TEAMCENTER

Aerospace and Defense on Rich Client — Usage

Teamcenter 2312

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

Getting started with Aerospace and Defense

Using CAGE codes and company location codes

Working with Aerospace and Defense objects

Working with notes

Change management in Aerospace and Defense

Managing contract data

Aerospace and Defense Training Program

Aerospace and Defense data model

1. Getting started with Aerospace and Defense

Overview of Teamcenter Aerospace and Defense

The Aerospace and Defense solution provides data model extensions and features that you access through standard Teamcenter applications, such as My Teamcenter, Change Manager, Schedule Manager, and Structure Manager. The solution provides multiple industry-specific features that enable you to:

- Execute complex document-centric or part-centric development programs.

- Manage parts lists and standard parts.

- Use standard and custom notes to provide additional design details about parts and documents.

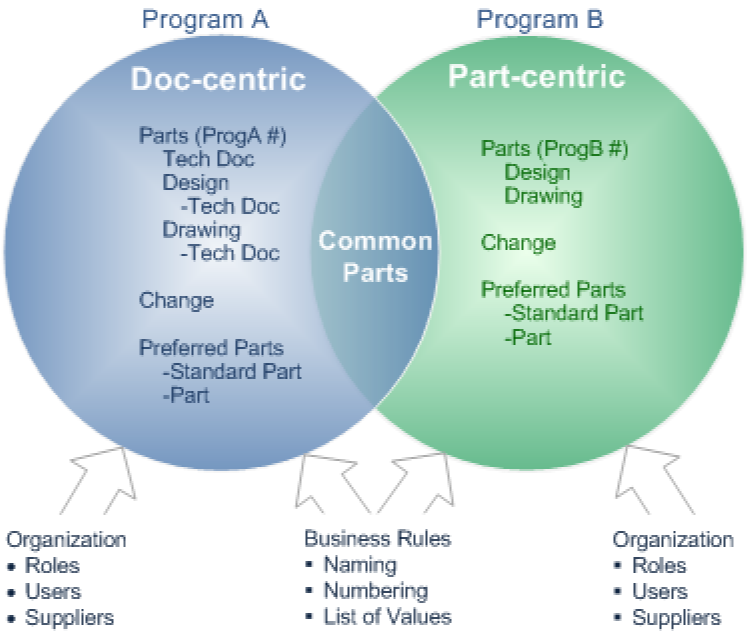
- Manage the program life cycle and its data, including program requirements, deliverable schedules, and related changes.

- Protect intellectual property and national security while promoting collaboration.

In addition, you can extend the Teamcenter Aerospace and Defense solution with other Teamcenter optional modules.

Components of the Aerospace and Defense solution

Program management



A program identifies a collection of resources and assets bound by a common objective, for example the design and development of a new aircraft.

The Program application in Teamcenter includes **Project Administration** to create and administer programs and **Smart Folder Administration** to configure filtering criteria for displaying program data to users.

Using programs you can:

- Segregate data and users.

  You can create, search, and modify only those objects of a program for which you have access.

- Write business rules based on programs.

  Business *rules* are decision points that govern the behavior of business objects, including how they are named, what actions can be undertaken on them, and so on.

  Business rules consist of naming rules, display rules, extension rules, deep copy rules, revision naming rules, list of values, conditions, and so on.

- Write access rules based on programs.

You can enable program-level security to restrict access to program data. You can extend the default security rules to the program team members on a program-by-program basis.

- Configure items to be used as preferred parts in a program.

- Create document-centric or part-centric programs.

  A document-centric program requires you to create a source document (technical document) before you create a part. A part-centric program does not require you to create a source document when creating parts.

  You can assign to or remove data can from programs manually or when the data item is created, and assign items to more than one program.

Programs versus projects

Programs and *projects* work the same. They are both used to organize data and both can grant or control data access to users. The only difference between them are the program security rules. The Aerospace and Defense industry works with programs because of the high level of security provided with the program security rules. Program-level security can be enabled to restrict access to program data. The default security rules can be extended to grant read access to program data to members of the program team, on a program-by-program basis.

The access that groups and roles have to programs is similar to the access they have to projects. Again, program security rules can be applied to programs, introducing a higher level of security. When the program security attribute on a project is set to *true*, Teamcenter considers the project as a program and program-level access rules are applied.

Part management

The parts management functionality offers the following features:

- Parts

  Aerospace and Defense solution provides three business objects relating to parts. These business objects contain attributes specific to Aerospace and Defense.

Part	Represents a product, part, or component. Parts represent the physical parts that make up the products of your enterprise. Parts can comprise assemblies, components, standard parts, and so on. Each part can have one or more computer-aided designs (CAD) associated with it. Components represent the individual pieces of a product. They are the lowest level of the product structure. A component could be any item used in an assembly as part of the assembly's product structure.
Design	Represents the design of a component, assembly, or part.

Drawing	Represents a technical illustration that details out one or more assemblies and parts created on a source technical document.

- Technical documents

  Technical documents are the written form of technical information such as part lists, drawings, procurement specifications, and schematics. A document-centric program requires a source document to create a part, assembly, design, or drawing. Aerospace and Defense uses the technical document as the source document.

  The **Technical Document** object represents a technical document.

- Assemblies

  Assemblies represent groupings of parts. It can include other assemblies, components, standard parts, and so on. The assembly structure can be built either through Structure Manager or through CAD applications.

- Standard parts

  Standard parts are specific to the Aerospace and Defense industry. They represent parts that are used across multiple programs and whose design is controlled by a standard specification specified by the military, an industry, or a company. You can set a standard part as a preferred part for a program.

  In the Aerospace and Defense solution, the **Commercial part** object represents a standard part.

  > Note:
  >
  > You can only use the **Commercial part** in an assembly if the part is a preferred part to all the programs to which the assembly is associated with.

- Reports

  You can generate part-centric part list reports.

  The part centric parts list reports show the details of the structure of an assembly.

- Notes

  You can enhance the source document and parts with notes. Notes provide additional design details about the product structure and configuration. Notes can be of the following types:

  Notes can be of the following types:

Standard notes	Contain generic information that is applicable globally and can be associated with multiple source documents or parts.

Custom notes Are defined for a specific document or part and provide specific information about that document or part.

Change management

You can create change objects containing Aerospace and Defense specific attributes using the Change Manager application. Aerospace and Defense supports the following change objects:

- Problem report
- Change request
- Change notice
- Deviation request
- Work breakdown

CAGE codes

A CAGE Code is a unique five-character identifier assigned as part of the NATO Codification Systems (NCS) to suppliers to various government or defense agencies. CAGE codes provide a standardized method of identifying a given facility at a specific location. This reference enables users of the NCS to determine who supplies any given part. A cage code chart provided by the NATO AC/135 committee (the group of National Directors on Codification) describes the syntax of CAGE codes in various countries.

With the Aerospace and Defense solution, you can:

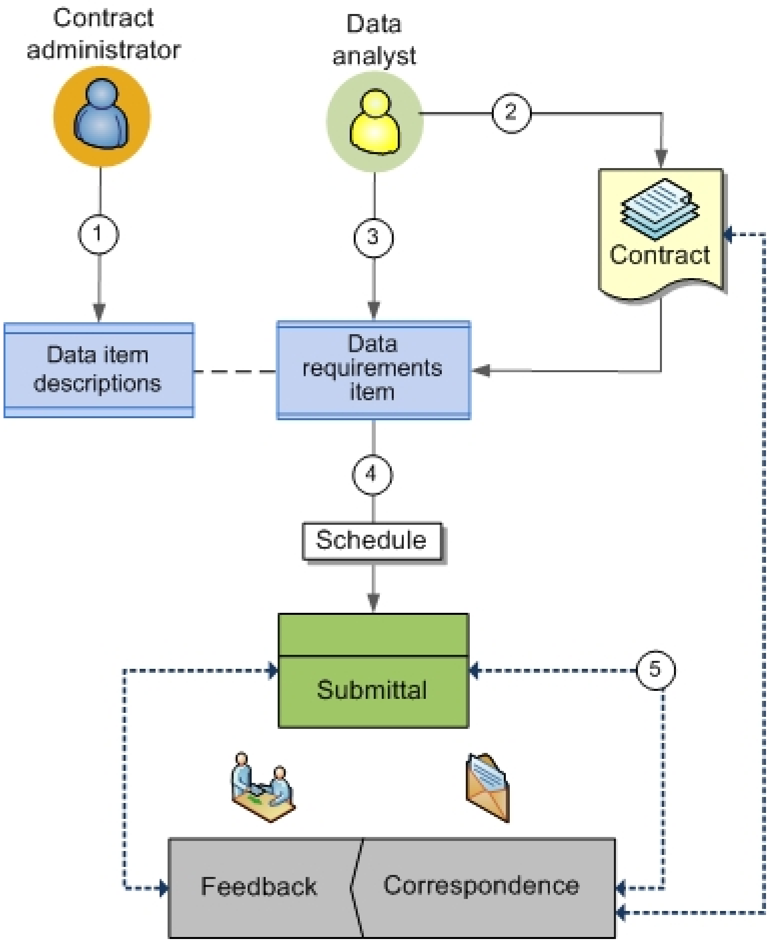
- Register CAGE code for a company location.
- Set a default CAGE code preference for a user.
- Assign CAGE code to a part.
- Upgrade legacy data to populate CAGE code from the owning organization.
- Assign CAGE code to a part through CAD integrations.

Contract data management

Contracts are pivots that drive business across different industry sectors. A contract is a structured procurement document listing the milestones and the schedule dates of the contract event. This procurement document can be a list of data requirement items, such as design information, drawings, status reports, and so on.

After the contract is created, the data analyst creates a workflow schedule to collate the feedback and the review comments that are generated during the data item approval process. This is done by initiating correspondence related to the contract.

A graphical representation of managing contract data through Teamcenter is shown in the following figure.



The sequence of steps is as follows:

1. Create data item descriptions (DIDs).

   DIDs are predefined contract forms that specify data content, data format, and other specific instructions for the suppliers.

   You can maintain a DID repository to associate a data requirement item (DRI) to the contract.

2. Create a contract.

   The administrator or the data analyst creates a contract and contract revision in Teamcenter. You can also choose to create a contract event for scheduling the contract review and feedback process.

3. Create data requirement items.

   You must attach a data requirement item (DRI) to the contract. A DRI is a technical document or deliverable that forms the basis for structured procurement. It represents a single contractual data item as described in DID.

4. Define submittal delivery schedule for the deliverable.

   After the DRI is specified, you must initiate the review process. Submittal represents the data item submittal package or documentation that relates to the actual documents submitted.

5. Provide feedback.

   It is important to monitor the correspondence and feedback related to the data item submittal package between you and the supplier.

