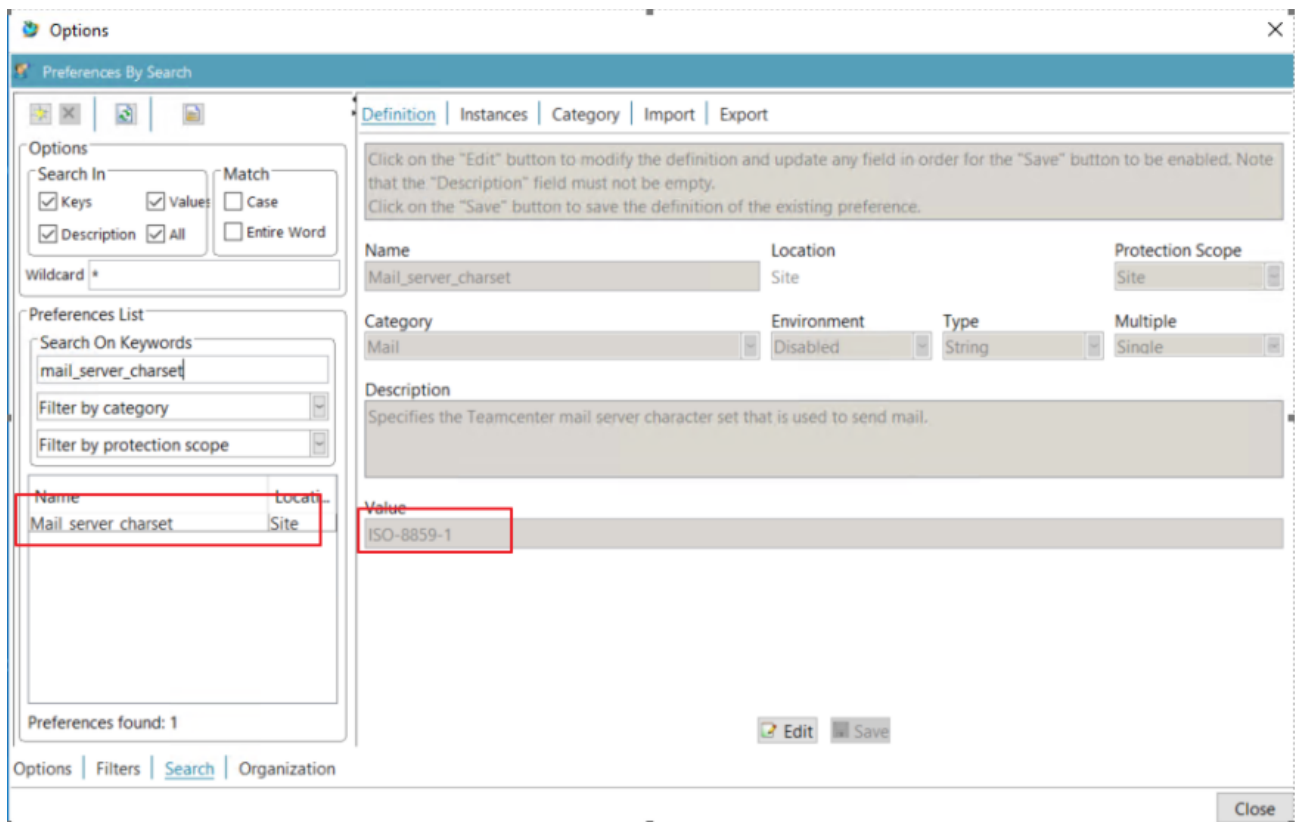
The Dispatcher related logs are available under DR [Dispatcher Root]. The reminder and escalation related logs are available in the `escalation_mgmt.exe_timestamp.syslog` file.

Set up email notifications

To set up the email notifications, you must set the following preferences:

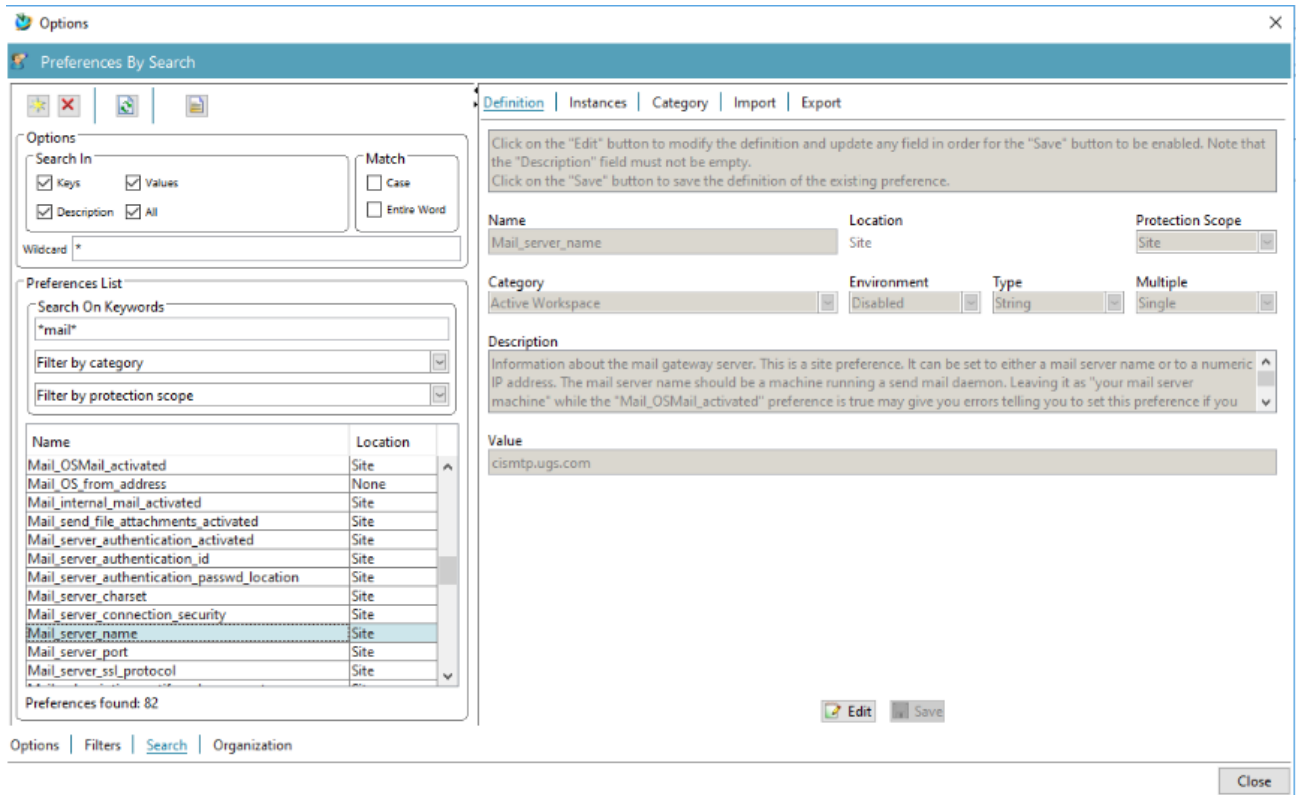
Mail Server Charset

The **Mail Server Charset** preference must be set to value that is defined as the Teamcenter mail sever character set for sending emails. For example, you can set it to **ISO-8859-1**.



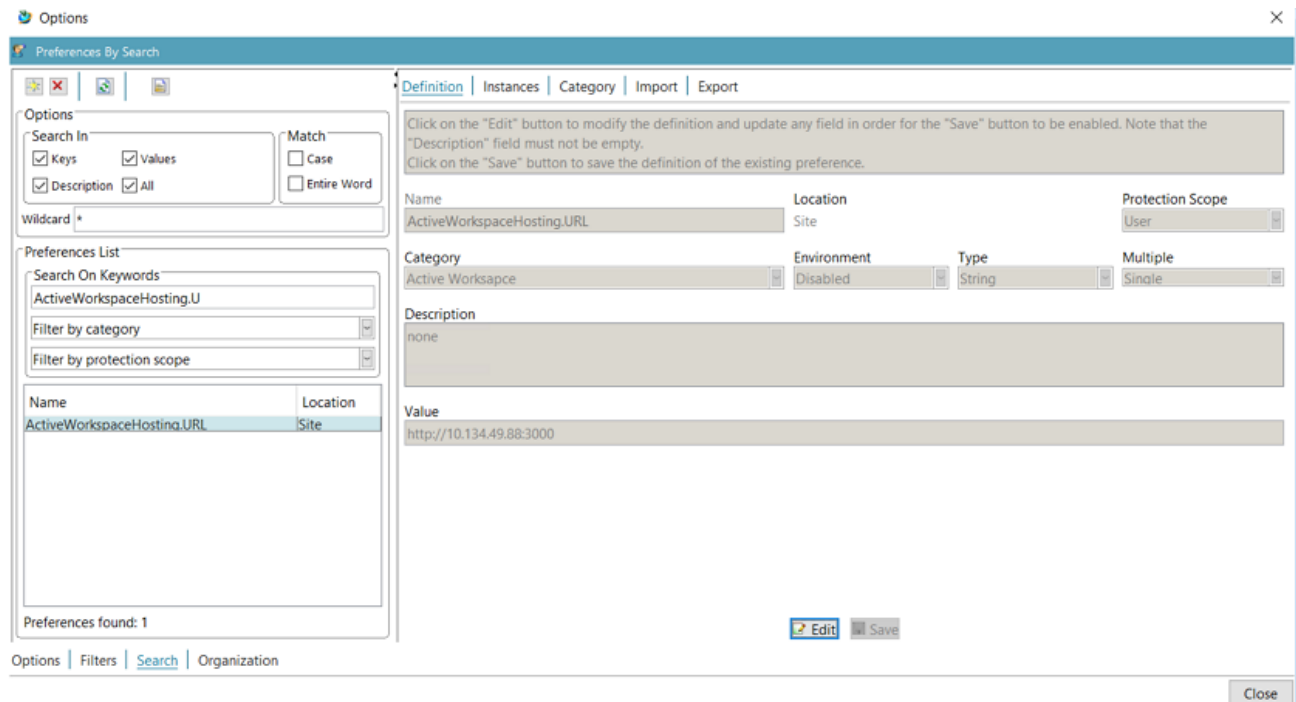
Mail Server Name

The value of the **Mail Server Name** must be set to the name of the gateway server. For example, **cismtp.ugs.com**.



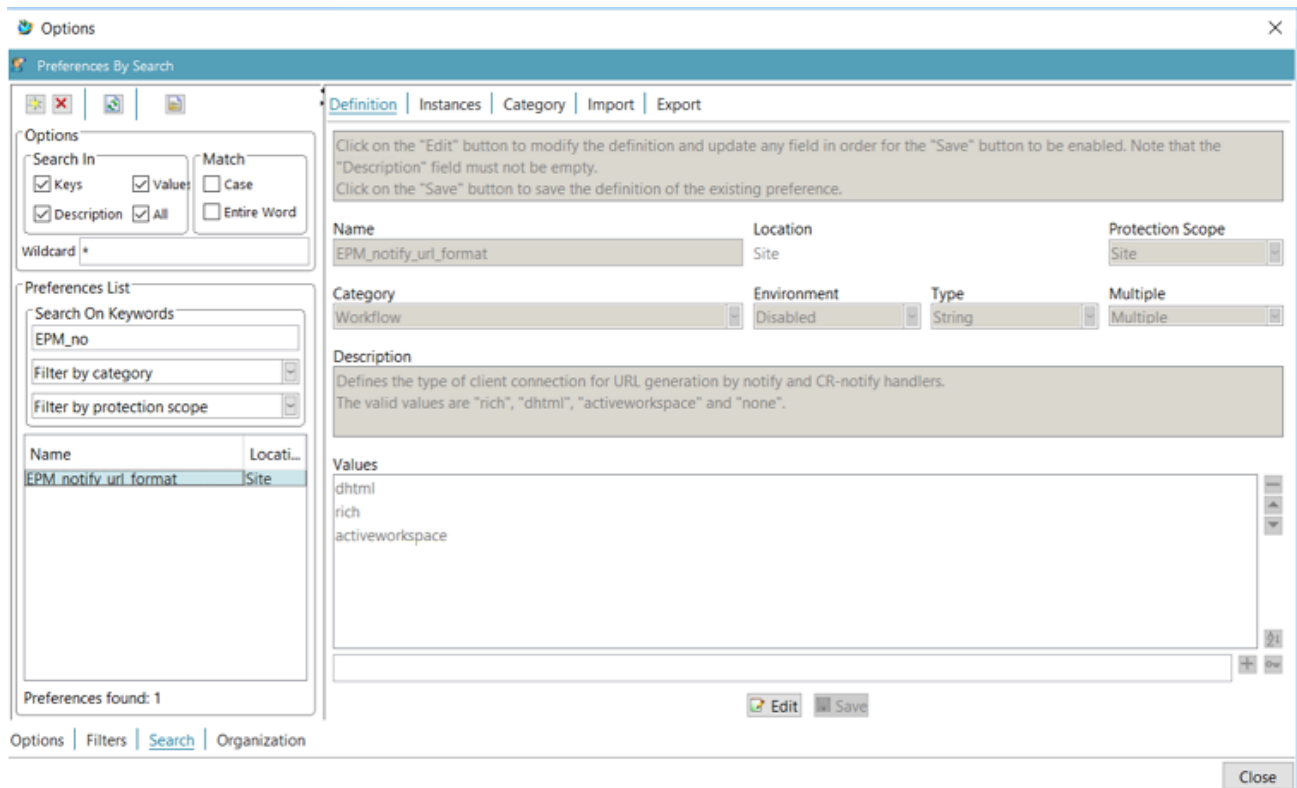
ActiveWorkspaceHosting.URL

The value for **ActiveWorkspaceHosting.URL** preference needs to be set to the Active Workspace URL. For example: `http://localhost:3000`



EPM_notify_url_format

Add a new value **activeworkspace** for the **EPM_notify_url_format** preference. When you set this preference, users will receive the Active Workspace URL in the email.