Aerospace and Defense objects

In the Aerospace and Defense solution, objects are used to represent product data, such as parts, assemblies, documents, drawings, designs, product requirements, and specifications. These objects are stored in the database and can be accessed by users across the enterprise.

Objects can be associated with each other. The association between the objects is defined through a relationship.

The following table lists the Aerospace and Defense solution data model objects.

Symbol	Object	Description
	Technical Document (**ADSTechDocument**)	Are the written (printed or digital) form of technical information such as reports, drawings, procurement specifications, parts, lists, interface control documents, and schematics. Technical documents serve as a source for creating parts and assemblies for a product.
	Part (**ADSPart**)	Are components or assemblies that are included in the product. A *component* is typically the smallest discrete piece considered in the design and production processes. An *assembly* comprises several components and (optionally) other assemblies.
	Design (**ADSDesign**)	Represents the computer-aided design (CAD) of a component or assembly or an ADS part.

Symbol	Object	Description
	Drawing (**ADSDrawing**)	Represents a drawing. They are related to technical documents.
	Commercial part (**CommercialPart**)	Represents a standard part. Standard parts represent commonly used parts that have been identified as a standard design by a company, an industry, or the military. Note: The commercial part business object is included in the vendor management template.
	Standard Note (**Ads0StdNote**)	Represents the notes established by the Standards Engineering organization that conform to the design practices of a program. Standard notes can be used by all designers.
	Custom Note (**Ads0CustomNote**)	Represents the unique notes for an individual part or document and are created by the designer responsible for the part or document.
	Contract data (**Cdm0Contract**)	Represents the contract document used in document-centric programs.
	Contract revision (**Cdm0Contract Revision**)	Represents the association of the contract revision data with data requirement item.
	Data requirement item (**Cdm0DataReqItem**)	Represents an individual data requirement in a data requirement list.
	Data requirement item revision (**Cdm0DataReqItem Revision**)	Represents the specifications of the data item requirement.
	Data item description (**Cdm0DID**)	Are predefined contract forms that specify data content, data format, and other specific instructions for the suppliers. Maintain a DID repository to associate a data requirement item (DRI) to the contract.
	Data item description revision (**Cdm0DIDRevision**)	Represents the specifications of the data item description.
	Submittal (**Cdm0Submittal**)	Represents the data item submittal package or documentation that relates to the actual documents

Symbol	Object	Description
		submitted. The data analyst creates a submittal delivery schedule for the tasks.
✓	Submittal revision (**Cdm0Submittal Revision**)	Represents the specifications of the submittal.
	Correspondence (**Cdm0 Correspondence**)	Represents the feedback received from customers. The data analyst attaches offline feedback from customers to the correspondence object that is created in Teamcenter.
💬	Correspondence revision data (**Cdm0 Correspondence Revision**)	Represents the specifications of the correspondence.
	Tabular data table (**Fnd0StaticTable**)	This object is created to store the rows for tabular data.
	Event data table (**Cdm0Events Table**)	This object is created to store event data in the data requirement item (DRI) event table.

The following table describes the Aerospace and Defense solution object relation types.

Relationship	Description
ADS lists parts revisions relation (**ADS_Lists_PartsRevisions**)	Defines the relationship between a technical document revision and an ADS part or ADS design revision. An ADS part or an ADS design that is created on a given technical document is associated to the technical document revision through the **ADS_Lists_PartsRevisions** relation.
ADS lists parts relation (**ADS_Lists_Parts**)	Defines the relationship between a technical document revision and an ADS part or ADS design item. Items corresponding to an ADS part or ADS design are created on a technical document and are used on an assembly. They are associated with the technical document revision corresponding to the assembly through the **ADS_Lists_Parts** relation.
ADS lists drawing revisions relation (**ADS_Lists_DrawingRevisions**)	Defines the relationship between a technical document revision and an ADS drawing revision.

Relationship	Description
	When ADS drawings are created on a given technical document, they are associated with it through the **ADS_Lists_DrawingRevisions** relation.
Program preferred items relation (**TC_Program_Preferred_Items**)	Defines the relationship between a business object and a program. When business objects are set up as preferred parts to programs, they are associated to the programs through the **TC_Program_Preferred_Items** relation.
Parametric requirements lists (**ListStandardNotes**)	Defines the relationship between a business object and a standard note. A standard note or a standard note revision is attached to a business object through the **ListStandardNotes** relation.
Parametric requirements lists (**ListsCustomNotes**)	Defines the relationship between a business object and a custom note. A custom note or a custom note revision is attached to a business object through the **ListsCustomNotes** relation.
Described by data item description (**Cdm0IsDescribedbyDID**)	Defines the relation between the objects, **Cdm0DIDRevision** and **Cdm0DataReqItem Revision**. This will be used to support the requirement of creating data requirement item revision and associating it with contract revision.
List data requirement item submittal schedule (**Cdm0ListsDRISubmittalSch**)	Defines the relation between data requirement item revision and submittal delivery schedule.
Lists data requirement item submittal (**Cdm0ListsDRISubmittal**)	Defines the relation between the objects, **Cdm0DataReqItemRevision** and submittal.
List correspondence submittal (**Cdm0ListsCorspSubmittals**)	Defines the relation between the objects, **Cdm0SubmittalRevision** and **Cdm0CorpondenceRevision**.
Lists correspondence revision and contract revision (**Cdm0ListsCorspRefItems**)	Defines the relation between the objects, **Cdm0CorpondenceRevision** and **Cdm0ContractRevision**.
Lists contract data requirement item (**Cdm0ListsContractDRI**)	Defines the relation between the objects, **Cdm0ContractRevision** and **Cdm0DataReqItemRevision**.

2. Defining session settings

You can define the following user settings for your Teamcenter session:

- Group

  A user group is a set of users who have a common set of rules. You can use groups to consolidate rules. Rules that are common to multiple users can be written for a group instead of separately for each user. You can assign users to several groups.

  Groups are created by administrators in the Organization application. If you do not specify a group, the default group associated with the user account is used.

- Role

  You can write rules that apply to roles without regard to the project, then create role assignments that link users to the roles within each project. The software uses the role assignments to control access and messaging.

  Roles are created by administrators in the Organization application. If you do not specify a role, the default role associated with the user account is used.

- Current project or program

  If your administrator has specified a default program for you, that program is displayed when you log on to Teamcenter. This setting allows you to switch between the programs that you are a member of.

- Volume

  Specifies the repository for storing data.

- Commercial And Government Entity Code

  You must specify the Commercial And Government Entity (CAGE) code or location code of the business objects. CAGE code or location code helps to identify the place from where the business objects are sourced.

- Organization

  In the aerospace and defense industry, data must be created within the context of an organization. Organizations uniquely identify a group of users in an ISO 6523–compliant format. By default, the organization is set as the group you specified when you logged on to Teamcenter. If your log on group is not designated as an organization, or if the group does not belong to a parent organization, the **Organization** field is blank.

  - All data creation activities are linked to the organization specified for your user session. When you create an item, the owning organization of the item is the organization specified for the user session.

  - You can only specify an organization for your user session if your system is configured to require data to be created in the context of an organization.

  - Setting the organization preference enables you to create data in the context of an organization to which you do not belong.

  - Your default organization is set to the group you specified when you logged on to Teamcenter. If the group is not designated as an organization or if the group does not belong to an organization, the organization preference is blank.

- Application Logging

  Specifies the record of logging request to the server.

- Journalling

  Specifies the detail records of the application logging and the changes incorporated.

- Change notice

  Specifies the change notices that you have permission to write to.

  > Note:
  >
  > This field is visible only when you have the Change Manager application installed.

3. Using CAGE codes and company location codes

Register company location

> Note:
>
> You must first assign a location code to a company. The Aerospace and Defense application evaluates the **Fnd0MaintainUniqueLocationCode** global constant to verify the uniqueness of the location code.

1. Open My Teamcenter.

2. From the menu bar, choose**File→New→Other**.

3. In the **New Business Object** dialog box, expand **Complete List**.

4. Select **Company Location** and click **Next** .

5. Define information for the company location, as follows.

Name	Specifies the name of the company location.
CAGE Code	Specifies the position code assigned by the Government. This code is unique for each company location.
Location Type	Specifies the type of location code, for example, CAGE (Commercial And Government Entity), GLN (Global Location Number), and so on.
Street	Specifies the street name where the company is located.
City	Specifies the city where the company is located.
State / Province	Specifies the state or province where the company is located.
Postal Code	Specifies the ZIP code.
Country	Specifies the country where the company is located.
Region	Specifies the region where the company is located.
URL	Specifies the company URL.

Description	Describes the company. Use a description that helps to distinguish the purpose of the company and the nature of its business.
Name	Specifies the name of the company.

6. Click **Finish**.

Edit company location

1. In the **Quick Links** pane, click **Home**.

2. In the **Home** pane, select the company location object that you have created.

3. In the **Application** pane, click the ✎ icon.

4. (Optional) In the **Check-Out** dialog box, define the following and then click **Yes** to continue.

Change ID	Specifies the ID of the changed company location.
Comments	Specifies comments about the change ID.

5. In the **Application** pane, select the **Viewer** tab to edit the company location information.

   For more information about the description of the fields, see Register company location.

6. Click **Finish**.

Add company code to an item

The location or Commercial And Government Entity (CAGE) code that your administrator has configured for your particular user profile, is automatically added as the original CAGE code to an item and as the current CAGE code to the item revision, when you create one.

To modify the CAGE code assigned to you:

1. Choose **Edit** → **User Setting**.

   > Tip:
   >
   > You can also click the user information link in the client window application header to access the **User Settings** dialog box.

2. In the **User Settings** dialog box, select an appropriate location code from the **CAGE Code** list.

Note:

You will only see those codes in the **CAGE Code** list that your administrator has assigned to your user group.

When you revise an item or an item revision, or perform the **Save As** or **Baseline** operations, the location code for the item displays the current location code that you set in your session.

4. Working with Aerospace and Defense objects

Creating Aerospace and Defense objects

Create a technical document

1. Open My Teamcenter.

2. Choose **File→New→Item**.

3. In the **New Item** dialog box, select **Technical Document**.

4. Define the basic information for the new item, as follows:

   In the **Naming and Numbering Pattern** section:

   ID

   Specifies the naming and the numbering pattern for the part.

   > Note:
   >
   > The naming and the numbering rules are set in Business Modeler IDE.

   Revision

   Specifies the revision pattern of the part.

   In the **Item Information** section:

   ID/Revision - Name

   Specifies the ID, revision ID, and name of the part. This is a mandatory field.

   Click the **Assign** button to automatically generate the item ID and revision identifiers.

   The **Assign** button is active only if naming rules and automatic generation have been implemented for the selected object type.

	The ID is based on the numbering pattern you selected in the **Naming and Numbering Pattern** section.
Description	Describes the part. Use a description that helps to distinguish the purpose of the part.
Unit of measure	Specifies the metric of the part.