View audit logs for modified quality actions

A quality manager uses quality actions to create, execute, and track quality-related tasks. It is also used to manage tasks specific to quality control applications such as corrective action, preventive action, or checklist action. After creating quality actions, the quality manager assigns them to the users responsible for them. They complete the quality action by providing a comment.

You can view the audit logs to look for modifications to the status of quality actions.

1. Change the workspace to the **Default** one.
2. Click the **Audit Logs** tab and open a log entry from the **General Logs** table.

The system displays the logged date, the event type name, user ID, group name, and other details.

3. Select an event and click **Information > All Properties** tab.

The system displays the old values and the new values of the fields that are modified.

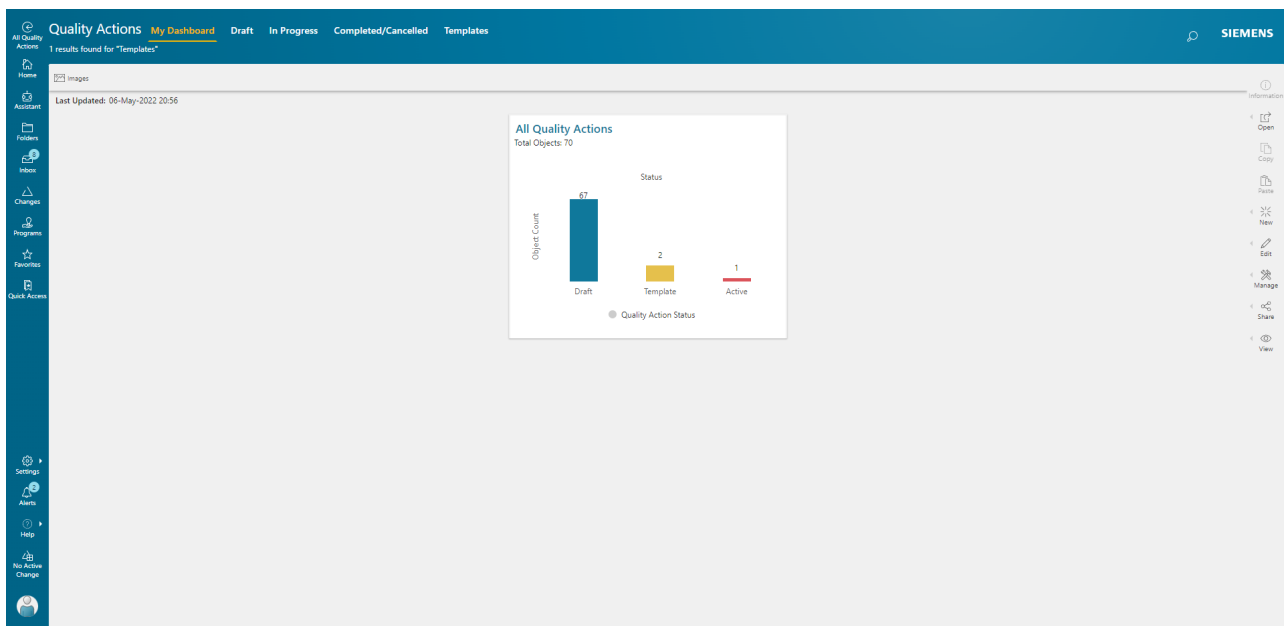
View the status of all quality actions

You can view the all quality actions in the **Quality Management** workspace. In this workspace, you can also view the status of the quality actions by clicking the **QUALITY ACTIONS** tile.

Procedure

1. Change the workspace to **Quality Management**.
2. Click the **QUALITY ACTIONS** tile.

The **Quality Actions** page is displayed.



You can view the reports for all the quality actions and their details in the following tabs according to their status:

- **My Dashboard:** Displays report of all the quality action statuses and the object count.
- **Draft:** Displays all quality actions that are in the draft state.
- **In Progress:** Displays all quality actions that are in the active state.
- **Completed/Canceled:** Displays all quality actions that are completed or canceled.
- **Templates:** Displays all quality actions that are created or used as templates.

Note:

- Each tab filters the results by the quality action type. From the **Chart by** list, select the filters as required.
- A **dba** user, you can modify the reports in the available tabs and see the updated reports.

Set a preference to add external users to a quality action

You can enable or disable adding external users, those who are not Teamcenter users, to a quality action.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. In the **Search** box, type **Qam0EnableExternalUser**.

Note:

By default, the preference value is set to `true`.





3. Do one of the following:
 - To enable adding external users to a quality action, set the preference value to `true`.
 - To disable adding external users to a quality action, set the preference value to `false`.
4. Sign out and log on again to apply changes.

Enable publishing and transferring ownership of actions and reminders to other sites

To share actions and reminders to other site in a Multi-Site Collaboration network, you must enable them for publishing and transferring ownership.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. To enable publishing the quality actions and reminders, do the following:
 - a. In the **Search** box, type **TC_publishable_classes** and select it.

- b. Click **Edit** .
 - c. Specify the following values for the quality objects that you want to publish.
 - To enable a quality actions, specify the value as **Qam0QualityAction**.
 - To enable a reminders, specify the value as **Rem0Reminder**.
 - d. Click **Save Edits** .
3. To enable transferring ownership of the quality actions and reminders, do the following:
- a. In the **Search** box, type **TC_directly_transferable_classes** and select it.
 - b. Click **Edit** .
 - c. Specify the following values for the quality objects that you want to transfer.
 - To enable a quality actions, specify the value as **Qam0QualityAction**.
 - To enable a reminders, specify the value as **Rem0Reminder**.
 - d. Click **Save Edits** .

8. Configure Quality Project Management

Configuring Quality Project Management

Quality Project Management helps quality engineers to avoid risks by creating quality plans upfront and anticipating failure points at each stage of the development phase.

You can do the following:

- **View the context of quality actions in Quality Project Management.**
- **Configure workflows to enable users to work on quality actions.**
- **Customize RYG rating behavior using BMIDE business object constants.**
- **View audit logs for modified RYG ratings.**
- **View the status of all quality projects and checklists.**

View the context for quality actions in Quality Project Management

To enable quality engineers to view the context of the quality actions in Quality Project Management, such as the programs, projects, or events which are associated with it, you must set the **Apm0ShowContextForAction** preference.

1. On the Active Workspace home page, click the **PREFERENCES** tile.
2. In the **Search** box, type **Apm0ShowContext**.
3. Select the **Apm0ShowContextForAction** preference from the search results.

The default preference value is **Prg0AbsProgramPlan**.

4. Click **Edit** to edit the preference value to add the required business objects in the **VALUES** list.

You can also modify the sequence of the objects in the list.

5. Save your changes.

Configure the workflows for quality actions in Quality Project Management

You can configure workflows to enable users to work on quality actions.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. In the **Search** box, type **Qam0QualityActionStatusWorkflow**.
3. Select the **Qam0QualityActionStatusWorkflow** preference from the search results.

The default preference value is **Quality Action Status Change Workflow**.

4. Click **Edit** to edit the preference value to add the required business objects in the **VALUES** box.
 - To initiate a custom workflow, type **Custom workflow** in the **VALUES** box.
 - If you do not want to initiate a workflow, type **Blank** in the **VALUES** box.

Customize RYG rating behavior

About customizing RYG rating behavior in Quality Project Management

The red, yellow, green (RYG) ratings are used at the program, project, subproject, event, checklist, and question level to indicate the status of the various criteria. The default rating behavior is follows:

- If the question is *mandatory*, the system automatically assigns an RYG color to it when it is answered. For example, based on the default rules, if the mandatory question is answered with a *no*, the RYG object automatically becomes red. Further, if there is a quality action associated with it, it becomes yellow. Similarly, if the answer is a *yes*, it becomes green.
- The RYG ratings of the questions are rolled over to the checklist to indicate the status of the checklist. For example, if a checklist has five questions, and all the four questions are answered with a *yes* (green) and one is answered as *no* (red), the RYG object of the checklist becomes red based on the default rules. Similarly, if all the four questions are green, and one has a quality action associated with it (yellow color), the RYG object of the checklist becomes yellow.
- The RYG ratings of the checklists are rolled over to the event to indicate the status of the event. For example, if all the checklists are green or have null values, the RYG object of the event becomes green. The RYG object of the event becomes yellow or red if one of the checklists is yellow or red, respectively. Similarly, a plan level can have multiple events. The RYG ratings of the events are rolled over to the plan level and its status is similar to events.

BMIDE business object constants are used to control RYG ratings. You can edit the following business objects to control the RYG rating behavior.

- The **Psi0ChecklistQuestion** business object contains the **Apm0RYGAssessmentRuleGreen** and **Apm0RYGAssessmentRuleRed** business object constants that control the assessment rule for the

checklist question business object. In addition, it contains the **Apm0RYGThresholdOfYellowGreen** and **Apm0RYGThresoldOfRedYellow** business object constants that determine the RYG rating.

- The **Psi0Checklist**, **Prg0AbsPlan**, and **Prg0AbsEvent** business objects contain the **Apm0RYGThresholdOfYellowGreen** and **Apm0RYGThresoldOfRedYellow** business object constants that determine the RYG rating.

Examples for rating behavior

Scenario 1

Let us assume that there are 10 questions in a checklist of which 5 are green, 3 are yellow, and 2 are red. The values of the business object constants are set as follows:

- **Apm0RYGThresoldOfRedYellow** to 2
- **Apm0RYGThresholdOfYellowGreen** to 3

In such a scenario, the final checklist rating is *Red* because **Apm0RYGThresoldOfRedYellow** is set to 2.

The precedence by which the system checks for conditions to be met is Red, Yellow, and Green.

Scenario 2

Let us assume that there are 10 questions in a checklist of which 8 are green, 1 is yellow, and 1 is red. The values of the business object constants are set as follows:

- **Apm0RYGThresoldOfRedYellow** to 2
- **Apm0RYGThresholdOfYellowGreen** to 3

In such a scenario, the final checklist rating is *Green* because the **Apm0RYGThresoldOfRedYellow** constant for the yellow or red condition is not achieved.

Scenario 3

Let us assume that there are 10 questions in a checklist of which 6 are green, 3 are yellow, and 1 is red. The values of the business object constants are set as follows:

- **Apm0RYGThresoldOfRedYellow** to 2
- **Apm0RYGThresholdOfYellowGreen** to 3

In such a scenario, the final checklist rating is *Yellow* because the **Apm0RYGThresoldOfRedYellow** constant is set to 3.

Customize the RYG rating behavior in Quality Project Management

The red, yellow, green (RYG) ratings are used at the program, project, subproject, event, checklist, and question level to indicate the status of the various criteria. You can customize the RYG rating behavior by using BMIDE business object constants.

1. Open BMIDE with the **apm0apqpprogrammgt** solution template.
2. Search for the **Psi0ChecklistQuestion** business object.
 - a. Click the **Business Objects Constant** tab and edit the **Apm0RYGAssessmentRuleGreen** business object constant. The default value is **Psi0IsMandatory:true,psi0Answer.Yes**. This value means that if the mandatory question is answered, the system automatically assigns the green color to the checklist business object based on default rules.

You can remove the default value, for example, if you do not want the rating behavior to be based on mandatory questions.

- b. Edit the **Apm0RYGAssessmentRuleRed** business object constant. The default value is **Psi0IsMandatory:true,psi0Answer.No**. This value means that if the mandatory question is not answered, the system automatically assigns the red color to the checklist business object based on default rules.

You can remove the default value, for example, if you do not want the rating behavior to be based on mandatory questions.

- c. Select the **Apm0RYGThresholdOfYellowGreen** business object constant, click **Edit**, and change the **Value** as appropriate. The default value is **1**.

Example:

If you specify the value of this constant as **2**, the RYG rating becomes *Red* if *two* RYG values of plan level objects such as such as program, event, checklist, or checklist question are red.

Similarly, if you specify the value of this constant as **3**, the RYG rating becomes *Red* if *three* RYG values of plan level objects such as such as program, event, checklist, or checklist question are red.

Caution:

Do not specify a negative value for this business object constant.

- d. Select the **Apm0RYGThresoldOfRedYellow** business object constant, click **Edit**, and change the **Value** as appropriate. The default value is **1**.

Example:

If you specify the value of this constant as **2**, the RYG rating becomes *Yellow* if two RYG values of plan level objects such as such as program, event, checklist, or checklist question are yellow.

Similarly, if you specify the value of this constant as **3**, the RYG rating becomes *Yellow* if *three* RYG values of plan level objects such as such as program, event, checklist, or checklist question are yellow.

3. Search for the **Psi0Checklist**, **Prg0AbsPlan**, and **Prg0AbsEvent** business objects and open them in separate tabs to edit them. Click the **Business Objects Constant** tab and repeat **step c** and **step d**.
4. Create a custom template with the changes in BMIDE.
5. Deploy the template with the changes.

For more information about deploying templates in BMIDE, see [Deploy a template](#).

View audit logs for modified RYG ratings

By default, the RYG ratings of the checklists are rolled over to the event, and from the event, to the plan level. However, there are some exceptions. The system evaluates only mandatory questions based on the default rules. If the question is not mandatory, the user responsible can set the RYG rating manually by setting the **Override Automatic Rollup** check box to **true**. Similarly, if an event is not significant to the overall project or program, the quality manager can manually edit the ratings.