5. Click **Next** .

6. Define additional item information, as follows:

Category	A user defined attribute for classifying the technical document.
Original CAGE Code	Specifies the location code assigned to an item when it was first created. Note: This box is not available by default.
Technical Document category	Specifies the category of the technical document, for example, assembly drawing, wire list, material specification, and so on.
Title	Specifies the title of the technical document.
Author	Specifies the author or the originator of the technical document. The author or the originator of a document is the person, office, or designated position responsible for its creation or issuance.
Subject	Specifies the subject of the technical document.
Keywords	Specifies a word or phrase used for indexing or cataloging the technical document.

7. Click **Next**.

8. (Optional) Define additional item revision information, as follows:

Custodial organization Name	Specifies the name of the organization that has the custody of the technical documents.

Note:

The **Custodial organization Name** is saved only if you specify the **Organization ID** of the group or organization.

You can set the **Organization ID** in the **ADA/ITAR Attributes** tab of the Group in the Organization application.

Design Level

Specifies the conceptual or developmental design, production prototype or limited production design, and engineering drawings required for the production of an item or a system.

Document Author

Specifies the author or the originator of the technical document. The author or the originator of a document is the person, office or designated position responsible for its creation or issuance.

Document Subject

Specifies the theme of the technical document.

Document Title

Specifies the heading of the technical document.

Undimensioned

Specifies whether a drawing defines the undimensioned flat patterns for parts.

Title

Specifies the title of the technical document.

Author

Specifies the author or the originator of the technical document. The author or the originator of a document is the person, office, or designated position responsible for its creation or issuance.

Subject

Specifies the subject of the technical document.

Keywords

Specifies a word or phrase used for indexing or cataloging the technical document.

9. Click **Next** to create an alternate identifier, if enabled.

10. (Optional) Select the program that you want to assign from the list of programs.

    Note:

    You can assign objects to programs only if you are a privileged program team member.

11. Select one or more programs from the **Programs for Selection** list.

12. Click the right-arrow button» to move the programs to the **Selected Programs** list.

13. Click **Next**.

14. (Optional) Define the display options to create a new Aerospace and Defense technical document in the selected folder in My Teamcenter, as follows:

    Show as a new root

    Specifies that the newly created object is opened as a root object. It is not pasted to the selected item. If you do not select this option, the new item is pasted as the child of the selected item.

    Use item identifier as default display

    Specifies item identifier as a default display object.

    Use revision identifier as default display

    Specifies that the alternate identifier for the item is the default display object.

    Check-Out Item Revision on Create

    Specifies that the item revision is checked out automatically after creation.

    > Note:
    >
    > You must check out the item revision before you modify it.

15. Click **Finish**.

Create a part or a commercial part

1. Open My Teamcenter.

2. Choose **File→New→Part**.

3. In the **New Part** dialog box:

    a. Choose **Part** for creating a part.

    b. Choose **Commercial Part** for creating a commercial part.

4. Click **Next**.

5. Define basic information for a part, as follows:

    In the **Naming and Numbering Pattern** section:

ID

Specifies the naming and the numbering pattern for the part.

> Note:
>
> The naming and the numbering rules are set in Business Modeler IDE.

Revision

Specifies the revision pattern of the part.

In the **Part Information** section:

ID/Revision - Name

Specifies the ID, revision ID, and name of the part. This is a mandatory field.

> Note:
>
> Click the **Assign** button to automatically generate the item ID and revision identifiers.
>
> The **Assign** button is active only if naming rules and automatic generation have been implemented for the selected object type.
>
> The ID is based on the numbering pattern you selected in the **Naming and Numbering Pattern** section.

Description

Describes the part. Use a description that helps to distinguish the purpose of the part.

Unit of measure

Specifies the metric of the part.

6. Click **Next**.

   The following happens when you click **Next**, **Back** or the **Finish** button:

   If you have enabled automatic selection and multifield keys for technical documents and if there are multiple technical documents matching the criterion:

   - You see the select **Unique Source Document** dialog box. This dialog box displays the list of technical documents matching the criteria you specified for technical documents.

   - To select a technical document, double-click the row that contains the technical document you want.

   - You can change the technical document in the **Define additional part information** pane.

7. Define additional part information, as follows:

Original CAGE Code

Specifies the location code assigned to an item when it was first created.

> Note:
>
> This box is not available by default.

Part Category

Specifies the category of part for the program, for example, supplier part, government furnished equipment, or military standard part.

Source Document category

Specifies the user-defined attribute for classifying document items.

Source Document ID

Specifies the unique identifier of the document associated with the part.

Source Document Revision

Specifies the revision of the document associated with the part.

Source Technical Document Category

Specifies the category of the technical document, for example, assembly drawing, wire list, material specification, and so on.

Design Required

Specifies whether a CAD design is required for the part. The Aerospace and Defense solution validates the designs for the part. This property is usually selected for consumables that have a representation in the parts list but no geometric representation. For example, glue, paint, grease, and so on.

> Note:
>
> You must specify the source document details and the attribute values before you create the part in document-centric programs.
>
> Enter the values for source document properties in the language specified by the master locale. You can copy master locale values from the respective source document properties using the **Localization** button 🌐.

8. Click **Next**.

The following happens when you click **Next**, **Back** or the **Finish** button and you have configured multifield keys for technical documents.

If there are multiple technical documents matching the criteria for technical documents you specified in the **Define additional part information** dialog box:

- You see the select **Unique Source Document** dialog box. This dialog box displays the list of technical documents matching the criteria you specified for technical documents.

- To select a technical document, double-click the row that contains the technical document you want.

9. (Optional) Define additional part revision information, as follows:

Make/Buy	Specifies whether the part is manufactured internally or sourced from a vendor.

10. Click **Next** to create an alternate identifier, if enabled.

11. (Optional) Select the program that you want to assign from the list of programs.

> Note:
>
> You can assign objects to programs only if you are a privileged program team member.

12. Select one or more programs from the **Programs for Selection** list.

13. Click the right-arrow button » to move the programs to the **Selected Programs** list.

14. Click **Next**.

15. (Optional) Define the display options to create a new Aerospace and Defense technical document in the selected folder in My Teamcenter, as follows:

Show as a new root	Specifies that the newly created object is opened as a root object. It is not pasted to the selected item. If you do not select this option, the new item is pasted as the child of the selected item.
Use item identifier as default display	Specifies the item identifier as a default display object.
Use revision identifier as default display	Specifies that the alternate identifier for the item is the default display object.
Check-Out Item Revision on Create	Specifies that the item revision is checked out automatically after creation. Note: You must check out the item revision before you modify it.

16. Click **Finish**.

Create a design

1. Open My Teamcenter.

2. Choose **File→New→Design**.

3. In the **New Design** dialog box, select **Design**.

4. Define basic information for a new design, as follows:

   In the **Naming and Numbering Pattern** section:

ID	Specifies the naming and the numbering pattern for the part. Note: The naming and the numbering rules are set in Business Modeler IDE.
Revision	Specifies the revision pattern of the part.

   In the **Design Information** section:

ID/Revision - Name	Specifies the ID, revision ID, and name of the part. This is a mandatory field. Click the **Assign** button to automatically generate the item ID and revision identifiers. The **Assign** button is active only if naming rules and automatic generation have been implemented for the selected object type. The ID is based on the numbering pattern you selected in the **Naming and Numbering Pattern** section.
Description	Describes the part. Use a description that helps to distinguish the purpose of the part.
Unit of measure	Specifies the metric of the part.

5. Click **Next** .

   The following happens when you click **Next**, **Back** or the **Finish** button:

If you have enabled automatic selection and multifield keys for technical documents and if there are multiple technical documents matching the criterion:

- You see the select **Unique Source Document** dialog box. This dialog box displays the list of technical documents matching the criteria you specified for technical documents.
- To select a technical document, double-click the row that contains the technical document you want.
- You can change the technical document in the **Define additional part information** pane.

6. Define the additional design information, as follows:

Design Category

Specifies the category of part for the program, for example, supplier part, government furnished equipment, or military standard part.

Original CAGE Code

Specifies the location code assigned to an item when it was first created.

> Note:
>
> This box is not available by default.

Source Document category

Specifies the user-defined attribute for classifying document items.

Source Document ID

Specifies the unique identifier of the document associated with the design.

Source Document Revision

Specifies the revision of the source document that must be used to create the design.

Source Technical Document Category

Specifies the category of the technical document, for example, assembly drawing, wire list, material specification, and so on.

> Note:
>
> You must specify the source document details and the attribute values before you create the design in document-centric programs.
>
> Enter the values for source document properties in the language specified by the master locale. You can copy master locale values from the respective source document properties using the **Localization** button 🌐.

7. Click **Next**.

The following happens when you click **Next**, **Back** or the **Finish** button and you have configured multifield keys for technical documents.

If there are multiple technical documents matching the criteria for technical documents you specified in the **Define additional part information** dialog box:

- You see the select **Unique Source Document** dialog box. This dialog box displays the list of technical documents matching the criteria you specified for technical documents.
- To select a technical document, double-click the row that contains the technical document you want.

8. (Optional) Define additional design revision information, as follows:

Make/Buy	Specifies whether the part is manufactured internally or sourced from a vendor.

9. Click **Next** to create an alternate identifier, if enabled.

10. (Optional) Select the program that you want to assign from the list of programs.

> Note:
>
> You can assign objects to programs only if you are a privileged program team member.

11. Select one or more programs from the **Programs for Selection** list.

12. Click the right-arrow button » to move the programs to the **Selected Programs** list.

13. Click **Next**.

14. (Optional) Define the display options to create a new Aerospace and Defense technical document in the selected folder in My Teamcenter, as follows:

Show as a new root	Specifies that the newly created object is opened as a root object. It is not pasted to the selected item. If you do not select this option, the new item is pasted as the child of the selected item.
Use item identifier as default display	Specifies item identifier as a default display object.
Use revision identifier as default display	Specifies that the alternate identifier for the item is the default display object.
Check-Out Item Revision on Create	Specifies that the item revision is checked out automatically after creation.

> Note:
>
> You must check out the item revision before you modify it.

15. Click **Finish**.

Create a drawing

While creating a drawing, you must mention the source document number. If there are multiple source documents available with the same number, provide additional information, such as source document category, and source technical document category.

1. Open My Teamcenter.

2. Choose **File→New→Drawing**.

3. Define basic information for the new item, as follows:

   In the **Naming and Numbering Pattern** section:

   ID

   Specifies the naming and the numbering pattern for the part.

   > Note:
   >
   > The naming and the numbering rules are set in Business Modeler IDE.

   Revision

   Specifies the revision pattern of the part.

   In the **Item Information** section:

   ID/Revision - Name

   Specifies the ID, revision ID, and name of the part. This is a mandatory field.

   Click the **Assign** button to automatically generate the item ID and revision identifiers.

   The **Assign** button is active only if naming rules and automatic generation have been implemented for the selected object type.

	The ID is based on the numbering pattern you selected in the **Naming and Numbering Pattern** section.
Description	Describes the part. Use a description that helps to distinguish the purpose of the part.
Unit of measure	Specifies the metric of the part.

4. Click **Next** .

   The following happens when you click **Next**, **Back** or the **Finish** button:

   If you have enabled automatic selection and multifield keys for technical documents and if there are multiple technical documents matching the criterion:

   - You see the select **Unique Source Document** dialog box. This dialog box displays the list of technical documents matching the criteria you specified for technical documents.
   - To select a technical document, double-click the row that contains the technical document you want.
   - You can change the technical document in the **Define additional part information** pane.

5. Define the additional item information, as follows:

Original CAGE Code	Specifies the location code assigned to an item when it was first created. Note: This box is not available by default.
Source Document category	Specifies the user-defined attribute for classifying document items.
Source Document ID	Specifies the ID of the source document.
Source Document Revision	Specifies the revision of the source document that must be used to create the drawing.
Source Technical Document Category	Specifies the category of the technical document, for example, assembly drawing, wire list, material specification, and so on.

> Note:
>
> In document-centric programs, you must specify the source document details and the attribute values before you create the drawing.
>
> Values for source document properties must be entered in the language specified by the master locale. You can copy master locale values from the respective source document properties using the **Localization** button 🌐.

6. Click **Next**.

   The following happens when you click **Next**, **Back** or the **Finish** button and you have configured multifield keys for technical documents.

   If there are multiple technical documents matching the criteria for technical documents you specified in the **Define additional part information** dialog box:

   - You see the select **Unique Source Document** dialog box. This dialog box displays the list of technical documents matching the criteria you specified for technical documents.

   - To select a technical document, double-click the row that contains the technical document you want.

7. (Optional) Define additional item revision information, as follows:

   Drawing Type

   Specifies the format of the drawing. For example, PDF, Microsoft Word, and so on.

   Drawing can be manufacturing drawing, schematic drawing, and so on.

8. Click **Next** to create an alternate identifier, if enabled.

9. (Optional) Select the program that you want to assign from the list of programs.

   > Note:
   >
   > You can assign objects to programs only if you are a privileged program team member.

10. Select one or more programs from the **Programs for Selection** list.

11. Click the right-arrow button, (») to move the programs to the **Selected Programs** list.

12. Click **Next**.

13. (Optional) Define the display options to create a new Aerospace and Defense technical document in the selected folder in My Teamcenter, as follows:

Show as a new root	Specifies that the newly created object is opened as a root object. It is not pasted to the selected item. If you do not select this option, the new item is pasted as the child of the selected item.
Use item identifier as default display	Specifies item identifier as a default display object.
Use revision identifier as default display	Specifies that the alternate identifier for the item is the default display object.
Check-Out Item Revision on Create	Specifies that the item revision is checked out automatically after creation. Note: You must check out the item revision before you modify it.

14. Click **Finish**.

Creating Aerospace and Defense business objects in document-centric programs

Document-centric programs require you to specify a source document while creating Aerospace and Defense business objects such as parts, designs, and drawings.

When creating a business object in a document-centric program, you must fill the **Source Document** property of the business object with the document number of an existing document.

In document-centric programs, when a part is created, the part is associated with the technical document through the **ADS_Lists_PartRevisions** relation. When a drawing is created, the drawing is associated with the technical document through the **ADS_Lists_DrawingRevisions** relation.

A single source document can list multiple parts. A document-centric program manages parts list documents that are revised independently of the part.

The behavior of the source technical document varies during creation of Aerospace and Defense business objects in a document-centric program when:

- Automatic selection of source technical documents is enabled and automatic creation of source technical documents is disabled.

If you do not specify the source document ID, the system extracts the document ID from the item ID.

However, if you specify both the source document ID and source document details, the system uses the details provided in the **Enter additional part/design information** pane to select the source technical document.

> Note:
>
> The string before the last hyphen in the item ID is the ID of the source technical document.
>
> If two or more technical documents exist with the same ID, you must provide specific details about the technical document by entering the source document category, and revision ID.
>
> The system displays an error message if:
>
> - The source document details are not provided.
>
> - The details provided do not correspond to an existing technical document.

- Automatic creation of source technical document is enabled and automatic selection of source technical document is disabled.

  If you provide the source document details, such as the source document category, and source document revision when you create a part, design, or drawing, the system automatically creates a new technical document corresponding to the source document details and associates it with the new business object through an appropriate relation.

  Automatic creation of technical document works only after the system ensures that:

  - An existing document cannot be selected from the source document ID provided as a part of the part, design or drawing ID in the **Enter part/design/drawing information** pane because automatic selection is disabled.

  - The details provided in the **Enter additional part/design/drawing information** pane does not match any of the existing technical documents.

- Both automatic selection and automatic creation of source technical document are enabled.

  When you specify the source document ID while creating Aerospace and Defense business objects, the system automatically selects the technical document corresponding to the source document ID. You do not need to manually specify the source document details. The new business object is associated to the automatically selected technical document through an appropriate relation.

  However, if you specify both the source document ID and source document details, the system uses the details provided to select the source technical document.

If the given source document details do not match any of the existing technical documents, the system automatically creates a new technical document based on the details you provided when creating the object and associates it with the new business object through an appropriate relation.

> Note:
>
> If you do not specify the document ID, Teamcenter creates the document ID based on the ID of the part, design, or drawing.

- Both automatic selection and automatic creation are disabled.

  When you specify the source document details, the system searches for the corresponding source document and updates the other attributes, such as the source document category, and source document revision. The selected technical document is associated to the new business object through an appropriate relation.

  When you specify details about the source document and the details do not match any of the existing source documents, the system fails to create the business object because automatic creation of source technical document is disabled and displays an error message.

Revising Aerospace and Defense objects

Revising parts and documents

In a document-centric program, items or item revisions are associated with each other through appropriate relations. Revision behavior of associated parts and documents can be specified by configuring the deep copy rules for the relations in the Business Modeler IDE.

Document revisions and part revisions are associated through the **ADS_Lists_PartRevisions** relation. Consider the following revision behaviors before revising associated parts and documents:

- Both a part and its associated source technical document can be revised independently of each other.

- A technical document item revision whose item is associated to a part or part revision can be revised independently of the part.

- When revising a part revision that has an associated technical document item, the relation with the previous revision of the part is deleted and only the new revision of the part is carried forward.

- When revising a technical document item revision that has associated technical document items, the associations are carried forward to the new revision.

Revise Aerospace and Defense objects

1. Open My Teamcenter.

2. Select the Aerospace and Defense business object item revision that you want to revise and choose **File→Revise**.

   The **Revise** dialog box appears.

3. In the **Define the basic information for the new item revision** pane, type the name of the new revision.

   The system generates a new revision ID based on the existing one.

4. Click **Next** or click the **Enter Additional Item Revision Information** link in the left pane of the **Revise** dialog box.

   The **Define additional item revision information** pane appears.

5. (Optional) Type the title, author, subject, and keywords in the **Define additional item information** pane.

6. Click **Next** or click the **Define Attached Objects** link in the left pane of the **Revise** dialog box.

   The **Define attached objects** pane displays the source revision and attachments and the destination revision and attachments. By default, all objects are copied forward to the new (destination) revision.

7. (Optional) In the left pane of the **Revise** dialog box, click the **Assign to Programs** link.

8. (Optional) Assign the new revision to a different project or program.

9. (Optional) Click the **Define Options** link and select the appropriate open, display, and checkout options.

10. Click **Finish**.

> Note:
>
> When a business object that must be associated with a change notice is revised, all revision activities on the business object are associated with the change notice that is set as the preferred change notice for your user session.

Modify Aerospace and Defense objects

1. Open My Teamcenter.

2. Select the Aerospace and Defense object that you want to modify and choose **View→Properties** or right-click the object and choose **Properties**.

The **Properties** dialog box appears.

3. Click the **Check out and Edit** button.

   The **Check Out** dialog box appears.

4. In the **Change ID** box, type a change ID.

5. (Optional) Type comments about the change in the **Comment** box.

6. Click **Yes** to confirm checkout.

   The **Edit Properties** dialog box appears.

7. Type the values in the **Object**, **Name**, and **Description** fields.

8. Verify the **Owner**, **Group ID**, and **Last modified user information** and click **Save**.

9. Click **Check-In** to check in the modified item.

10. Click **Yes** to confirm the check in operation.

The existing Aerospace and Defense object is modified.

Delete Aerospace and Defense objects

Note:

You cannot delete a business object if it is referenced by other objects. For example, if an ADS part is associated with a source technical document through the **ADS_Lists_PartsRevisions** relation in a document-centric program, it cannot be deleted. You must first select the part revision displayed under the **ADS_Lists_PartsRevisions** relation and click the ✂ button on the toolbar to remove the part from the relation. The part is now independent of the technical document and can be deleted.

1. Open My Teamcenter.

2. Select the ADS item business object that you want to delete.

3. Choose **Edit→Delete**, or click ❌ .

   The **Confirmation** dialog box appears.

4. Click **Yes** to confirm deletion.

Working with preferred standard parts

Add a preferred standard part to a program

To add a preferred standard part to a program:

1. Configure an item business object as a preferred standard part for the program.

   > Note:
   >
   > You must be an administrator or a user with DBA privileges to enable an item business object as a preferred standard part.

2. Create the item business object that has been configured as a preferred standard part.

3. Associate the preferred standard part with the program.

Associate preferred standard parts to programs

1. Open My Teamcenter.

2. Select the standard part and choose **Edit→Copy**.

3. Select the program to which you want to attach the preferred standard part.

   > Note:
   >
   > You can search for a program by clicking the **Search** button 🔍 on the **My Teamcenter** toolbar.

4. Expand the program tree in the **Search Result** pane and navigate to the **TC_Program_Preferred_Items** relation.

5. Right-click the **TC_Program_Preferred_Items** relation and click **Paste**.

   The preferred standard part is attached to the program.

Remove preferred parts from assemblies

1. Open My Teamcenter.

2. Select the preferred part.

> Note:
>
> You can search for the preferred part by clicking the **Search** button 🔍 on the toolbar.

3. Double-click the preferred part to open it in the **Impact Analysis** view.

   The **Impact Analysis** pane displays the assemblies that use the selected preferred part.

4. Right-click the assembly and choose **Send to→Structure Manager**.

   The assembly appears in the **Structure Manager** window.

5. Select the preferred part and click the **Remove a Line** ▦ button on the toolbar.

> Note:
>
> Ensure that you remove the preferred part from all the assemblies that use it.

6. Choose **File→Save** to save the changes in the **Structure Manager** window.

Remove preferred parts from programs

1. Open My Teamcenter.

2. Select the program that uses the preferred part.

> Note:
>
> You can search for the program by clicking the **Search** button 🔍 on the toolbar.