You can view the audit logs to look for such modified RYG ratings.

1. Change the workspace to the **Default** one.
2. Click the **Audit Logs** tab and open a log entry from the **General Logs** table.

The system displays the logged date, the event type name, user ID, group name, and other details.

3. Click the **All Properties** tab.

The system displays the old values, and new values of the fields that are modified.

Example:

Rating: Green

Rating Old Value: Yellow

View the status of all quality projects and checklists

You can view the quality projects in the **Quality Management** workspace. In here, you can view the status of the quality checklists of that project by clicking the **QUALITY PROJECTS** tile.

Procedure

1. Change the workspace to **Quality Management**.
2. Click the **QUALITY PROJECTS** tile.

The **Quality Projects** page is displayed.


You can see reports of all the quality projects and their details in the following tabs according to in their status:




- **Not Started:** Displays quality checklists that are in the draft state and yet to be started.
- **In Progress:** Displays all quality checklists that are in the active state.
- **Complete:** Displays all quality checklists that are completed.
- **Closed:** Displays all quality checklists that are closed.
- **My Dashboard:** Displays a configured report of all the status and object count of the quality checklists.

Enable publishing and transferring ownership of checklists and checklist questions to other sites

To share checklists and checklist questions to other site in a Multi-Site Collaboration network, you must enable them for publishing and transferring ownership.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. To enable publishing the quality checklist, program checklist, or checklist questions, do the following:
 - a. In the **Search** box, type **TC_publishable_classes** and select it.
 - b. Click **Edit** .
 - c. Specify the following values for the quality objects that you want to publish.

- To enable a quality checklist, specify the value as **Apm0QualityChecklist**.
 - To enable a program checklist, specify the value as **Psi0Checklist**.
 - To enable a program checklist question, specify the value as **Psi0ChecklistQuestion**.
- d. Click **Save Edits** .
3. To enable transferring ownership of the quality checklist, program checklist, or checklist questions, do the following:
- a. In the **Search** box, type **TC_directly_transferable_classes** and select it.
- b. Click **Edit** .
- c. Specify the following values for the quality objects that you want to transfer.
- To enable a quality checklist, specify the value as **Apm0QualityChecklist**.
 - To enable a program checklist, specify the value as **Psi0Checklist**.
 - To enable a program checklist question, specify the value as **Psi0ChecklistQuestion**.
- d. Click **Save Edits** .

9. Configure Quality Audit

Specify the answer and the rating of a checklist based on the finding guideline

You can set the **Qa0QualityAudit_FindingGuideline_Based_Rating** preference to specify the answer corresponding to the finding guideline such that the rating is automatically calculated and rolled over to the checklist.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. In the **Search** box, type **Qa0QualityAudit_FindingGuideline_Based_Rating**.
3. Specify the value of preference in the following format:

```
FindingGuidelineName:ChecklistSpecificationName:AnswerText
```

FindingGuidelineName - The name of the finding guideline for which you are configuring the answer rating.

ChecklistSpecificationName - The name of the checklist specification that contains the question for which you are configuring the answer.

AnswerText - The text that specifies the answer of the question in the checklist specification.

For example:

If you set the **Qa0QualityAudit_FindingGuideline_Based_Rating** preference value as **F-Major:Checklist1:No**, when you specify the finding guideline as *F-Major*, the questions in the *Checklist1* are automatically answered as *No* and the corresponding rating of the answer is rolled over to the checklist.

Set a preference for co-auditor's qualification

Specify if a co-auditor is certified or not by using the **Qa0QualityAudit_RequiredQualifiedCoAuditor** preference.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. In the **Search** box, type **Qa0QualityAudit_RequiredQualifiedCoAuditor**.

Note:

By default, the preference value is set to `true`.

3. Do one of the following:
 - To enable adding certified co-auditors, set the preference value to `true`.
 - To disable adding certified co-auditors, set the preference value to `false`.
4. To apply the changes, sign out and log on again.

View audit logs for modified quality audit objects

As a **dba** user, you can view the audit logs to look for modifications to the quality audit objects.

1. Change the workspace to the **Default** one.
2. Navigate to the quality audit object and click the **History** tab.

The **QUALITY AUDIT HISTORY** section displays the audit log of the selected object.

Enable user access to the audits created from audit templates

As an administrator, you must update the user information in the *audit_mgmt_trigger.bat* file located within the Audit and Assessment Translator. This update enables the user to access or modify audits created from the audit templates. If you want to add multiple users to modify these audits, you must set required users in the Access Control List (ACL).

Procedure

1. Open the *TC_ROOT\DR\Module\Translators\auditmgmt* folder.
2. Open the *audit_mgmt_trigger.bat* file.

```
audit_mgmt_trigger.bat
35 echo      Param 3) is the output file
36 echo      Example:-
37 echo      audit_mgmt_trigger.bat inputfile.txt
38 endlcal
39 goto :EOF
40
41
42 :setenv
43
44 REM TC_ROOT
45 set TC_ROOT=C:\Apps\tc\tc2312\TR
46 REM TC_DATA
47 set TC_DATA=C:\Apps\tc\tc2312\TD
48
49 REM EMT_USER
50 set EMT_USER=infodba
51 REM PWD_FILE
52 set PWD_FILE=C:\Apps\tc\tc2312\TR\security\config1_infodba.pwf
53
54 if "%TC_ROOT%"=="CHANGE_ME" (
55     echo set TC_ROOT to point to TC root
56     goto end
57 )
58
```

3. Set the required user name and password.

```

audit_mgmt_trigger.bat
35 echo    Param 3) is the output file
36 echo    Example:-
37 echo    audit_mgmt_trigger.bat inputfile.txt
38 endlcal
39 goto :EOF
40
41
42 :setenv
43
44 REM TC_ROOT
45 set TC_ROOT=C:\Apps\tc\tc2312\TR
46 REM TC_DATA
47 set TC_DATA=C:\Apps\tc\tc2312\TD
48
49 REM EMT_USER
50 set EMT_USER=<user>
51 REM PWD_FILE
52 set PWD_FILE=<pwd-file of the user>
53
54 if "%TC_ROOT%"=="CHANGE_ME" (
55     echo set TC_ROOT to point to TC root
56     goto end
57 )
58


```




The updated user can now access the audits create from audit templates.

Enable publishing and transferring ownership of audits and audit plans to other sites

To share audits and audit plans to other sites in a Multi-Site Collaboration network, you must enable them for publishing and transferring ownership.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. To enable publishing the quality audits and audit plans, do the following:
 - a. In the **Search** box, type **TC_publishable_classes** and select it.
 - b. Click **Edit** .
 - c. Specify the following values for the quality objects that you want to publish.
 - To enable a quality audit, specify the value as **Qa0QualityAudit**.

- To enable an audit plan, specify the value as **Qa0QualityAuditPlan**.
- d. Click **Save Edits** .
3. To enable transferring ownership of the quality audit and audit plan, do the following:
 - a. In the **Search** box, type **TC_directly_transferable_classes** and select it.
 - b. Click **Edit** .
 - c. Specify the following values for the quality objects that you want to transfer.
 - To enable a quality audit, specify the value as **Qa0QualityAudit**.
 - To enable an audit plan, specify the value as **Qa0QualityAuditPlan**.
 - d. Click **Save Edits** .

10. Configure Problem Solving and Quality Issue

Disable the Due Date verification rule for associated quality actions and checklists

By default, when quality engineers assign a **Due Date** to the associated quality actions and checklists, they must ensure that the date is not later than the due date of the parent Problem Solving process or issue. If your organization does not implement this verification rule, you can disable it for Problem Solving processes or issues.

Procedure

1. On the home page, click the **PREFERENCES** tile.
2. Search for the **CPA0_enable_quality_action_checklist_due_date_control** preference.
3. Set this preference to **False**.

Configure the fonts used in reports

By default, reports are generated by using the fonts that are automatically detected on the host computer where Problem Solving and Quality Issues is installed. If you want to use specific fonts, you must ensure that the fonts are available on this computer, and specify those fonts in the *pdfFop.xconf* file located at *TC_ROOT\bin\reportpdf*.

Procedure

1. In the *TC_ROOT\bin\reportpdf* directory, open the *pdfFop.xconf* file.
2. Navigate to the `` element of this file.

By default, this file has the following entries for commonly used fonts, and the entries are commented out:

```
<!-- <font kerning="yes" embed-url="file:///C:/Windows/Fonts/
segoeui.ttf" embedding-mode="subset"> -->
<!-- <font-triplet name="segoeui" style="normal" weight="normal"/> --
>
<!-- </font> -->
<!-- <font kerning="yes" embed-url="file:///C:/Windows/Fonts/
segoeuib.ttf" embedding-mode="subset"> -->
<!-- <font-triplet name="segoeuib" style="normal" weight="bold"/> -->
```

```

<!-- </font> -->
<!-- <font kerning="yes" embed-url="file:///
C:\Windows\Fonts\YuGothM.ttc" embedding-mode="subset"> -->
<!-- <font-triplet name="YuGothM" style="normal" weight="normal"/> --
>
<!-- </font> -->
<!-- <font kerning="yes" embed-url="file:///
C:\Windows\Fonts\YuGothB.ttc" embedding-mode="subset"> -->
<!-- <font-triplet name="YuGothB" style="normal" weight="bold"/> -->
<!-- </font> -->
<!-- <font kerning="yes" embed-url="file:///C:/Windows/Fonts/
malgunbd.ttf" embedding-mode="subset"> -->
<!-- <font-triplet name="malgunbd" style="normal" weight="bold"/> -->
<!-- </font> -->
<!-- <font kerning="yes" embed-url="file:///C:/Windows/Fonts/
malgun.ttf" embedding-mode="subset"> -->
<!-- <font-triplet name="malgun" style="normal" weight="normal"/> -->

```

3. If any of these fonts are available on the computer, and you want to use them in your report, uncomment the entry for the specific font.
4. If additional fonts are available on the computer, and you want to use them, ensure that you add an entry for the font in this format in the `` element:

```

<font kerning="yes" embed-url="file:<Path where the font file is located>\<file
name of the font>" embedding-mode="subset">
<!-- <font-triplet name="<file name of the font>" style="normal"
weight="normal"/> -->
<!-- </font> -->

```

Display your company logo in Problem Solving and issue reports

You can customize the Problem Solving and issue reports to display your company logo.

1. On the home page, click the **PREFERENCES** tile.
2. Search for the **ProblemSolvingReportCompanyLogo** preference.

The search results display the preferences used to display the logo in the Problem Solving and issue reports. The value of the preferences is set to the name of the dataset that contains the report logo. By default, the **ProblemSolvingReportCompanyLogo** dataset contains the report logo.

Preference name	Type of report
C2CapaRevision_report_icon	Problem Solving report
C2IssueComplaintRevision_report_icon	Customer Complaint Issue report
C2IssueNonConfRevision_report_icon	Non-Conformance Issue report

Preference name	Type of report
C2IssueRevision_report_icon	Quality Issue report
CAW05Why_report_icon	5Why Root Cause Analysis report
CAW0Defect_report_icon	Root Cause Analysis report
CAW0Ishikawa_report_icon	Ishikawa Root Cause Analysis report

3. Perform an advanced search for the **ProblemSolvingReportCompanyLogo** dataset as follows:
 - a. Click **Advanced Search** from the **Search** box or from the **ADVANCED SEARCH** tile to open **Advanced Search**.
 - b. Select **Dataset** as the type of search.
 - c. In the **Name** box, type **ProblemSolvingReportCompanyLogo**.
 - d. From the **Dataset Type** list, select **Image**.
 - e. Clear the values in the **Owning User** list, and click **Search**.

Advanced Search 📄 Undock ✕ Close

Dataset... ▾

Preferred:

Clear All ^

Name:

ProblemSolvingReportCompanyLogo

Dataset ID:

Revision:

Dataset Type:

Image ▾

Description:

Owning User:

Search v

4. Choose **More Commands** ... > **Edit**  > **Replace File** .

5. In the **Replace** panel, click **Select File** to browse to and select the new logo file, and click **Replace**.

When a quality engineer generates a report, the new logo appears in the report.

Note:

The new logo file must be in the Portable Network Graphic (PNG) format.

Configure the duration assigned to the Proposed Due Date for a specific priority

You can define a standard operating procedure where a Problem Solving process must be completed within a specific duration. This duration is based on the **Priority** of the Problem Solving process or issue.

Depending on the **Priority** of the Problem Solving process or issue, the quality engineers must set the **Proposed Due Date** as follows:

Priority	Duration assigned to Proposed Due Date
High	Proposed Due Date must be set to 30 days and earlier from the Creation Date
Medium	Proposed Due Date must be set to 45 days and earlier from the Creation Date
Low	Proposed Due Date must be set to 60 days and earlier from the Creation Date

Note:

You can also choose to disable assigning a duration to the **Proposed Due Date** for a specific priority.

Procedure

1. On the home page, click the **PREFERENCES** tile.
2. To disable assigning a duration to the **Proposed Due Date** for a specific priority, do the following:
 - a. Search for the **CPA0_enable_due_date_using_priority** preference.
 - b. Set this preference to **False**.
3. To modify the duration assigned to the Proposed Due Date for a specific priority, do the following:
 - a. Search for the **CPA0_priorities_duration_limits** preference.
 - b. Edit this preference to the required durations for the associated priorities.