3. Expand the program tree and navigate to the **Tc_Program_Preferred_Items** relation.

4. Select the preferred part that appears under the **Tc_Program_Preferred_Items** relation and click ✂ or choose **Edit→Cut**.

> Note:
>
> Ensure that you cut the relation in all the programs that use the preferred part.

Change the owning program

When changing owning programs, consider the following:

- To change owning programs, you must be a member of both: the current program and the new owning program.
- You can change the owning program only for objects of the type **Item**. You cannot change the owning program for objects of the type **Dataset**.
- The owning program must be a *privileged* program.
- The program will be set as the owning program for all other related objects that are specified in the program propagation rule. This rule takes care of assigning other related objects to the program.

To change the owning program of an object:

1. In My Teamcenter or Structure Manager, select the objects, right-click, and choose the **Change Owning Program** command.

   Alternatively, choose **Tools**→**Program**→**Change Owning Program**.

2. In the **Change Owning Program** dialog box, select the new program from the **Select Program to Assign** list and click **OK**.

Assemblies in Aerospace and Defense

Assemblies for document-centric programs

You can add parts, components, and other assemblies to build a new assembly in your current (active) program. In document-centric programs, assemblies can consist of:

- ADS parts or designs belonging to the same source document as that of the assembly.
- Foreign components. ADS parts or designs that belong to a different source document but not to the assembly's source document are considered to be foreign components.

  The foreign components are related to the assembly's source technical document through the **ADS_Lists_Parts** relation.

The Aerospace and Defense solution template helps you build an assembly by:

- Automatically identifying foreign components when they are added to the assembly and allowing them to be associated with the assembly's source technical document revision through the **ADS_Lists_Parts** relation.
- Allowing a single component to be associated with multiple assemblies belonging to the same source document through a single relation (**ADS_Lists_Parts**). You can remove the part or design from one assembly without having to remove the relation that exists until you remove the part from all the assemblies.

Build an assembly for a document-centric program

> Note:
>
> An assembly can include multiple parts. You can copy the parts that you want to include in the new assembly from the business objects list in My Teamcenter and paste them in the assembly in the Structure Manager.
>
> You must paste all the components on the first component that was sent to the Structure Manager and not on any of the components pasted under the first component in the assembly.
>
> The system automatically identifies the first component's source technical document as the assembly's source technical document and associates the foreign components to the source technical document through the **ADS_Lists_Parts** relation.

1. Open My Teamcenter.

2. Right-click the first component to be added to the assembly and choose **Send to→Structure Manager**.

   The Structure Manager opens and displays the first component. This component is considered to be the parent component in the assembly.

3. Switch to My Teamcenter, select the second component that you want to include in the new assembly under the parent component, and choose **Edit→Copy**.

4. Switch to Structure Manager, select the parent component of the assembly, and choose **Edit→Paste** or right-click the parent component and choose **Paste**.

   The second component is added to the assembly in the Structure Manager.

5. Choose **File→Save** to save the changes in the **Structure Manager** window.

   The new assembly is saved as BOM view revisions in the source technical document in My Teamcenter.

5. Generate part-centric parts list reports

1. Open My Teamcenter.

2. Right-click the assembly for which you want to generate the part-centric parts list report and choose **Send To→Structure Manager**.

3. (Optional) Configure the revision rule and effectivity for the assembly, as required.

4. Open the Structure Manager from the left pane of the **Favorites** view in My Teamcenter.

   The selected assembly appears in the BOM line view in the Structure Manager.

5. In the BOM line view, select the assembly.

6. Choose **File→Print** to generate the parts list report. Select the output contents for the report.

   The report appears in the report pane:

   - The assembly from which the report is generated.

   - The parts in the first level of the product structure.

   - The creation date of the parts list report.

   - The part number, revision, and sequence of the assembly that the part-centric parts list report is created for.

   The following details are also available for each component and subassembly:

   - Find number

   - Part number

   - Nomenclature

   - Quantity

   - Remark

   The report is available in plain text, XML, and HTML.

6. Working with notes

Working with standard notes

Basic concepts about standard notes

Standard notes represent the notes established by the Standards Engineering group that conform to the design practices of a program. Standard notes are often categorized and maintained in a library for use by design engineers in the creation of parts lists.

The following concepts apply to standard notes:

- Standard notes specify parametric variable values that designers select when attaching the note to an item or an item revision.

- Standard notes are associated with items, such as parts or documents, through the **Standard Notes Lists** relation where the part or document revision is the primary object and the standard note or standard note revision is the secondary object.

- The parameters for the note are specified in the **Note Text** property of the note revision.

  Note:

  Text cannot be modified after it is saved. You must create a new revision of the standard note and make changes to the **Note Text** property of the new revision.

- The text of the note revision contains text, parameters, and applicable values using the syntax shown below.

  text [*parameter name*: *parametric value1* *delimiter* *parametric value2 delimiter*... *parametric value n*]

  Example:

  ```
  Round all sharp corners to [Radius: 0.4, 0.5, 0.6] inch
  and heat treat to [Temperature: 200, 220, 230] degrees Fahrenheit.
  ```

- The default delimiter for parameters in the note text is a comma (,). However, your Teamcenter administrator can configure the delimiter to use a different character.

- When a standard note is created, a **Standard Note Text** dataset is created that contains the text. The dataset is attached to the standard note revision.

Note:

You cannot delete the dataset or detach the dataset from the standard note revision.

- Standard notes are used to associate parts with documents.
- Standard notes and standard note revisions can be related to multiple items or item revisions.
- You can export parameters and parametric values of standard notes using the **Export to Excel**or tcxml export feature only.
- You can export parameters and parametric values of standard notes using the **Export to Excel** or tcxml export feature only.
- Only a single revision of a standard note can be attached to an item or item revision. By default, you cannot attach multiple revisions of the same note to an item or item revision.

  Note:

  Your Teamcenter administrator can configure the system to allow multiple revisions of a standard note to be attached to an item or item revision by setting the value of the **AllowMultipleRevisionsofStdNotes** global constant to **true**.

Create a standard note

Note:

To create standard notes, you must have **DBA** privileges with administrative bypass set, or you must fill a role that your Teamcenter administrator has designated as having standard note creation privileges.

1. Open My Teamcenter.
2. Choose **File→New→Item**.

   The **New Item** dialog box appears.
3. In the **New Item** dialog box, select the **Standard Note** item type.
4. Click **Next**.

   The system displays the **Define basic information for new item** pane.
5. Type an item ID, revision ID, and name for the standard note, or click **Assign** to automatically generate the item ID and revision identifiers.

> Note:
>
> The **Assign** button is active only if naming rules and automatic generation are implemented.

6. (Optional) Type a description of the standard note and select a unit of measure.

7. Click **Next**.

   The **Define additional item information** pane appears.

8. In the **Note Category** box, choose a category for the standard note.

9. (Optional) Enter values for the note attributes.

10. Click **Next**.

    The **Define additional item revision information** pane appears.

11. (Optional) Enter values for the revision attributes.

12. In the left pane of the **New Item** dialog box, click the **Assign to Programs** link.

    The system displays the list of programs to which the standard note can be assigned.

    > Note:
    >
    > You can assign objects to programs only if you are a privileged program team member.

13. (Optional) Select one or more programs from the **Programs for Selection** list and move them to the **Selected Program** list.

14. (Optional) Click **Next** or click the **Define Options** link in the left pane of the **New Item** dialog box.

15. (Optional) Select open, display, and checkout options.

16. Click **Finish**.

    A new standard note is created in the selected folder in My Teamcenter. It consists of a:

    - Standard note master form

    - Standard note revision

    - Standard note revision master form

- Standard note text dataset

Specify standard note text

> Note:
>
> To specify standard note text, you must have **DBA** privileges with administrative bypass set, or you must fill a role that your Teamcenter administrator has designated as having standard note definition privileges.

1. Open My Teamcenter.

2. Right-click the standard note revision and choose **Properties**. Alternatively, select the standard note revision and choose **View → Properties**.

   > Note:
   >
   > You cannot double-click the note text dataset to open it. Although the note text is stored in the dataset, you must enter the data in the note revision properties.

   The **Properties** dialog box appears.

3. Click the **Check-Out and Edit** button.

   The **Check-Out** dialog box appears.

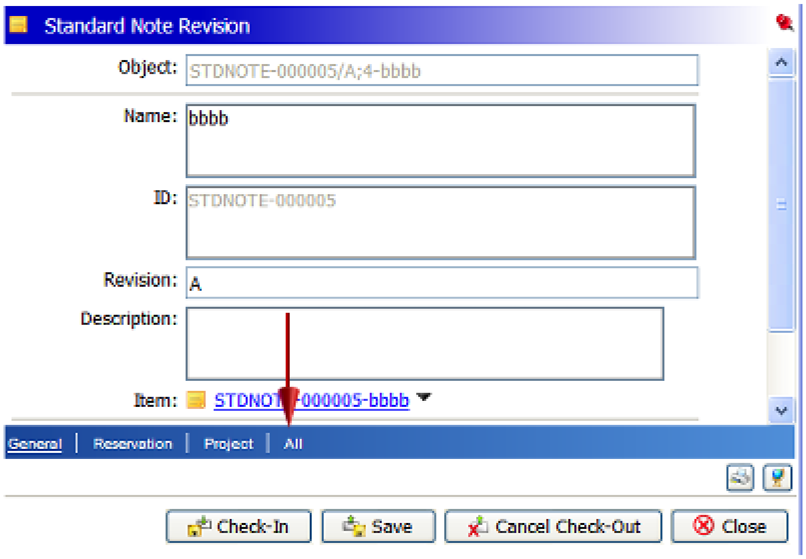
4. In the **Change ID** box, type the change ID.

5. In the **Comments** box, type notes for checkout.

6. Click **Yes**.

   The **Edit Properties** dialog box appears.

7. Click the **All** link in the **Edit Properties** dialog box, and scroll to the **Note Text** box.



Standard Note Revision

Object: STDNOTE-000005/A;4-bbbb

Name: bbbb

ID: STDNOTE-000005

Revision: A

Description:

Item: STDNOT -000005-bbbb

8. In the **Note Text** box, type the note using the following syntax:

   text [*parameter name*: *parametric value1 delimiter parametric value2 delimiter* ..... *parametric value n*]

Example:

```
Round all sharp corners to [Radius: 0.01, 0.02, 0.03] inches
and heat treat to [Temperature: 100, 120, 130] degrees Fahrenheit.
```

Note:

The delimiter used when specifying parametric values is defined by your Teamcenter administrator. The default delimiter is a comma (**,**).

9. Click **Save**.

Note:

You cannot modify the text after it has been saved. To update parameters, you must create a new revision of the standard note.