Create the Problem Solving checklist master data

A checklist consists of a list of questions that define the acceptance criteria for the various stages of a Problem Solving process. It is used to evaluate the completion of a Problem Solving process. Each Problem Solving process can have one or more checklists associated to it, and each checklist can have multiple chapters, questions, and subquestions. Multiple questions and subquestions ensure that none of the acceptance criteria is inadvertently overlooked. Further, the questions are designed in such a way that they can be answered with a simple *yes* or *no*.

You, as the Quality Manager, must create and release the checklists in the **Quality Master Data** library. The Quality Management Representative (QMR) assigns the checklists to the required Problem Solving process. The QMR can assign multiple checklists to a Problem Solving process. Depending on the selected **Sub-area** of the checklist, the chapter and associated questions and subquestions are populated in the stage of the Problem Solving process. For example, if you have selected **Containment Actions** as the **Sub-area** of a chapter, then the chapter and its associated questions and subquestions are populated in the **Containment Actions** tab of the Problem Solving process.

1. On the **Home** page, click the **QUALITY MASTER DATA** tile.
2. Click the **Checklists** tab.
3. Click **New** ✨ > **Add**.
4. In the **Add** panel, specify a value for the **Checklist ID**, and enter a **Name** for the checklist.

You can optionally add a description if required.

5. From the **Checklist Type**, select the **Checklist**.
6. From the **Checklist Area**, select the **PSP**.

Warning:

Do not specify the **Sub-area** for **Checklist**. You must specify the **Area** and **Sub-area** only for chapters, questions, and subquestions to populate them in the respective stage of the Problem Solving process.

7. Select a **Calculation Method** and specify the checklist number, if required.
8. In the **Settings** section, select the **Assessment Required** and **Mandatory** check boxes if required.
9. Click **Add** to create the checklist in the **Quality Master Data** library.
10. To add a **Chapter**, **Question**, or **Sub-Question** to the checklist, do the following:
 - a. Select the checklist and click **New** ✨ > **Add**.

- b. In the **Add** panel, specify the **Checklist ID** and **Name**.
- c. From the **Checklist Type** list, select **Chapter**, **Question**, or **Sub-Question**.

In the tree view, you can also drag and drop checklists, chapters, questions, or sub-questions from one element to other.

- d. From the **Checklist Area** list, select **PSP**.
- e. From the **Sub-area** list, select the Problem Solving process stage where you want to populate the chapter, question, or sub-question.
- f. Select a **Calculation Method** and specify the checklist number, if required.
- g. In the **Settings** section, select the **Assessment Required** and **Mandatory** check boxes if required.
- h. Click **Add**.

The **Chapter**, **Question**, or **Sub-Question** are included in the checklist. After you release the checklist, the chapters, questions, or sub-questions are populated in the respective stages of the Problem Solving process when the checklist is added to the Problem Solving process.

Update the transfer mode to use the custom objects when exporting objects

When you are exporting a Problem Solving process or a Quality Issue, you must select the appropriate transfer mode during the export. If you are using a custom object for Problem Solving or Quality Issues, you must update the existing transfer mode to use the custom object. For more information about performing this update, see *Transfer mode objects* in the Teamcenter documentation.

Configure reminders and trigger escalation workflows for Problem Solving

You can configure Problem Solving such that the responsible users get reminders prior to the due date. Additionally, you can also initiate escalation workflows for Problem Solving processes that are not in the required state by the due date.

1. On the home page, click the **PREFERENCES** tile.
2. Search for the **Qam0ReminderAndEscalationProfiles** preference, select it from the search results, and click **Edit**.

The format of the preference value is as follows:

```
BusinessObjectName:DateProperty:Property1=Value1 |
Property2=value1,value2 |
Property3=value1:NoOfDayPrior:Notifier1,Notifier2:workflow-
workflowName:email-NotificationWorkflowName;
```

Parameter	Description
BusinessObjectName	Specify the value of the business object for which you are setting up the notification trigger.
DateProperty	Specify the date property of the business object for which you are setting up the notification trigger.
Property1=Value1, Property2=value1,value2 Property3=value1	Specify the properties and their values will be validated before triggering the notification.
NoOfDaysPrior	Specify the number of days before or after which you want to send the notification. <ul style="list-style-type: none"> Specify a positive value to send a notification prior to the due date. Specify a negative value to send a notification after the due date. Specify 0 to send a notification on the due date.
Notifier1,Notifier2	Specify the users to whom you want to send the notification. Acceptable values are Owner, Responsible User, Last Modified User.
workflow-workflowName	Specify the name of the escalation workflow that you want to trigger to send reminders to the notifier.
email-NotificationWorkflowName	Specify the name of the email notification workflow that you want to trigger to send emails to the notifier.

- In the **Value** box, specify a new preference value.

You can copy the sample preference value and modify the parameters as required. The values for **DateProperty**, **BusinessObjectName**, and **NoOfDaysPrior** are mandatory. The default OOTB system interval is 24 hours. However, you can set the frequency of the reminders or their interval depending on the business requirement.

For the **NoOfDaysPrior** parameter, use the following to define when the reminder and escalation occurs:

Value of the NoOfDaysPrior parameter	Results in
0	Sends the reminder on the same day.
5	Sends the reminder five days prior to the due date.
-5	Triggers an escalation workflow for objects that are not completed five days past the due date.

Example:

- To configure the system such that the owner, all team members, and Problem Solving process owner get a reminder five days prior to the due date for an open Problem Solving process, for which a workflow should also be triggered, you can set the preference as follows:

```
C2CapaRevision:c2_Due_date:CMClosure=Open:5:Owner:workflow-PSP
Reminder Workflow;
```

- To configure the system such that the owner, all team members, Problem Solving process owner, and Champions get a reminder five days prior to the due date for an open Problem Solving process, for which a workflow should also be triggered, you can set the preference as follows:

```
C2CapaRevision:c2_Due_date:CMClosure=Open:5:Owner:workflow-PSP
Reminder Notify Champion Workflow;
```

- To configure the system such that it triggers an escalation workflow and an email notification for open Problem Solving processes that are not completed five days past the due date, you can set the preference as follows:

```
C2CapaRevision:c2_Due_date:CMClosure=Open:-5:Owner:workflow-
PSP Escalation Mgmt Email Notification;
```

- To configure the system such that the owner, issue owner, and issue approver get a reminder five days prior to the due date for an open issue, for which a workflow should also be triggered, you can set the preference as follows:

```
C2IssueRevision:c2_ProposedDueDate:CMClosure=Open:5:Owner:workf
low-Quality Issue Reminder Workflow;
```

- To configure the system such that it triggers an escalation workflow and an email notification for open issues that are not completed five days past the due date, you can set the preference as follows:

```
C2IssueRevision:c2_ProposedDueDate:CMClosure=Open:-1:Owner:work
flow-Quality Issue Escalation Mgmt Email Notification;
```

- Click **Save** to save your changes.

Migrate the root cause defects linked to corrective actions

After you upgrade to Active Workspace 6.3 or later, run the **Migrate Linked Root Cause** utility to migrate the existing root cause defects linked to corrective actions of all Problem Solving processes. Now, quality engineers can view the existing root causes in the **Corrective Actions** tab. If you do not run this utility, you cannot view the existing root causes.

Run the **Migrate Linked Root Cause** utility with the following arguments

```
caw0_migrate_linked_root_cause [-u=user-id] [-p=password] [-g=group] [-m=dry | full]
```

-u

Specifies the user ID.

This is generally a user with administration privileges.

-p

Specifies the password.

-g

Specifies the group associated with the user with administration privileges.

If used without a value, the user's default group is considered.

-m

Specifies the mode in which the utility is run.

Valid mode options are **dry** run or **full** run. In the **dry** mode, the data is not migrated. A summary log file is created with information about all the Problem Solving processes with impacted quality action relations and linked root cause defects. In the **full** mode, all linked root cause defects are migrated.

-h

Displays the help for this utility.

Environment

As specified in *Manually configure the Teamcenter environment* in *Teamcenter Utilities* in Teamcenter help.

Files

The summary log files are generated in the path specified in the **TC_TMP_DIR** environment variable. If you do not set the **TC_TMP_DIR** variable, then the system generates the summary log files in the *C:\temp* directory for Windows or */tmp* directory for Linux systems.

Restrictions

Requires Teamcenter administrator privileges.

Configure the Confidentiality Level of a Problem Solving process

Companies use their company-specific values to indicate the **Confidentiality Level** of a Problem Solving process. By default, the **Confidentiality Level** list in the **Overview** tab does not have any values. You must populate the values that match your business requirements. Sample XML files and the accompanying localization XML files are provided for you in the *TC_ROOT\bmide\client\samples\externallymanagedlovs* directory. You can update the values in the files, and run the **bmide_manage_batch_lovs** utility to populate the values in the database.

To configure the Confidentiality Level of a Problem Solving process, do the following:

1. Navigate to the *TC_ROOT\bmide\client\samples\externallymanagedlovs* directory, and update the company-specific values for **Confidentiality Level** in the *lov_values_add.xml* file.

```
<TcBusinessData xmlns="http://teamcenter.com/BusinessModel/
TcBusinessData" batchXSDVersion="1.0" Date="" >
  <Change>
    <TcLOV description="This list indicates
the confidentiality levels of Problem Solving."
lovType="ListOfValuesString" name="CPA0ConfidentialityLevelLOV"
usage="Exhaustive" isManagedExternally="true" >
      <TcLOVValue conditionName="isTrue" description="Confidential
description" value="Confidential"/>
      <TcLOVValue conditionName="isTrue" description="Public
description" value="Public"/>
      <TcLOVValue conditionName="isTrue" description="Restricted
description" value="Restricted"/>
      <TcLOVValue conditionName="isTrue" description="Highly
Confidential description" value="Highly Confidential"/>
    </TcLOV>
  </Change>
</TcBusinessData>
```

2. In the accompanying localization XML file, update the locale-specific values.

```
<TcBusinessDataLocalization xmlns="http://teamcenter.com/
BusinessModel/TcBusinessDataLocalization" batchXSDVersion="1.0"
```

```

Date="" >
  <Add>
    <key locale="en_US"
id="LOVValue{:}:CPA0ConfidentialityLevelLOV{:}:Confidential"
status="Approved" >German Confidential</key>
    <key locale="en_US"
id="LOVValueDescription{:}:CPA0ConfidentialityLevelLOV{:}:Confidential"
status="Approved" >German Confidential Description</key>
    <key locale="en_US"
id="LOVValue{:}:CPA0ConfidentialityLevelLOV{:}:Public"
status="Approved" >German Public</key>
    <key locale="en_US"
id="LOVValueDescription{:}:CPA0ConfidentialityLevelLOV{:}:Public"
status="Approved" >German Public Description</key>
    <key locale="en_US"
id="LOVValue{:}:CPA0ConfidentialityLevelLOV{:}:Restricted"
status="Approved" >German Restricted</key>
    <key locale="en_US"
id="LOVValueDescription{:}:CPA0ConfidentialityLevelLOV{:}:Restricted"
status="Approved" >German Restricted Description</key>
    <key locale="en_US"
id="LOVValue{:}:CPA0ConfidentialityLevelLOV{:}:Highly Confidential"
status="Approved" >German Highly Confidential</key>
    <key locale="en_US"
id="LOVValueDescription{:}:CPA0ConfidentialityLevelLOV{:}:Highly Confidential"
status="Approved" >German Highly Confidential
Description</key>
  </Add>
</TcBusinessDataLocalization>

```

3. Run the **bmide_manage_batch_lovs** utility with the following arguments

```
bmide_manage_batch_lovs [-u=user-id] [-p=password] [-g=group] [-option=update] [-file=lov_values_add.xml]
```

-u

Specifies the user ID.

This is generally a user with administration privileges.

-p

Specifies the password.

-g

Specifies the group associated with the user with administration privileges.

If used without a value, the user's default group is considered.

-option=update

Updates the values for all the externally managed LOVs defined in the input file.

-file

Specifies the path and the file containing LOV values.

-h

Displays the help for this utility.

After running the utility, the values are populated in the **Confidentiality Level** list in the **Overview** tab.

Disable the risk calculation of a Problem Solving process

By default, you can use Teamcenter to calculate the risk of a Problem Solving process. The risk of a Problem Solving process is calculated as the product of **Severity**, **Probability**, and **Detectability**. If your organization does not use risk calculation, then set the **CPA0_enable_risk_calculation** preference to **False**. This hides these fields in the **Risk Evaluation** section of the **Overview** tab. After you set this preference, the quality engineers can specify only the **Priority** for the Problem Solving process.

▼ RISK EVALUATION	
Severity:	10
Probability:	10
Detectability:	10
Risk:	1000
Priority:	Medium

Define the type of contacts and locations that can be assigned to a Problem Solving process

In the **Plan** tab of a Problem Solving process, you can view contacts and locations from either the vendor added in the **Vendor** section or the partner contract added in the **Partner Contract** section. To decide which type of contacts and locations can be assigned to a Problem Solving process, you can configure this in the **Caw0PlanSublocationHierarchy** preference. Specify the object and its parent object in the following format:

```
object_name , parent_name
```

Example:

To only display contacts and locations of the vendors added in the **Vendor** section, add the following as values in the **Caw0PlanSublocationHierarchy** preference:

```
CompanyContact, Vendor
CompanyLocation, Vendor
```

▼ DEFINITION

Name:	Caw0PlanSublocationHierarchy
Product Area:	Active Workspace
Description:	Caw0PlanSublocationHierarchy registers parent object for an object in Plan Sublocation. The valid values are in the format object_name,parent_name.
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	Yes

▼ VALUES

Values:

```
Ret0Brand, Vendor
CompanyContact, Vendor
CompanyLocation, Vendor
Vm0PrtnrContractRevision, Vendor
Aps0PlantLocation, Factory
An2CompanyContact, Factory
```

Configure which pages appear in the Quality Management workspace header

The **Quality Management** workspace is a workspace dedicated to working with issues and Problem Solving processes. For more information about workspaces, see *Configuration and Extensibility* in the Teamcenter documentation.

In this workspace, when you open the **PROBLEM SOLVING PROCESS** tile or the **QUALITY ISSUE MANAGEMENT** tile, by default, the following pages are available in the header:

- **New**

- In Progress
- Overdue
- Closed

Problem Solvings New In Progress Overdue Closed My Dashboard

You can customize which pages appear in this header as follows:

1. In the Query Builder application, create the required query for the page to be displayed in the header.

Example:

You want to add a page that displays all canceled Problem Solving processes. In Query Builder, create a query to retrieve canceled Problem Solving processes.

2. For the **PROBLEM SOLVING PROCESS** tile, edit the **PSPSublocationQuery** preference to display the new header in the following format:

`<name_of_page> , <query_for_page>`

Example:

Add the following to the **PSPSublocationQuery** preference:

`Canceled , ShowPSPCanceled`

By default, the **PSPSublocationQuery** preference has the following values:

- **New,ShowPSPNew**
- **Overdue,ShowPSPOverdue**
- **InProgress,ShowPSPInProgress**
- **Closed,ShowPSPClosed**

▼ VALUES

New,ShowPSPNew

Overdue,ShowPSPOverdue

InProgress,ShowPSPInProgress

Closed,ShowPSPClosed

- For the **QUALITY ISSUE MANAGEMENT** tile, edit the **QIMSublocationQuery** preference to display the new header in the following format:

`<name_of_page> , <query_for_page>`

- Use configuration builder to manage routes and create a sublocation.for the new header. For more information, see *Configuration and Extensibility* in the Teamcenter documentation.
- (Optional) To remove an existing page in the header, delete the required page from the **PSPSublocationQuery** preference.

Configure the Filters panel and Chart by list

You can add properties to the **Filters** panel to narrow your search results when you perform a search. Additionally, you can add properties to provide new chart options for the **Chart by** list of the **Quality Management** workspace. To do this, for the added properties, you must set the following property constants as follows:

- Set the **Caw0PSPSearchCanFilter** property constant to **true**.
- Set a priority number for the **Caw0PSPSearchFilterPriority** property constant.

Procedure

Configure the **Filters** panel and **Chart by** list to add the properties as follows:

- In the Business Modeler IDE (BMIDE) application, in the **Business Objects** folder, locate the business object that has the property you want to add.

For Problem Solving processes, locate the **C2CapaRevision** business object. For issues, locate the **C2IssueRevision** business object.

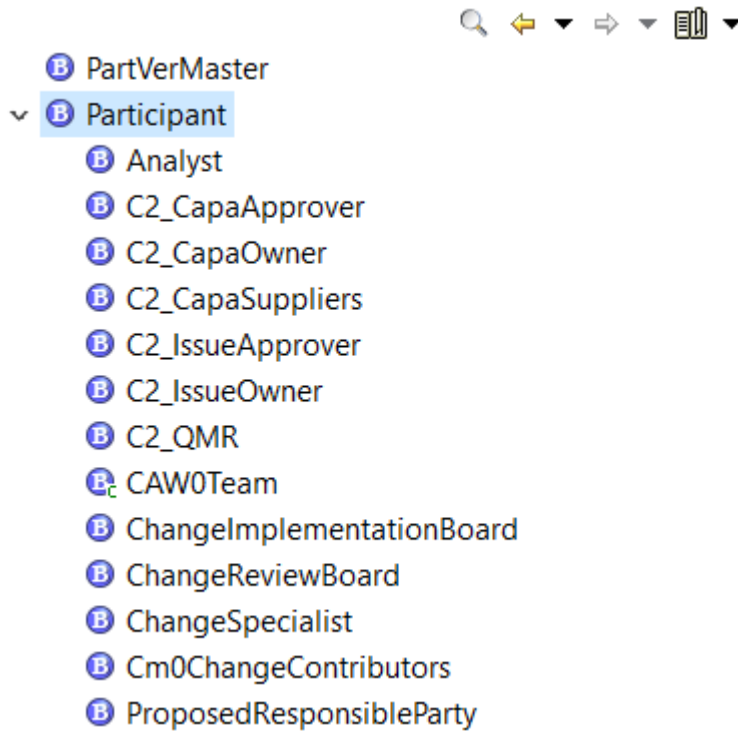
- Right-click the business object, choose **Open**, and click the **Properties** tab in the resulting view.
- In the **Properties** table, select the property whose characteristics you want to change.

4. In the **Property Constants** tab, do the following:
 - a. Set the **Caw0PSPSearchCanFilter** property constant to **true**.
 - b. Set a priority number for the **Caw0PSPSearchFilterPriority** property constant.
5. To save the changes to the data model, choose **BMIDE > Save Data Model**, or on the main toolbar click **Save Data Model**.

Configure the types of users in the My Roles property of the Filters panel and the Chart by list

By default, the **My Roles** property displays the following sections where you can filter users by using these types of users:

- Problem Solving process Team Leader
 - Problem Solving process Champion
 - Problem Solving process Supplier
 - Problem Solving process Team Member
 - Unassigned
1. In the Business Modeler IDE (BMIDE) application, in the **Business Objects** folder, locate the **Participant** business object.
 2. In the **Participant** business object, add the type of user to be displayed in the **My Roles** filter.



Example:

You want to display **Change Specialist** in the **My Roles** filter, so you must add **Change Specialist** to the **Participant** business object.

3. For Problem Solving processes, edit the **CAW0SPParticipantType** preference to add the type of user to be displayed in the **My Roles** filter.

Example:

Add the following to the **CAW0SPParticipantType** preference:

```
ChangeSpecialist
```

4. For issues, edit the **CAW0QIMParticipantType** preference to add the type of user to be displayed in the **My Roles** filter.

Example:


Add the following to the **CAW0QIMParticipantType** preference:

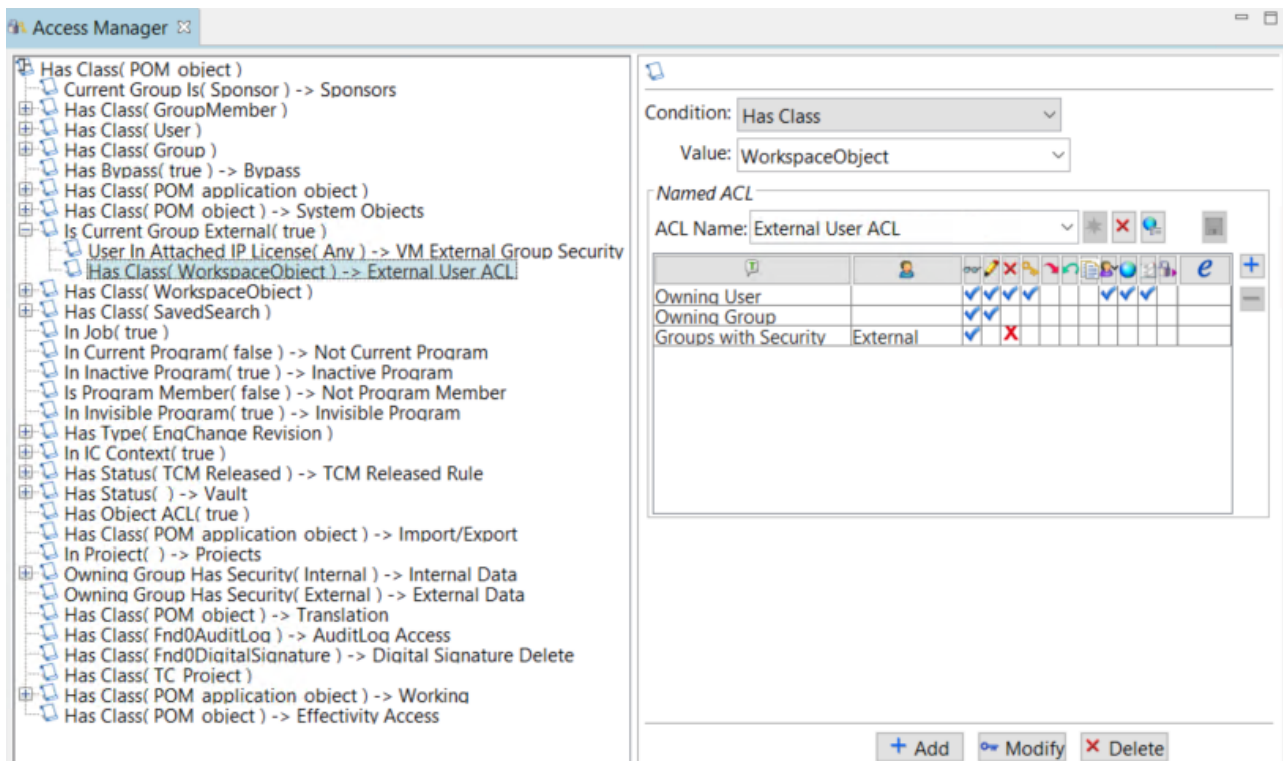
```
ChangeSpecialist
```

Allow External Users to view details of the assigned Problem Solving process

If your company has signed a partner contract with one of your vendors, then the Owners of the issue or the Problem Solving process can assign the partner representatives of the vendor to work on the issue or Problem Solving process as *External Users*. For more information about partner contracts, see *Partner Connect — Usage* in the Teamcenter documentation.

In the assigned Problem Solving process, the *External Users* can view details, such as the problem description defect, root causes, or quality actions, only when you configure the access control list (ACL) for Workspace objects to grant **Read** access to *External Users*. You can choose to allow *External Users* to view the details by configuring the ACL as follows:

1. In Access Manager, select the **Has Class(POM_object)** node in the access rule tree.
2. Expand **Is Current Group External** and select **External User ACL** for Workspace objects.
3. For the External Group, double-click the ACE box for the **Read** privilege, and choose the check mark  to grant privileges.



4. Click **Modify** in the **Named ACL** pane to save the updates to the ACL.
5. Click the **Save** button in the Access Manager toolbar.

Business objects used to plan a Problem Solving process or an issue

You can set up the supplier information by using the following business objects to create the required data:

Supplier Information section	Business object
Vendor	Vendor
Brand	Brand
Partner Contract	Partner Contract
Contact	Company Contact
Location	Company Location

You can set up the customer information by using the following business objects to create the required data:

Customer Information section	Business object
Customer	In BMIDE, create a subclass of the Vendor object to be used as a customer.
Location	In BMIDE, create a subclass of the CompanyLocation object to be used as a customer's location.
Contact	In BMIDE, create a subclass of the CompanyContact object to be used as a customer's contact.

Note:

- In a Problem Solving process or an issue, you can use the **Plan** tab to configure connections to other objects. If you require the **Vendor** or **Retail** objects, then you must install them. If you do not use some of the sections in this tab, you can remove them.
- If you want to use custom objects, you must **specify these objects in the XRT Editor of the web client**.

Configure which business objects can be used to plan a Problem Solving process or an issue

The quality engineer can create a plan to track the associated projects, events, schedules, supplier information, and customer information for a Problem Solving process or an issue. You, as an administrator, can configure which business objects can be used in the **Plan** tab to create this plan.

The following table shows the business objects and the feature to be selected during installation for the various sections of the **Plan** tab:

Section of the Plan tab	Business objects	Feature to be selected during installation
Project Information	Plan and Event	Program Planning
Schedules	Schedule	Schedule Manager
Supplier Information section		
Vendor	Vendor	Vendor Management
Brand	Brand	Retail Footwear and Apparel
Partner Contract	Partner Contract	Vendor Management
Contact	Company Contact business object	This section does not have a dependency on a TEM feature.
Location	Company Location	This section does not have a dependency on a TEM feature.
Customer Information section		
Customer	In BMIDE, create a subclass of the Vendor object to be used as a customer.	This section does not have a dependency on a TEM feature.
Contact	In BMIDE, create a subclass of the CompanyContact object to be used as a customer's contact.	This section does not have a dependency on a TEM feature.
Location	In BMIDE, create a subclass of the CompanyLocation object to be used as a customer's location.	This section does not have a dependency on a TEM feature.

1. In the **XRT Editor** of the web client, open the **Plan** page of the Problem Solving style sheet.
2. In the `objectSet` `source` field, enter the name of the object to be added in the following format:

`< provider_name.object_type >`

The object is attached to Problem Solving process with the **CMReferences** relation.

Example:

```
<section titleKey="tc_xrt_PSP_Brand">
  <objectSet source="CAW0VendorTableDataProvider.Ret0Brand" defaultdisplay="listDisplay" sortby
    ="object_name" sortdirection="ascending">
```

- To remove a section from the **Plan** page of the Problem Solving style sheet, delete the section.

Example:

To remove the **Brand** section, delete the below section in the style sheet.

```
<section titleKey="tc_xrt_PSP_Brand">
  <objectSet source="CAW0VendorTableDataProvider.Ret0Brand" defaultdisplay="listDisplay" sortby
    ="object_name" sortdirection="ascending">
```

View and set up audit logs

For each Problem Solving process and issue, audit logs are generated to track each action being performed on the Problem Solving process and the issue. For a Problem Solving process, the audit logs help you to track the life cycle, from creating the process to defining its preventive actions. For an issue, the audit logs help you to track the life cycle from creating the issue and adding symptom defects to deriving an engineering change request (ECR), engineering change notice (ECN), or a Problem Solving process from the issue. The audit logs also help to troubleshoot any errors that might occur.