10. Click the **Check-In** button.

    The **Check-In** dialog box appears.

11. Click **Yes**.

Attach a standard note to an item or item revision

> Note:
>
> You do not need **DBA** privileges or special role access to attach a standard note to an item or item revision.

1. Open My Teamcenter.

2. Right-click the standard note or the standard note revision that you want to attach and choose **Copy**. Alternatively, select the standard note and choose **Edit → Copy**.

3. Select the destination item or item revision and choose **Edit → Attach Requirements/Notes → Parametric Requirement**.

   The **Input Parametric Values** dialog box displays the parameters and values defined for the note.

4. Choose values for the parameters.

5. (Optional) Select the **Flag Note** option.

6. Click **OK**.

   The standard note revision is attached to the selected object (item or item revision) through the **Standard Notes Lists** relation.

If the note is not visible under the item or item revision to which it is attached, you must specify **Standard Notes Lists** as a shown relation for the item and item revision. To do this:

1. Choose **Edit→Options**.

2. Expand the **Options** tree and expand the **General** folder.

3. Click the **Item** or **Item Revision** node.

4. In the **General** options tab, choose **Standard Notes Lists** from the **Available Relations** list.

5. Click + to move it to the **Shown Relations** list.

6. Click **OK**.

View parametric values in standard notes

To view the parametric values in a standard note which is attached to an item or item revision:

1. In My Teamcenter, select the standard note revision and choose **Window→Show View→Other**.

2. In the **Show View** dialog box, expand the **Teamcenter** navigation tree and select **MS Word**.

3. Click **OK**.

   You can see the parametric values of the standard note in a new view.

Revise a standard note

To modify the data associated with a standard note, you must first create a new revision of the standard note.

1. Open My Teamcenter.

2. Select the standard note revision that you want to revise and choose **File→Revise**.

   The **Revise** dialog box appears.

3. (Optional) In the **Define the basic information for the new item revision** pane, type the name of the new revision.

4. (Optional) Type a description of the revision and choose a unit of measure for the revision.

5. Click **Next**.

   The **Define additional item revision information** pane is displayed.

6. (Optional) Enter values for the note revision attributes.

7. Click **Next**.

   The **Define attached objects** pane displays the source revision and attachments and the destination revision and attachments of the destination revision. By default, all objects are copied forward to the new (destination) revision.

8. (Optional) Choose copy options for the objects related to the source revision by clicking the icon to the right of the object in the tree.

   ⊘ **Don't Copy**

   Allows you to selectively copy objects from one revision to another. Objects that are designated as **Don't Copy** appear with a line through them in the **Destination** tree.

   🗋 **Copy as Object**

Creates a new object of the same type and relation to the parent object. Objects created by this method are totally independent of the parent object. Therefore, any changes to the copied object are not reflected in the original object.

Copied objects appear in bold in the **Destination** tree and can be renamed.

Copy as Reference

Copies the object as a reference to the original object. All changes to the reference copy affect the original object. The names of attachments that are copied as references cannot be modified.

9. Click **Next**.

   The **Select projects or programs that you want to assign new object to** pane is displayed.

10. (Optional) Assign the new revision to a different project or program.

11. Click **Next**.

    The **Select open option and alternate id display option** pane is displayed.

12. (Optional) Set the open, display, or checkout option for the new revision.

13. Click **Finish**.

    The new revision is created as a copy of the previous revision object. The new revision is independent of the source revision. Modifications to the new revision are not reflected in the source revision, which allows you to edit the note text for each revision.

14. Specify the note data.

Replace a standard note revision with another revision

By default, only one revision of a standard note can be associated with an item or item revision. However, you can replace an existing note with a new revision of the note.

1. Open My Teamcenter.

2. Right-click the revision of the standard note that will replace the revision attached to the item or item revision, and choose **Copy**.

3. Expand the item or item revision and select the standard note revision that you want to replace.

4. Choose **Edit→ Replace ParametricRequirement Revision**.

If there are parametric values defined for the note, the **Input Parametric Values** dialog box appears.

5. Choose values for the parameters.

6. (Optional) Select the **Flag Note** option.

7. Click **OK**.

   The standard note revision attached to the item or item revision is replaced by the new revision of the note.

Validate whether a note is qualified for a technical document

When working in document-centric programs, designers are required to identify the notes that are applicable to various parts from among the qualified standard notes for their technical document.

To validate a note:

1. Open My Teamcenter.

2. Copy the standard note or the standard note revision.

3. Select the technical document revision and choose **Edit → Attach Requirements/Notes → Parametric Requirement**.

4. Expand the technical document revision to which the standard note is attached and expand the appropriate relations folder. For example, for an ADS part, expand the **ADS_Lists_PartRevisions** folder.

5. Right-click the corresponding item revision (ADS part revision, ADS drawing revision, or ADS design revision) and click **Properties on Relation**.

   > Note:
   >
   > If you chose the **ADS_Lists_PartRevisions** in step 4, right-click the ADS part revision.

   The **Properties on Relations** dialog box appears.

6. In the **Properties on Relations** dialog box, click the **Expand to Modify** button adjacent to the **Notes** box.

7. Click the **Add objects from Clipboard** button.

   The standard note is pasted in the **Notes** box.

8. Click **Apply**.

9. Click **OK**.

   The system verifies if the note that is pasted in the **Notes** box is associated with the technical document. If the note is not associated with the technical document revision, an error message is displayed.

Working with custom notes

Basic concepts about custom notes

Custom notes represent the unique notes for an individual parts list technical document or part. Unlike standard notes, which must be created and maintained by users with special privileges, custom notes are created by the designers who are responsible for the parts list technical document or part.

The following concepts apply to custom notes:

- Custom notes specify data that is unique to a part or document.

- Custom notes can be related to a single item or to multiple revisions of a single item.

  Example:

  Custom Note 0001 is attached to **Item 0001/A**.

  Or

  Custom Note 0001 is attached to **Item 0001/A**, **Item 0001/B**, and **Item 0001/C**.

- Custom notes cannot be attached to multiple items or to revisions of multiple items.

  Example:

  Custom Note 0001 is attached to **Item 0001/A**; therefore, **Custom Note 0001** cannot also be attached to **Item 0002/A**.

- Custom notes are associated with an item, such as a part or document, through the **Custom Requirements Lists** relation where the part revision or document revision is the primary object and the custom note or custom note revision is the secondary object.

- The text for custom note data is specified in the **Note Text** property of the note revision.

- Custom notes can be created in the context of another item, such as a parts list technical document or a part (assembly, component, standard part, and so on).

- When a custom note is created, a **Custom Note Text** dataset is created. This dataset contains the text and is attached to the custom note revision.

> Note:
>
> You cannot delete the dataset or detach the dataset from the custom note revision.

Create a custom note

1. Select the object to which the custom note applies.

2. Choose **File→New→Custom Note** or click the **Custom Note** button.

3. Select **Custom Note** as the type, and then click **Next**.

4. Type the revision ID, revision, and name for the custom note.

   > Note:
   >
   > You must type a name for the custom note in the **Name** box. If you leave the **Requirement ID** and **Revision** boxes blank, the values are filled in automatically when you click **Finish** to create the custom note.

5. (Optional) Type a description for the custom note.

6. Click **Finish**.

   The custom note symbol appears in the **Custom Notes** column for the selected item.

Add or edit custom note text

1. Open My Teamcenter.

2. Specify the custom note text in one of the following formats.

   Plain text format

   a. Right-click the custom note revision and choose **Edit Properties**.

      The **Check-Out** dialog box appears.

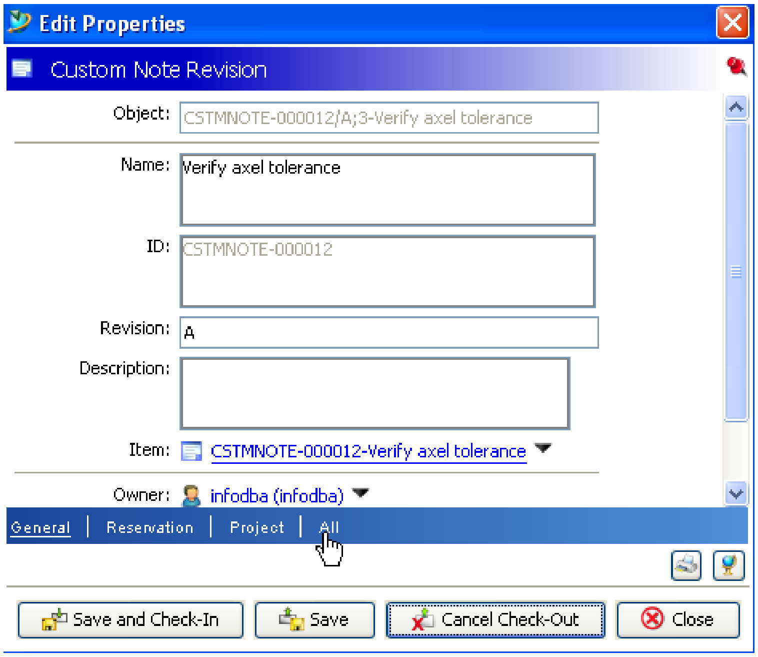
   b. In the **Change ID** box, type the change ID.

   c. In the **Comments** box, type notes for checkout.

d. Click **Yes**.

The **Edit Properties** dialog box appears.

e. Click the **All** link in the **Edit Properties** dialog box.



Object: CSTMNOTE-000012/A;3-Verify axel tolerance

Name: Verify axel tolerance

ID: CSTMNOTE-000012

Revision: A

Description:

Item: CSTMNOTE-000012-Verify axel tolerance

Owner: infodba (infodba)

f. Scroll to the **Text** box and type the custom note. For example:

```
Verify axle tolerance using company standard test
methods.
```

g. Click **Save and Check-In**.

The **Check-In** dialog box appears.

h. Click **Yes**.

The custom note is checked in and the new text appears in **Body Cleartext** property. The **Body Cleartext** property is displayed on the **Overview** tab in the **Summary** view.

Rich text format

Note:

To open the dataset in rich text format, you must have Microsoft Office installed on your machine.

a. Double-click the **Full Text** dataset associated with the note revision, or select the dataset and choose **File→Open**.

   The note opens in Microsoft Word for editing.

b. Type the note text and then click **Save**.

c. Close the Microsoft Word document.

> Note:
>
> After a note has been specified or updated in rich text format, it cannot be edited in plain text format.

Attach a custom note to an item or item revision

You can a attach a custom note revision to an object, such as a part, drawing, or design. Based on your requirement, you can also reuse or clone an existing custom note and attach it to another part. To clone a custom note, ensure that your administrator has set the **Fnd0AttachCustomNoteToMultiItems** global constant to *false* and the **Allow_Custom_Note_Cloning** preference to *true*.

1. Open My Teamcenter.

2. Right-click the custom note or the custom note revision that you want to attach and click **Copy**.

3. Select the part or part revision and choose **Edit→Attach Requirements/Notes→Custom Note**.

   Alternatively, in the **Summary** tab→**Overview** tab, in the **Custom Notes Lists** section, click **Attach**.

   The custom note is attached to the selected part or its revision and is listed in the **Summary** tab→**Overview** tab→**Custom Notes Lists** section.

> Note:
>
> In case you have attached a note that was previously attached to a part, the reused note is added as a new note with a different ID than the original one.

If the note is not visible under the item or item revision, you must specify **Custom Requirements Lists** as a shown relation for the item and item revision. To do this:

1. Choose **Edit→Options**.

2. Expand the **Options** tree and expand the **General** folder.

3. Click the **Item** or **Item Revision** node.

4. In the **General** options tab, choose **Custom Requirements Lists** from the **Available Relations** list.

5. Click + to move it to the **Shown Relations** list.

6. Click **OK**.

7. Choose **View→Refresh** or **View→Refresh Window** to refresh the display.

View and edit a custom note

1. Open My Teamcenter.

2. View or edit the custom note text in either of the following formats.

Plain text format

a. Right-click the custom note revision and click **Properties**. Alternatively, select the custom note revision and choose **View→Properties**.

   The **Properties** dialog box appears.

b. Click the **Check-Out and Edit** button.

   The **Check-Out** dialog box appears.

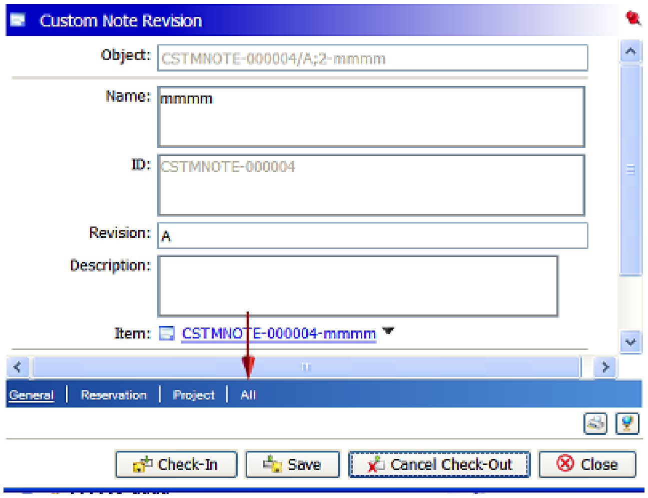
c. In the **Change ID** box, type the change ID.

d. In the **Comments** box, type notes for checkout.

e. Click **Yes**.

   The **Edit Properties** dialog box appears.

f. Click the **All** link in the **Edit Properties** dialog box and scroll to the **Text** box.



Custom Note Revision

Object: CSTMNOTE-000004/A;2-mmmm

Name: mmmm

ID: CSTMNOTE-000004

Revision: A

Description:

Item: CSTMNOTE-000004-mmmm

g. In the **Text** box, edit the custom note.

> Note:
>
> If the note text was defined in Microsoft Word, the **Text** box displays a message stating that you must edit the note using Microsoft Word.

h. Click **Save**.

i. Click the **Check-In** button.

The **Check-In** dialog box appears.

j. Click **Yes**.

Rich text format

> Note:
>
> To open the dataset in the rich text format, you must have Microsoft Office installed on your machine.

a. Double-click the **Full Text** dataset associated with the custom note revision, or select the dataset and choose **File→Open**.

The note opens in Microsoft Word for editing.

b. Update the note text and click **Save**.

c. Close the Microsoft Word document.

> Note:
>
> After a note is specified or updated in rich text format, it cannot be edited in plain text format.

Revise a custom note

1. Select the custom note revision that you want to revise and choose **File→Revise**.

   The **Revise** dialog box appears.

2. (Optional) In the **Define the basic information for the new item revision** pane, type the name of the new revision.

3. (Optional) Type a description of the revision and choose a unit of measure for the revision.

4. Click **Next**.

   The **Define additional item revision information** pane is displayed.

5. (Optional) Enter values for the note revision attributes.

6. Click **Next**.

   The **Define attached objects** pane displays the source revision and attachments and the destination revision and attachments. By default, all objects are copied forward to the new (destination) revision.

7. (Optional) Click the **Copy as object option set** button to the right of each object in the revision tree and select the desired copy option.

   ⊘ **Don't Copy**

   Allows you to selectively copy objects from one revision to another. Objects that are designated as **Don't Copy** appear with a line through them in the **Destination** tree.

   🗋 **Copy as Object**

   Creates a new object of the same type and relation to the parent object. Objects created by this method are totally independent of the parent object. Therefore, any changes to the copied object are not reflected in the original object.

Copied objects appear in bold in the **Destination** tree and can be renamed.

Copy as Reference

Copies the object as a reference to the original object. All changes to the reference copy affect the original object. The names of attachments that are copied as references cannot be modified.

8. Click **Next**.

   The **Select projects or programs that you want to assign new object to** pane is displayed.

9. (Optional) Assign the new revision to a different project or program.

10. Click **Next**.

    The **Select open option and alternate id display option** pane is displayed.

11. (Optional) Set the open, display, or checkout option for the new revision.

12. Click **Finish**.

    The new revision is created as a copy of the previous revision object. The new revision is independent of the source revision. Modifications to the new revision are not reflected in the source revision, which allows you to edit the note text for each revision.

7. Change management in Aerospace and Defense

Managing changes

The Aerospace and Defense solution extends the change management data model and adds attributes specific to the aerospace and defense industry to the following change objects:

- Change requests and change request revisions
- Change notices and change notice revisions
- Tasks and task revisions
- Deviation requests and deviation request revisions

Change Manager can be easily configured to use your change process.

- Use Change Manager to manage your change objects.
- Use Change Manager with Workflow Designer to track the evolution of changes through your organization according to a controlled, repeatable process.
- Use Change Manager with Schedule Manager to create work breakdown structures you can use to plan and schedule the changes you are making to your product.

Create a problem report

There are three different contexts in which to create a problem report.

1. (Optional) Select the item revision that you want to create a problem report for.
2. Choose **File→New→Change** or right-click the item revision and choose **New Change in context**.
3. In the **New Change** dialog box, select **Problem Report** and click **Next**.
4. Type a name in the **Synopsis** box and a description of the problem in the **Description** box. You can also type a problem report ID and revision in the **PR No.** and **Revision** boxes. If you do not provide an ID and revision number, Teamcenter provides them automatically.
5. Click **Finish**.

If you created the problem report by right-clicking the item revision and choosing **New Change in context**, the item revision is automatically copied into the problem report **Problem Items** folder.

Create a change request

There are three different contexts in which to create a problem report.

1. (Optional) Select the item revision or problem report revision that you want to create a change request for. You can select multiple problem report revisions.

2. Create the change request in one of the following ways:

   - Choose **File→New→Change**.

   - Right-click an item revision and choose **New Change in context**.

   - Select one or more problem report revisions, right-click, and choose **Derive Change**.

     The problem report revisions must have their properties set to:

Property	Must be
Maturity	**Reviewing**
Disposition	**Approved**
Closure	**Open**

     You can configure which change object you can derive from the problem report revision with the **CM_change_derivations** preference.

3. In the **New Change** dialog box, select **Change Request** and click **Next**.

4. If this is a new change, type a name in the **Synopsis** box and a description of the change request in the **Description** box. If this is a derived change, the boxes contain the name and description from the problem report revision. You can edit these boxes.

   You can also type a change request ID and revision in the **ECR No.** and **Revision** boxes. If you do not provide an ID and revision number, Teamcenter provides them automatically.

   If you derived this change request from a single problem report, the **Propagate relations** check box is available. If you want to copy the relationships (such as reference items and problem items) from the problem report to the change request, select the check box.

> Note:
>
> **Propagate relations** is available only when a change request is derived from a single problem report. When more than one problem report exists, this relationship must be created manually.

You can configure which relations to propagate when you derive a change object from a problem report with the **CM_ProblemReportRevision_Relations_To_Propagate** preference.

5. In the **Change Type** box, type the acronym that designates the type of change documentation used on a program.

6. In the **Change Class** box, type the government or company change classification code.

7. In the **Change Category** box, type the specific category of the change.

8. In the **Change Item Affected?** box, click either **True** or **False** to indicate if the change item affects other configuration items.

9. In the **Warranty Affected?** box, click either **True** or **False** to indicate if the change item affects the warranty.

10. In the **In Production?** box, click either **True** or **False** to indicate if the change item is in production.

11. In the **Is Primary Change?** box, click either **True** or **False** to indicate if the change item is a primary change.

12. In the **Retrofit Required?** box, click either **True** or **False** to indicate if the change item requires a retrofit.

13. Click **Finish**.

    If you created the change request by right-clicking the item revision and choosing **New Change in context**, the item revision is automatically copied into the change request **Problem Items** folder.

Create a deviation request

There are three different contexts in which to create a problem report.

1. (Optional) Select the item revision or problem report revision that you want to create a deviation request for. You can select multiple problem report revisions.

2. Create the deviation request in one of the following ways:

    - Choose **File→New→Change**.

- Right-click an item revision and choose **New Change in context**.
- Select one or more problem report revisions, right-click, and choose **Derive Change**.

  The problem report revisions must have their properties set to:

Property	Must be
Maturity	**Reviewing**
Disposition	**Approved**
Closure	**Open**

  You can configure which change object you can derive from the problem report revision with the **CM_change_derivations** preference.

3. In the **New Change** dialog box, select **Deviation Request** and click **Next**.

4. If this is a new change, type a name in the **Synopsis** box and a description of the deviation request in the **Description** box. If this is a derived change, the boxes contain the name and description from the problem report revision. You can edit these boxes.

   You can also type a deviation request ID and revision in the **ECR No.** and **Revision** boxes. If you do not provide an ID and revision number, Teamcenter provides them automatically.

   Enter the change in the **Change Type** box. You can enter **RFD** for a request for deviation or **RFW** for a request for waiver.

   If this deviation is recurring, select the **Is deviation recurring?** check box and type the reason in the **Rationale** box.

   If you derived this deviation request from a single problem report, the **Propagate relations** check box is available. If you want to copy the relationships (such as reference items and problem items) from the problem report to the deviation request, select the check box.

   You can configure which relations to propagate when you derive a change object from a problem report with the **CM_ProblemReportRevision_Relations_To_Propagate** preference.

5. In the **Change Item Affected?** box, click either **True** or **False** to indicate if the change item affects other configuration items.

6. In the **Warranty Affected?** box, click either **True** or **False** to indicate if the change item affects the warranty.

7. Click **Finish**.

If you created the deviation request by right-clicking the item revision and choosing **New Change in context**, the item revision is automatically copied into the deviation request **Problem Items** folder.