The **GENERAL LOGS** section displays the general actions that have been performed on the Problem Solving process or issue. The **WORKFLOW LOGS** section displays the workflow-related actions that have been performed on the Problem Solving process or issue.

You can view the generated audit logs from the following locations:

- Audit Logs** tab available next to the **Reports** tab.

Logged Date	Event Type Name	User ID	Change ID	Reason	Group Name	Role Name	Secondary Object DI...	Secondary Object Ty...	a2bssupplierTableID...	a2bssupplierTableIO...
25-Apr-2023	__Modify	ed			demo	Designer				
25-Apr-2023	__Modify	ed			demo	Designer				
25-Apr-2023	__Attach	ed			demo	Designer	Wiper Assembly	Image		
25-Apr-2023	__Attach	ed			demo	Designer	Damaged_Windshield_...	JPEG		
25-Apr-2023	__Attach	ed			demo	Designer	carwiper	JPEG		
25-Apr-2023	__Attach	ed			demo	Designer	Car wiper not working	CAW0Defect		
25-Apr-2023	__Modify	ed			demo	Designer				
25-Apr-2023	__Create	ed			demo	Designer				

Logged Date	Event Type Name	Performer	Object Name	Object Type	Job Name	Process Template N...	Task State	Task Result	Comments	Task Responsible Pa...
25-Apr-2023	Process Initiated	ed	PSP__Create_Markups_f...	Job	PSP__Create_Markups_f...	PSP__Create_Markups_f...				
25-Apr-2023	Add Attachment	ed	PSP__Create_Markups_f...	EPMTask	PSP__Create_Markups_f...	PSP__Create_Markups_f...	Unassigned	Unset		PSP
25-Apr-2023	Assign	ed	PSP__Create_Markups_f...	EPMTask	PSP__Create_Markups_f...	PSP__Create_Markups_f...	Pending	Unset		ed (ed)
25-Apr-2023	Assign	ed	Create Defects For Mark...	EPMTask	PSP__Create_Markups_f...	PSP__Create_Markups_f...	Pending	Unset		ed (ed)
25-Apr-2023	Start	ed	PSP__Create_Markups_f...	EPMTask	PSP__Create_Markups_f...	PSP__Create_Markups_f...	Started	Unset		ed (ed)
25-Apr-2023	Start	ed	PSP__Create_Markups_f...	Job	PSP__Create_Markups_f...	PSP__Create_Markups_f...				
25-Apr-2023	Start	ed	Create Defects For Mark...	EPMTask	PSP__Create_Markups_f...	PSP__Create_Markups_f...	Started	Unset		ed (ed)

- Audit Logs** tab in the right pane of an Ishikawa, Ishikawa cause, 5Why, Why question, and defects in the **Root Cause Analysis** tab

1. Set the **TC_audit_manager_version** preference to **3**.
2. Set the **AWC_show_audit_logs** preference to **true**.

Configure the availability of related Problem Solving processes

In the **Repeated Defects Evaluation** section of the **Problem Description** tab, by default, you can view all related Problem Solving processes that have been created in the last six months prior to the current one. You can configure this period in the **CAW0_SpanOfRepeatedDefectInDays** preference by specifying the required number of days in the preference.

Configure the display of a Microsoft Excel report

For a Problem Solving Owner to view the contents of a Microsoft Excel report, you must set the **Crf_Report_Excel_Use_OOXML** preference to **True**.

Configure Problem Solving to display localized content

If you are working with Problem Solving in a supported localized language, the Ishikawa cause groups and the Why prefix in a CAPA are still displayed in English.

To display this content in the local language, as a Teamcenter administrator, you must configure the following preferences:

- **CAPA_WhySequencePrefix**

This preference defines the prefix of the Why questions that you add to a 5Why in the **Root Cause** tab of a CAPA. Replace **Why** in the **Value** box with the local language's translation of Why.

▼ DEFINITION

Name:	CAPA_WhySequencePrefix
Product Area:	CAPA Management
Description:	Preference for prefix to populate Why Sequence. Example for the Value Why etc
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	No

▼ VALUES

Values:

Why

- **CAW0IshikawaCauseGroupVal**

This preference defines the names of the Ishikawa cause groups in the **Root Cause** tab of a CAPA. Replace the names in the **Value** box with the local language's translation of the names of the Ishikawa cause groups.

▼ DEFINITION

Name:	CAW0IshikawaCauseGroupVal
Product Area:	Active Workspace
Description:	This pref is to store default value for ICG
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	Yes

▼ VALUES

Values:

Man
Machine
Method
Material
Measurement
Mother Earth

Update the COTS workflow templates using the PSP-derive-problem-solving-items workflow handler

Description

A user would be able to update the COTS workflow templates using the **PSP-derive-problem-solving-items** workflow handler to automate deriving a change request from a problem report, or a change notice from a change request.

The workflow handler provides additional arguments to derive multiple changes based on the number of secondary objects. If the action handler passes all the validations, then the action handler will derive multiple changes corresponding to each secondary object.

Setting **auto_derive** to *true* will automatically derive the proper result when submitted using the default workflow template.

Syntax

PSP-derive-problem-solving-items

[-from_type=<valid change object type>]

[-to_type=<valid change object type>]

[-synopsis=<valid string>]

[-description=<valid string or empty>]

[-template=<process-template-name or default>]

[-propagate_secondary_as=<valid string>]

[-source_property_name=<valid property name>]

[-source_property_value=<valid corresponding property value or empty>]

[-source_relation=<valid corresponding property value or empty>]

[-target_relation=<valid corresponding property value or empty>]

[-description_from_secondary=<empty>]

[-synopsis_from_secondary=<empty>]

Arguments

-from_type

Target object type to derive from. **Derive** will derive from all workflow targets of this type, or subtypes of this.

-to_type

The change object type to be created by the **Derive** operation. Must be a valid change type that can implement the designated **-from_type**.

-synopsis

Synopsis to be assigned to the created change object.

-description

Description to be assigned to the created change object.

-template

Name of the workflow template the newly created change object should be submitted to.

Or, if empty, then use the default workflow template configured for this newly created change type.

Note:

If this argument is not provided, then the newly create change object will not be submitted to a workflow.

-propagate_secondary_as

Type of propagation being done to the secondary object when it is being propagated from the source change item (**from_type**) to the derived change item (**to_type**).

Valid values are **Carryover** and **Duplicate**. All other values are invalid. When **Carryover** is specified, the object is propagated from the source change item to the derived change item as a reference. The same object is attached to the derived change item with the relation specified in the **Argument** list. When **Duplicate** is specified, the object is propagated from the source change item to the derived change item as an object. The existing object is saved as a new object, and the new object is attached to the derived change item with the relation specified in the **Argument** list.

-source_relation

Valid relation name to get the secondary objects from the source target object. If no secondary object is found, the action handler derives the change item only once. It does not propagate a secondary object with this relation to the target relation. An invalid relation name generates an error. If this argument is used with a valid relation name, the **-target_relation** argument is required with a valid relation name.

-target_relation

Valid relation name to attach the object to the new derived object. An invalid relation name generates an error. This argument must be used with **-source_relation**.

-source_property_name

Property name to read from the secondary objects for validation. If this argument is used, **source_property_value** is required too.

This argument is optional.

-source_property_value

Value should match the value from secondary object's property value. If the validation is false, the change object not derived for the corresponding secondary object. An empty value is valid. This argument must be used with - **source_property_name**.

-description_from_secondary

Description to be sourced from the secondary object's description. This argument takes precedence over the -**description** argument.

An empty value is valid.

-synopsis_from_secondary

Synopsis to be sourced from the secondary object's synopsis. This argument takes precedence over the -**synopsis** argument.

An empty value is valid.

Placement

Place on any action. Typically attached to the **Complete** action.

Restrictions

None.

Examples

- This example derives a Problem Solving object from a Quality Issue object only once, without propagating any secondary objects.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision

- This example generates an error because a value is not specified for the -**propagate_secondary_as** argument.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	

- This example generates an error because the **-source_relation** and **-target_relation** arguments are not specified.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover

- This example generates an error because values are not specified for the **-source_relation** and **-target_relation** arguments.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	
-target_relation	

- This example generates an error because invalid relation names are specified for the **-source_relation** and **-target_relation** arguments.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	Invalid relation name
-target_relation	Invalid relation name

- This example derives a Problem Solving object from a Quality Issue object for each source secondary object from the source relation. If there are two source secondary objects, the Problem Solving object is derived twice from the Quality Issue object. The action handler propagates each secondary source object from the source Quality Issue to the target derived Problem Solving object with the specified target relation.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	CPA0SymptomDescription
-target_relation	CPA0ProblemDescription

- This example generates an error because a value is not specified for the **-source_property_name** argument.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	CPA0SymptomDescription
-target_relation	CPA0ProblemDescription
-source_property_name	

- This example generates an error because the **-source_property_value** argument is not specified. This argument is required with the **-source_property_name** argument.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	CPA0SymptomDescription
-target_relation	CPA0ProblemDescription
-source_property_name	caw0Evaluation

- This example derives a Problem Solving object from a Quality Issue object for each source secondary object from the source relation. If there are two source secondary objects, before deriving, the action handler validates that **caw0Evaluation** of the secondary object has an empty value. If it has an empty value, the Problem Solving object is derived from the Quality Issue object. If it does not have an

empty value, the Problem Solving object is not derived from the Quality Issue object for the secondary object.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	CPA0SymptomDescription
-target_relation	CPA0ProblemDescription
-source_property_name	caw0Evaluation
-source_property_value	

- This example derives a Problem Solving object from a Quality Issue object for each source secondary object from the source relation. If there are two source secondary objects, before deriving, the action handler validates that **caw0Evaluation** of the secondary object is **Investigating**. If it is **Investigating**, the Problem Solving object is derived from the Quality Issue object. If it is not **Investigating**, the Problem Solving object is not derived from the Quality Issue object for the secondary object.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	CPA0SymptomDescription
-target_relation	CPA0ProblemDescription
-source_property_name	caw0Evaluation
-source_property_value	Investigating

- This example derives a Problem Solving object from a Quality Issue object for each source secondary object from the source relation. If there are two source secondary objects, before deriving, the action handler validates that **caw0Evaluation** of the secondary object is **Investigating**. If it is **Investigating**, the Problem Solving object is derived from the Quality Issue object. If it is not **Investigating**, the Problem Solving object is not derived from the Quality Issue object for the secondary object.

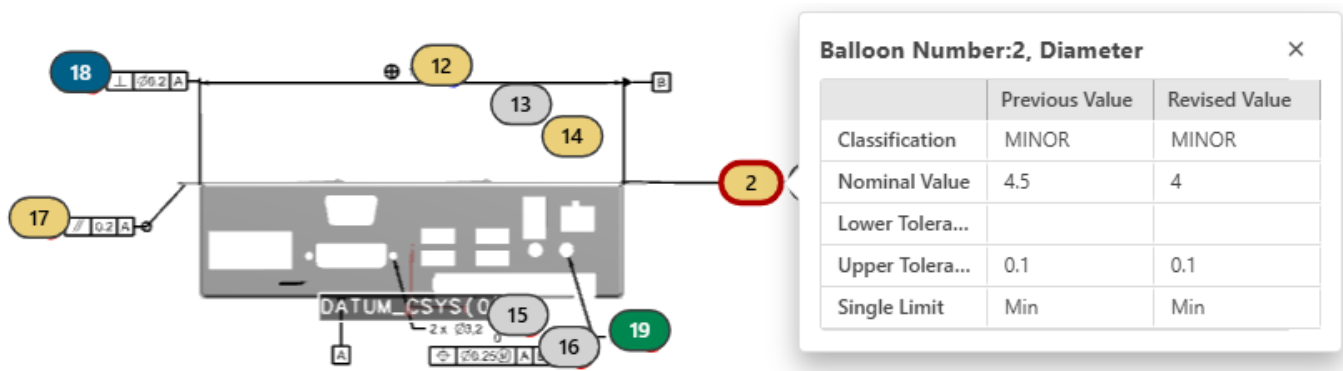
When you use the **description_from_secondary** and **-synopsis_from_secondary** arguments, for each new derived object, the synopsis and description are taken from each secondary object.

Argument	Values
-to_type	C2CapaRevision
-from_type	C2IssueRevision
-propagate_secondary_as	Carryover
-source_relation	CPA0SymptomDescription
-target_relation	CPA0ProblemDescription
-source_property_name	caw0Evaluation
-source_property_value	Investigating
-description_from_secondary	
-synopsis_from_secondary	


11. Configure Control Plan

Configure the properties displayed while comparing PMI for a modified balloon

For a Control Plan or an operation in the Control Plan, after generating Inspection Definitions from a part with product and manufacturing information (PMI), a quality engineer can compare the PMI in the revised part to the original imported PMI and choose to import the modified PMI. When a quality engineer clicks a single modified balloon or a row in the table, a table appears displaying the previous and revised values. As an administrator, you can configure which properties are used to define the uniqueness of characteristics, and these same properties are displayed in the comparison table too.



Procedure

1. On the home page, click the **PREFERENCES** tile.
2. Search for the **ACP_IPXMLKEYS_TO_COMPARE_TWO_BALLOONS** preference, select it from the search results, and click **Edit** .
3. Add additional properties in this format:

IpxmlKey: Internal Name of the Teamcenter property: Business Object Name

The following properties are available by default:

```
B2005:qc0Criticality:Qc0MasterCharSpec  
K2101:qc0NominalValue:Qc0VariableCharSpec  
K2112:qc0LowerTolerance:Qc0VariableCharSpec  
K2113:qc0UpperTolerance:Qc0VariableCharSpec  
K2892:qc0SingleLimit:Qc0VariableCharSpec
```

4. Click **Save** .

Customize the style sheet used to print the Control Plan

To customize the style sheet used to print the Control Plan, you must update the **ControlPlanAIAGReportStyle.xml** file with the required customizations, and import the Control Plan into Teamcenter. When quality engineers print the Control Plan, it will be displayed in the customized style sheet.

To perform the customization, do the following:

1. Locate the **ControlPlanAIAGReportStyle.xml** file in the `\acp0awcontrolplan\controlplanreports\Resources` path of the Teamcenter installation directory.

Example:

```
C:\Apps\tc\tc2406\TR\install\acp0awcontrolplan\controlplanreports\Resources
```

2. Update the **ControlPlanAIAGReportStyle.xml** file with the required customizations.
3. Import the Control Plan report definitions to the Teamcenter server by running the **import_export_reports** utility from the Teamcenter command prompt.

```
import_export_reports [-u=user-id] [-p=password] [-g=group]
[-import]
[-overwrite]
[-stageDir=fully qualified name of the directory that contains all of the report definitions and
its associated data in predefined format.]
[-reportFile=name of the XML file containing the list of report templates. ]
```

-u

Specifies the user ID.

This is generally a user with administration privileges.

-p

Specifies the password.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is considered.

-import

Specifies that report definitions are to be imported to the Teamcenter server.

-overwrite

Overwrites the existing report definitions during import.

-stageDir

Specifies the fully qualified name of the directory that contains all of the report definitions and its associated data in predefined format.

-reportFile

Specifies the name of the XML file containing the list of report templates.

-h

Displays the help for this utility.

Example:

```
import_export_reports -import -overwrite -u=user_name -p=password
-g=dba
-stageDir=${TC_INSTALL_DIR}/acp0awcontrolplan/controlplanreports
-reportFile=${TC_INSTALL_DIR}/acp0awcontrolplan/controlplanreports/
ControlPlanReport.xml
```

- Restart the Teamcenter server manager.

Define the size and resolution of images added to Visual Quality Characteristics

When a quality engineer adds an image to a Visual Quality Characteristic, the image must meet the defined size and resolution limits. This ensures that the image size is not too large. It also ensures the resolution does not exceed the maximum resolution or is smaller than the minimum resolution. You can define the size and resolution limits in the **AQC_VIS_CHAR_IMAGE_CRITERIA** preference. Update the following parameters in the specified format:

- Define the maximum size of the image in kilobytes:

```
max_size:5120
```

- Define the minimum resolution of the image by specifying the minimum width and height in pixels:

```
min_res:50:50
```

- Define the maximum resolution of the image by specifying the maximum width and height in pixels:

```
max_res:1500:1500
```

▼ DEFINITION

Name:	AQC_VIS_CHAR_IMAGE_CRITERIA
Product Area:	Active Workspace
Description:	<p>Defines the image criteria (size and resolution) to validate the image before attaching to Visual Characteristic.</p> <p>Define the maximum size of image in below format. max_size:[Maximum size of image in kb]</p> <p>Define the image resolution limit in below format. min_res:[Minimum width of image in pixels]:[Minimum height of image in pixels] max_res:[Maximum width of image in pixels]:[Maximum height of image in pixels]</p>
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	Yes

▼ VALUES

Values:

max_size:5120

min_res:50:50

max_res:1500:1500

Configure the name pattern for Inspection Definitions

When a quality engineer adds Inspection Definitions to an operation or generate Inspection Definitions from a part that contains product and manufacturing information (PMI), the Inspection Definitions are created with a specific name pattern. You can configure this name pattern in the **AQC_DEFINE_INSPEC_DEF_NAME_BY_CHAR_SPEC_PROPS** preference. By default, the Inspection Definitions are created with the **object_name** pattern. The name of the Inspection Definition appears with the **Inspect-** prefix in the following format: **Inspect-*<value defined in the preference>***.

You can configure the **Inspect-** prefix according to your business requirements. To update the value of the prefix, navigate to `C:\Apps\tc\tc<version number>\TR\lang\textserver\en_US`, open the `aqcOqcmaw_text_locale.xml` file, and update the value of the `tc_prefix_Inspection_Definition_Name` element to the required prefix.

▼ DEFINITION

Name:	AQC_DEFINE_INSPEC_DEF_NAME_BY_CHAR_SPEC_PROPS
Product Area:	Active Workspace
Description:	Specify the properties of Characteristic Specification required to form the Inspection Definition name.
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	Yes

▼ VALUES

Values:

object_name

Define the mapping between the properties of PMI balloons and Inspection Definitions

When a quality engineer imports the product and manufacturing information (PMI) balloons from a part into the Inspection Definitions of a Control Plan, Quality Characteristics are generated from the source balloons and are added to the generated Inspection Definitions. You can define the mapping between the properties of PMI balloons and Inspection Definitions in the **ACP_IPXMLKEY_INSPDEFPROPS_MAPPING** preference. If you have custom balloon properties, you can define their mapping into an Inspection Definition property in this preference.

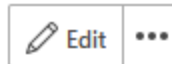
Note:

To define the mapping for First Article Inspection (FAI) balloons, add the following:

```
<Property name of the FAI balloon in the IPXML file>:aqc0IsFAIRquired
```



ACP_IPXMLKEY_INSPDEFPROPS_MAPPI...



▼ Definition

Name:	ACP_IPXMLKEY_INSPDEFPROPS_MAPPING
Product Area:	Active Workspace
Description:	Maps the IPXML keys (Balloon information) to the Inspection Definition(Aqc0CharElementRevision) properties. Accepts IPXML keys and Inspection Definition props separated by a colon, as follows: IPXML Key:Inspection Definition-property-name Default Values:- K2002:aqc0CharacteristicNumber K2900:aqc0Remark Do not remove Default values
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	Yes

▼ Values

Values:

K2002:aqc0CharacteristicNumber
K2900:aqc0Remark
BCT_UNIQUE_OBJ_HANDLE:aqc0PmiUid
SVG:Aqc0LinkToControlFrame
B2004:aqc0IsFAIRRequired

Define the mapping between the properties of PMI balloons and Quality Characteristics

When a quality engineer imports the product and manufacturing information (PMI) balloons from a part into the Inspection Definitions of a Control Plan, Quality Characteristics are generated from the source balloons and are added to the generated Inspection Definitions. You can define the mapping between the properties of PMI balloons and Quality Characteristics in the **ACP_IPXMLKEY_TCQPROPS_MAPPING**

preference. If you have custom balloon properties, you can define their mapping into a Quality Characteristic property in this preference.

Note:

To define the mapping for **Single Limit Tolerance**, add the following:

```
<Property name of the Single Limit Tolerance in the IPXML file>:qc0SingleLimit
```



ACP_IPXMLKEY_TCQPROPS_MAPPING



K2892 key is used for Single Limit Tolerance like Min, Max, etc. If value value is present for this key, we will consider it Single Limit Tolerance.
Do not remove mandatory values

Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	Yes

▼ Values

Values:

K2101:qc0NominalValue
 K2112:qc0LowerTolerance
 K2113:qc0UpperTolerance
 B2005:qc0Criticality
 B2009:object_desc
 B2009:qc0DimensionType
 TQC1000:qc0Context
 K2142:qc0UnitOfMeasure
 B2005:qc0OkDescription
 K2110:qc0LowerAllowance
 K2111:qc0UpperAllowance
 K2504:mci0ChangeStatusText
 K2002:mci0PmiID
 BCT_UNIQUE_OBJ_HANDLE:mci0BCTUniqueObjHandle
 BCT_CHX_UID:mci0PMICharacteristicsUid
 K2110:qc0LowerSpecificationLimit
 K2111:qc0UpperSpecificationLimit
 K2892:qc0SingleLimit

While importing PMI, in some cases, the Quality Characteristics already exist in the Quality Master Data. In such cases, Control Plan does not create new Quality Characteristics, but uses the existing Quality

Characteristics, and attaches them to the generated Inspection Definitions. To find out if the Quality Characteristics already exist in the Quality Master Data, Control Plan performs a search by using Quality Characteristic properties as the search criteria. You can define which properties are used as the search criteria by adding the internal names of the properties in the following preferences:

- **ACP_Qc0VariableCharSpec_FINDPROPS**: Used for searching Variable Quality Characteristics
- **ACP_Qc0AttributiveCharSpec_FINDPROPS**: Used for searching Attributive Quality Characteristics

Note:

String, Double, Boolean, Integer, and Reference properties are supported as search criteria. **Relation** properties are not supported in the search criteria.

If you have added custom List of Values (LOV) properties to the Quality Characteristic, and you want to import the balloon property into the custom LOV property, you must create preferences to define the mapping between the custom LOV property and the balloon property. If the mapping is not done, then the correct value from the balloon will not be assigned to the custom LOV property in the Quality Characteristic, and the default value will be assigned. You must create preferences for each of the balloon properties that you want to import into the custom LOV properties in the following format:

ACP_<Internal name of the property>_VALUES

In this preference, you must specify the Display Value or IPXML Key Value and internal name of LOV separated by a colon in this format:

IPXML Key Value or LOV Display Value:internal LOV value

Example:

For the **ACP_qc0Criticality_VALUES** preference, specify the values as:

```
MINOR:indicatorFMEAMinorSeverity
MAIN:indicatorFMEAMajorSeverity
CRITICAL:indicatorCriticalCharacteristic
```

You must also create a preference to define the default value that is assigned to the custom LOV property in the following format:

ACP_<Internal name of the property>_DEFAULT

In this preference, you must specify the default value of the Display Value or IPXML Key Value and internal name of LOV separated by a colon in this format:

IPXML Key Value or LOV Display Value:internal LOV value

Example:

For the **ACP_qc0Criticality_VALUES** preference, specify the values as:

```
indicatorFMEAMinorSeverity
```

Enable creating Quality Characteristics from parts with product and manufacturing information

To allow quality engineers to create Quality Characteristics from parts with product and manufacturing information (PMI), by using the **Import PMIs As Characteristics** workflow, set the **ACP_SUPPORT_PART_BASED_CHARACTERISTICS** preference to **True**. This workflow imports the balloons with PMI from this part, and creates Quality Characteristics for each balloon. The Quality Characteristics are attached to the part, which allows you to maintain the link between the Quality Characteristics and their source parts.

This workflow imports the balloons and creates the Characteristic Groups and their Quality Characteristics in the **Quality Master Data** location. When you open the Quality Characteristic and select its **Where Used** tab, the source part is displayed in the **Used in Structures** section.

Configure which rule file is used to generate, compare, and align a Control Plan with FMEA

After a quality engineer generates a Control Plan from an FMEA, the quality engineer might make updates to the source FMEA and the target Control Plan. To ensure that the target Control Plan and source FMEA contain the latest data, the quality engineer must compare and align the updates between the source FMEA and the target Control Plan.

The **ACP_GENERATE_CPIP_FROM_FMEA_STRUCTURE_CONFIG_FILE** preference value defines the `Dataset` name that holds the rule file, which will be used to generate, compare, and align a Control Plan with FMEA.

▼ DEFINITION

Name:	ACP_GENERATE_CPIP_FROM_FMEA_STRUCTURE_CONFIG_FILE
Product Area:	Active Workspace
Description:	Defines Dataset name which holds the rule file, which will be used to generate, compare and align CPIP structure from or with FMEA structure.
Protection Scope:	Site
Environment:	Disabled
Location:	Site
Type:	String
Multiple Values:	No

▼ VALUES

Values:

Acp0GenerateCpipFromFmeaStructureRules

Configure the default Tolerance Type for Variable Quality Characteristics

When a quality engineer imports the product and manufacturing information (PMI) balloons from a part into the Inspection Definitions of a Control Plan, Quality Characteristics are generated from the source balloons and are added to the generated Inspection Definitions. You can define the default value for **Tolerance Type** of **Variable** Quality Characteristics in the **AQC_VarCharSpecToleranceType** preference.

Configure the relation of a drawing with its attached IPXML file

When you select a part to display its drawing in the **PMI** tab, the drawing has an attached IPXML file with a specific relation. You can configure the type of relation in the **ACP_OBJECT_SEARCH_TO_RENDER_DATASET_INFO** preference.

Define which groups can view reference details of an Inspection Definition

You can view the following reference details when you select an Inspection Definition in the primary work area:

- **Operation**

- **Part Reference**
- **Manufacturing Device**
- **Inspection System**

To define which groups can view the reference details of an Inspection Definition, run the following utility:

```
import_uiconfig -u=tc_admin_user_name -p=tc_admin_password -g=groups that must  
be given access to view reference details -file=${TC_INSTALL_DIR}/aqc0qcmaw/data/  
Aqc0CharSpecUiConfigCots.xml
```

Configure the mode of importing PMI balloons for a quality engineer

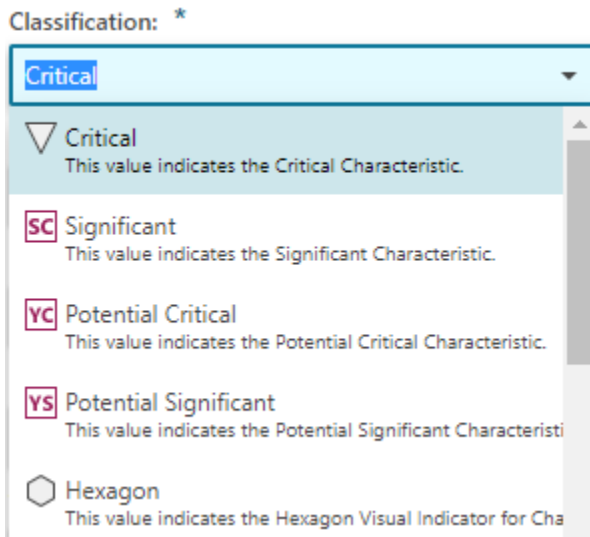
When a quality engineer imports the product and manufacturing information (PMI) balloons from a part into the Inspection Definitions of a Control Plan, Quality Characteristics are generated from the source balloons and are added to the generated Inspection Definitions.

By default, Control Plan imports the PMI balloons in the *synchronous* mode when a quality engineer selects less than 20 balloons in the **PMI** tab. If a quality engineer selects more PMI balloons, Control Plan imports the balloons in the *asynchronous* mode. In any case, selecting a large number might impact performance if the balloons are imported in the synchronous mode.

In the **ACP_Config_SyncAsync_Import_Balloon_Call** preference, you can specify the number of balloons that can be imported in the synchronous mode. If the quality engineer selects more than the number of balloons specified in this preference, Control Plan imports the balloons in the asynchronous mode.

Customize the visual indicators for the Classification list of values in a Quality Characteristic

You can customize the visual indicators that appear in the values of the **Classification** list in a Quality Characteristic. This allows a quality engineer to identify the criticality of a specific classification when they are working with a Quality Characteristic.



To perform the customization, do the following:

1. In BMIDE, add the custom values of the **Classification** list in the following format:

```
iconName = 'indicator'+<name of the value that appears in the Classification list>
```

In this example: `iconName = indicatorCriticalCharacteristic`

The internal name of the **Classification** list is **Qc0Criticality**.

2. Use configuration builder to register indicators in Configuration and Extensibility in the Teamcenter documentation.
3. To display the custom visual indicators in the Control Plan norm document, do the following:
 - a. Convert the visual indicator image to the Base64 string format by using an online tool, such as www.base64-image.de.
 - b. Locate the **ControlPlanAIAGReportStyle.xsl** file in the `\acp0awcontrolplan\controlplanreports\Resources` path of the Teamcenter installation directory.

Example:

```
C:\Apps\ctc2312\TR\install\acp0awcontrolplan\controlplanreports\Resources
```

- c. Add an entry for the custom visual indicator image by using the new LOV value and the image Base64 encoding string as below:

```

<xsl:if test="$imageName = '<i>name of the value that appears in the
Classification list</i>' ">
    <xsl:value-of select="'data:image/svg+xml;base64,<i>image
base64 encoding string</i>' " />
</xsl:if>

```

- d. Import the Control Plan report definitions to the Teamcenter server by running the **import_export_reports** utility from the Teamcenter command prompt.

```

import_export_reports [-u=user-id] [-p=password] [-g=group]
[-import]
[-overwrite]
[-stageDir=fully qualified name of the directory that contains all of the report definitions
and its associated data in predefined format.]
[-reportFile=name of the XML file containing the list of report templates. ]

```

-u

Specifies the user ID.

This is generally a user with administration privileges.

-p

Specifies the password.

-g

Specifies the group associated with the user.

If used without a value, the user's default group is considered.

-import

Specifies that report definitions are to be imported to the Teamcenter server.

-overwrite

Overwrites the existing report definitions during import.

-stageDir

Specifies the fully qualified name of the directory that contains all of the report definitions and its associated data in predefined format.

-reportFile

Specifies the name of the XML file containing the list of report templates.

-h

Displays the help for this utility.

Example:

```
import_export_reports -import -overwrite -u=user_name -p=password  
-g=dba  
-stageDir=${TC_INSTALL_DIR}/acp0awcontrolplan/  
controlplanreports  
-reportFile=${TC_INSTALL_DIR}/acp0awcontrolplan/  
controlplanreports/ControlPlanReport.xml
```

- e. Restart Teamcenter server manager.

12. Configure Training and Qualification

Define the access privileges for Training and Qualification

When the team members work with *qualification units*, *qualification profiles*, and *qualification records* in Training and Qualification, you must ensure that they have access to only the data that they have created. You must also set up users who manage all types of Training and Qualification data. These users must be able to view, modify, and delete Training and Qualification data. For example, you can consider these users as the *Training Coordinators* of your team.







To complete the previous requirements, you must create a role and a group for Training and Qualification.






Example:

For example, create **Training Coordinator** as the role and **Training and Qualification** as the group. Then, assign the **Training Coordinator** role to the **Training and Qualification** group. Add users who are *Training Coordinators* for your team to the **Training and Qualification** group. Then, you create the access control lists (ACL) with the required accessors to ensure that only the owning users and the users assigned with the **Training Coordinator** role can manage Training and Qualification data.

Procedure

1. Click the **PEOPLE** tile.
2. Create a role as follows:
 - a. Click the **Roles** page.
 - b. From the **Primary** toolbar, click **New** ✨ > **Add**.
 - c. Enter the name of the role, for example, **Training Coordinator**, in the **Role** box.
 - d. Click **Add**.
3. Create a group for Training and Qualification as follows:
 - a. Click the **Groups** page.
 - b. From the **Primary** toolbar, click **New** ✨ > **Add**.
 - c. Enter the name of the group, for example, **Training and Qualification**, in the **Name** box.
 - d. Enter the remaining properties, and then click **Add**.

- e. Ensure that the group is selected, and click **Add Role**  in the **ROLES** section.
 - f. In the **Search** tab, search for and select the role that you have created.
 - g. Click **Add**.
4. Create the ACLs to ensure that only the owning users and users assigned with the required role, for example, **Training Coordinator**, can view, modify, and delete the qualification units, qualification profiles, and qualification records:
- a. In Teamcenter, open Access Manager.
 - b. Select the **Has Class (WorkspaceObject)** node in the access rule tree.
 - c. In the branch editor pane, in the **Condition** box, select **Has Class**.
 - d. Select an option from the **Value** list:
 - **Tq0QualificationUnit** for qualification units
 - **Tq0QualificationProfile** for qualification profiles
 - **Tq0QualRecord** for qualification records
 - e. In the **ACL Name** box, type the ACL name.
 - f. Click **Create**  next to the **ACL Name** box.
 - g. Grant **Read**, **Write**, and **Create** privileges to owning users as follows:
 - A. Click **Add**  to add a new row to the ACE table.
 - B. Double-click the cell in the **Type of Accessor** column, and select **Owning User**.
 - C. Double-click the empty ACE boxes for the **Read**, **Write**, and **Delete** privileges, and choose the check mark  to grant privileges.
 - D. Click **Save**  in the **Named ACL** pane to save the ACL.
 - h. Grant **Read**, **Write**, and **Create** privileges to the required role, for example, **Training Coordinator**, as follows:
 - A. Click **Add**  to add a new row to the ACE table.















- B. Double-click the cell in the **Type of Accessor** column, and select **Role**.
 - C. Double-click the cell in the **ID of Accessor** column, and select **Training Coordinator**.
 - D. Double-click the empty ACE boxes for the **Read**, **Write**, and **Delete** privileges, and choose the check mark  to grant privileges.
 - E. Click **Save**  in the **Named ACL** pane to save the ACL.
- i. Deny **Read**, **Write**, and **Create** privileges to all users as follows:
 - A. Click **Add**  to add a new row to the ACE table.
 - B. Double-click the cell in the **Type of Accessor** column, and select **World**.
 - C. Double-click the empty ACE boxes for the **Read**, **Write**, and **Delete** privileges, and choose the cross mark  to deny privileges.
 - D. Click **Save**  in the **Named ACL** pane to save the ACL.

Condition:

Value:

Named ACL

ACL Name:

					
Owning User					
Role	Training Coordinator				
World					

- j. Click **Add** to modify the branch.
 - k. Click **Save** in the Access Manager toolbar.
 - l. Repeat the ACL process, beginning with **selecting an option from the Value list**, until you have completed the creating ACLs for qualification units, qualification profiles, and qualification records.
5. After you set up the roles, groups, and ACLs, **assign designated users to the role that you have created for the Training and Qualification group**.

Assign designated users to manage Training and Qualification data

After you set up the role, group, and access control lists (ACL) for Training and Qualification data, you must assign designated users to manage this data.

Procedure

1. Click the **PEOPLE** tile.
2. Click the **Organization** page.
3. Add users to the role you have created for the Training and Qualification group as follows:
 - a. Select the group, for example, **Training and Qualification**, and expand it.
 - b. Select the role, for example, **Training Coordinator**, you have created for the Training and Qualification group, and, in the **Users** pane, click **Add User** ⊕.
 - c. In the **Search** tab, search for and select the users that you have created.
 - d. Click **Add**.


Create the Training and Qualification checklist master data

A checklist consists of a list of questions that define the acceptance criteria for a qualification unit. The checklists allow the manager to perform an effectiveness review of the training for all employees to ensure that the employees understood the training effectively. Additionally, trainers can use the checklist to prepare an Effectiveness Review test. Also, the trainees can go through the checklist at the end of the training to verify that they have understood the training content correctly.

Each qualification unit can have one or more checklists associated to it, and each checklist can have multiple chapters, questions, and subquestions. Multiple questions and subquestions ensure that none of the acceptance criteria is inadvertently overlooked. Further, the questions are designed in such a way that they can be answered with a simple *yes* or *no*. You, as the *Training Coordinator*, must create and release the checklists in the **Quality Master Data** library. You can then assign the checklists to the required qualification unit.

1. On the **Home** page, click the **QUALITY MASTER DATA** tile.
2. Click the **Checklists** tab.
3. Click **New** ✱ > **Add**.
4. In the **Add** panel, specify a value for the **Checklist ID**, and enter a **Name** for the checklist.

You can optionally add a description if required.

5. From the **Checklist Type**, select the **Checklist**.
6. From the **Checklist Area** list, select **T&Q**.
7. From the **Industry Standard** list, select **T&Q Checklists**.
8. Select a **Calculation Method** and specify the checklist number, if required.
9. In the **Settings** section, select the **Assessment Required** and **Mandatory** check boxes if required.
10. Click **Add** to create the checklist in the **Quality Master Data** library.
11. To add a **Chapter**, **Question**, or **Sub-Question** to the checklist, do the following:
 - a. Select the checklist and click **New**  **Add**.
 - b. In the **Add** panel, specify the **Checklist ID** and **Name**.
 - c. From the **Checklist Type** list, select **Chapter**, **Question**, or **Sub-Question**.

In the tree view, you can also drag and drop checklists, chapters, questions, or sub-questions from one element to other.

- d. From the **Checklist Area** list, select **T&Q**.
- e. Select a **Calculation Method** and specify the checklist number, if required.
- f. In the **Settings** section, select the **Assessment Required** and **Mandatory** check boxes if required.
- g. Click **Add**.

The **Chapter**, **Question**, or **Sub-Question** are included in the checklist. After you release the checklist and its chapters, questions, or sub-questions are available to be added to the qualification unit.

Enable quality engineers to add reference documents to a qualification unit

You can enable users to add a reference document by setting a preference.

In Teamcenter, quality engineers can add the following types of documents to a qualification unit:

- **Quality documents:** Automatically updates linked qualification units when the document is revised. This provides trainees with the latest information through the updated document.
- **Reference documents:** Revisions to these documents do not update linked qualification units. This type of document includes resources, such as case studies, training materials for reference, videos, or images that serve as supplementary material for the qualification unit.

Procedure

1. On the **Home** page, click the **PREFERENCES** tile.
2. In the **Search** box, type **TQ_QualityDocumentEnableFlag**.

Note:

By default, the preference value is set to `true`.

3. Do one of the following:
 - To enable the users to add reference document, set the preference value to `false`.
 - To disable the adding of a reference document, set the preference value to `true`.
4. Sign out and log in again, to apply the changes.

Configuring preferences for custom Training and Qualification objects

You can configure preferences to control which type of qualification units and qualification definitions are available for assignment to different types of qualification profiles and qualification profile groups. Users then view only the type of Training and Qualification objects based on the defined preferences, relevant to them.

To do this, you must set the following preferences:

To control this	Set the preference	Sample Value
The different types of qualification units that are available for assignment to different type of	TQ0_QDvsQU_Allowed_Types	<i>Tq9DISWDefinition:Tq9DISWUnit</i>

To control this	Set the preference	Sample Value
qualification definitions.		
The different types of qualification units that are available for assignment to different type of qualification profiles.	TQ0_QPvsQU_Allowed_Types	<i>Tq4DISWProfile:Tq4DISWUnit</i>
The different types of qualification units that are available for assignment to different type of qualification profile groups.	TQ0_QPGvsQU_Allowed_Types	<i>Tq6DISWProfileGroup:Tq6DISWUnit</i>
The different types of qualification definitions that are available for assignment to different type of qualification profiles.	TQ0_QPvsQD_Allowed_Types	<i>Tq8DISWProfile:T80DISWDefinition</i>
The different types of qualification definitions that are available for assignment to different type of qualification profiles groups.	TQ0_QPGvsQD_Allowed_Types	<i>Tq2DISWProfileGroup:Tq2DISWDefinition</i>
The different types of qualification definitions that are available for assignment to different types	TQ0_SOURCEOBJECTvsQD_Allowed_Types	<i>PartRevision:Tq9CustomQualDefinition</i>

To control this	Set the preference	Sample Value
of source objects.		

Note:

If you do not have any custom objects, then do not delete the default values.