Create a change notice

There are three different contexts in which to create a problem report.

1. (Optional) Select the problem item revision or change request revision that you want to create a change notice for.

2. Create the change notice in one of the following ways:

   - Choose **File→New→Change**.

   - Select the item revision, right-click, and choose **New Change in context**.

   - Select one or more change request revisions, right-click, and choose **Derive Change**.

     The change request revisions must have their properties set to:

Property	Must be
Maturity	**Reviewing**
Disposition	**Approved**
Closure	**Open**

     You can configure which change object you can derive from the change request revision with the **CM_change_derivations** preference.

3. In the **New Change** dialog box, select **Change Notice** and click **Next**.

4. Type a name in the **Synopsis** box and a description of the change notice in the **Description** box. You can also type a change notice ID and revision in the **ECN No.** and **Revision** boxes. If you do not provide an ID and revision number, Teamcenter provides them automatically.

   If you derived this change notice from a single change request, the **Propagate relations** check box is available. If you want to copy the relationships (such as reference items and problem items) from the change request to the change notice, select the **Propagate relations** check box.

   You can configure which relations to propagate when you derive a change object from a change request with the **CM_ChangeRequestRevision_Relations_To_Propagate** preference.

5. In the **Change Type** box, type the acronym that designates the type of change documentation used on a program.

6. In the **Paper Change?** box, select either **True** or **False** to indicate if the change notice is an unincorporated change.

7. In the **Change Class** box, type the government or company change classification code.

8. In the **Category** box, type the specific category of the change.

9. Click **Finish**.

   If you created the change notice by right-clicking the item revision and choosing **New Change in context**, the item revision is automatically copied into the change notice **Problem Items** folder.

Create a work breakdown using Schedule Manager

Note:

You can copy an existing schedule but it can only be associated with one change object.

1. Select the **Plan Items** folder of the change request or change notice that you want to create a work breakdown for.

   For a ECR or ECN, you can add objects to the **Plan Items** folder if you are an assigned participant. The change object property settings are as follows.

Assigned participant	Closure	Disposition	Maturity
Analyst	**Open**	**None**	**Elaborating**
	or		
	Open	**Investigate**	**Reviewing**

2. Choose **File→New→Schedule**.

   This starts the New Schedule wizard in Schedule Manager where you create your work breakdowns. You can create several schedules to help you analyze the change.

   Note:

   You must be an **Author** user to create schedules.

3. In the **Open Change** view, select the schedule in the **Plan Items** folder and click the **Open Schedule** button .

   The **Schedule Manager** view is displayed.

4. Add tasks to your schedules.

   When you create a task, add the following Aerospace and Defense specific attributes in the **Select task type, start, and finish dates** dialog box.

   a. In the **Administrative Task?** box, click **True** or **False** to indicate if the task is an administrative task.

   b. In the **Category** box, type the category of the task.

   c. In the **Complexity** box, type the category of the change, document, or part.

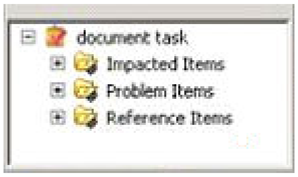
   d. In the **Impact Assessment Required** box, click **True** or **False** to indicate if the task requires impact assessment.

   e. In the **Proposed Task?** box, click **True** or **False** to indicate if the task is a proposed task versus an implemented task.

5. Assign resources to tasks, including selecting responsible people using the **Membership** button .

6. In the **Schedule Manager** view, click the **View Task folders** button .

   The system displays the **Open Task** view.



⊟ document task
⊞ Impacted Items
⊞ Problem Items
⊞ Reference Items

7. Add item revisions to the task folders as necessary.

8. (Optional) Roll up the item revisions in your task folders to the change object folders.

Managing unincorporated changes

About incorporating changes

An unincorporated change is a change that is approved but not yet incorporated in the design of a product. Usually, changes are not incorporated fully because there is not enough time, importance, or budget to incorporate the changes completely through the development process.

The unincorporated change functionality is intended to track unincorporated changes to existing designs. It does not support tracking unfinished work of new, unreleased designs.

Unincorporated changes can only be applied to ECNs.

> Example:
>
> A designer releases Revision A of **Item1** for production, and begins working on a new revision (Revision B). On the shop floor, a minor modification is performed over Revision A and production starts. This changed version needs to be recorded in Teamcenter. Therefore, a new supplemental revision (A01) is created and released. Because the change still needs to be incorporated into the main release, (revision B, in this case), the change is considered partially incorporated.

An ECN can also either partially or fully incorporate markups. However, a markup is considered fully incorporated into all the solution items of the incorporating ECN, which has the change partially incorporated into the same item.

> Example:
>
> A designer creates a ECN (**CN1**), with a markup (**MU1**) as a solution item. The designer sets the Incorporation status of several of the impacted items to **Partially Incorporated**.
>
> The designer then creates a second ECN (**CN2**) to incorporate markup **MUI**. The designer creates an incorporates relation between **CN2** and **MU1**.
>
> **CN1/MU1** are now considered fully incorporated into the solution items of **CN2** that belong to the impacted items of **CN1** and have an Incorporation status of **Partially Incorporated**.

For a full example of incorporating changes, see Example of incorporating changes.

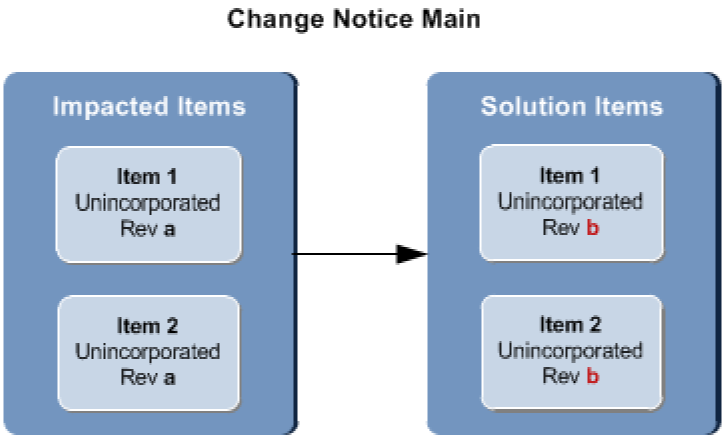
Example of incorporating changes

The following is an example showing the steps to incorporate changes partially and then fully when an engineering change notice (ECN) impacts multiple change items.

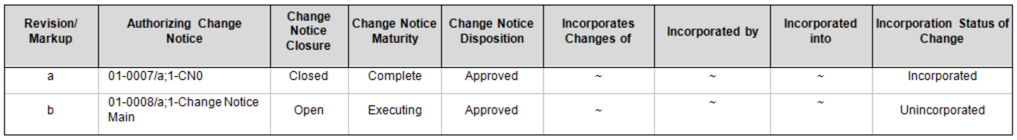
Step 1 – Create engineering change notice implementing changes

Company ABC creates an ECN (**Change Notice Main**) to implement a change needed on two items (**Item 1** and **Item 2**). The item revision of the items in the **Solutions** folder is Revision b.

The ECN change state is **Open**, **Executing**, and **Approved**.



The following shows the status for the items as it would appear in the **Change History** dashboard.



Revision/ Markup	Authorizing Change Notice	Change Notice Closure	Change Notice Maturity	Change Notice Disposition	Incorporates Changes of	Incorporated by	Incorporated into	Incorporation Status of Change
a	01-0007/a;1-CN0	Closed	Complete	Approved	~	~	~	Incorporated
b	01-0008/a;1-Change Notice Main	Open	Executing	Approved	~	~	~	Unincorporated

Note:

ECN **CN0** in the **Change History** dashboard is not shown in the figures to avoid complexity. **CN0** is the ECN that initially created **Item 1**, Revision a, and **Item 2**, Revision a as solution items.

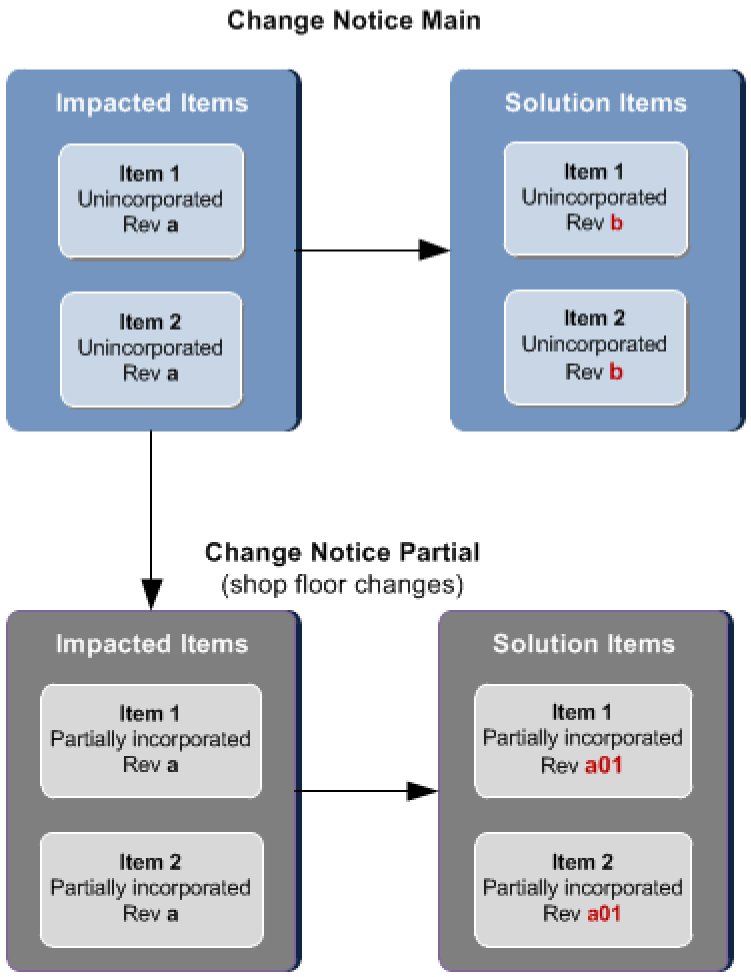
Step 2 – Make the partially incorporated changes

On the shop floor, a minor modification is done over Revision a of the two items (**Item 1** and **Item 2**) and production is started. Company ABC needs to immediately incorporate the change into Teamcenter. The items have work-in-process revisions so the change cannot be incorporated into these.

Therefore, a designer creates new revisions to incorporate the shop-floor changes and starts a new change notice (**Change Notice Partial**) to manage the implementation. The designer sets the incorporation status on the items in the **Impacted Items** folder of **Change Notice Partial** to **Partially Incorporated**.

The designer closes **Change Notice Partial** after creating Revision a01 of **Item 1** and **Item 2**.

Work still continues on Revision b in **Change Notice Main**, and it remains open.

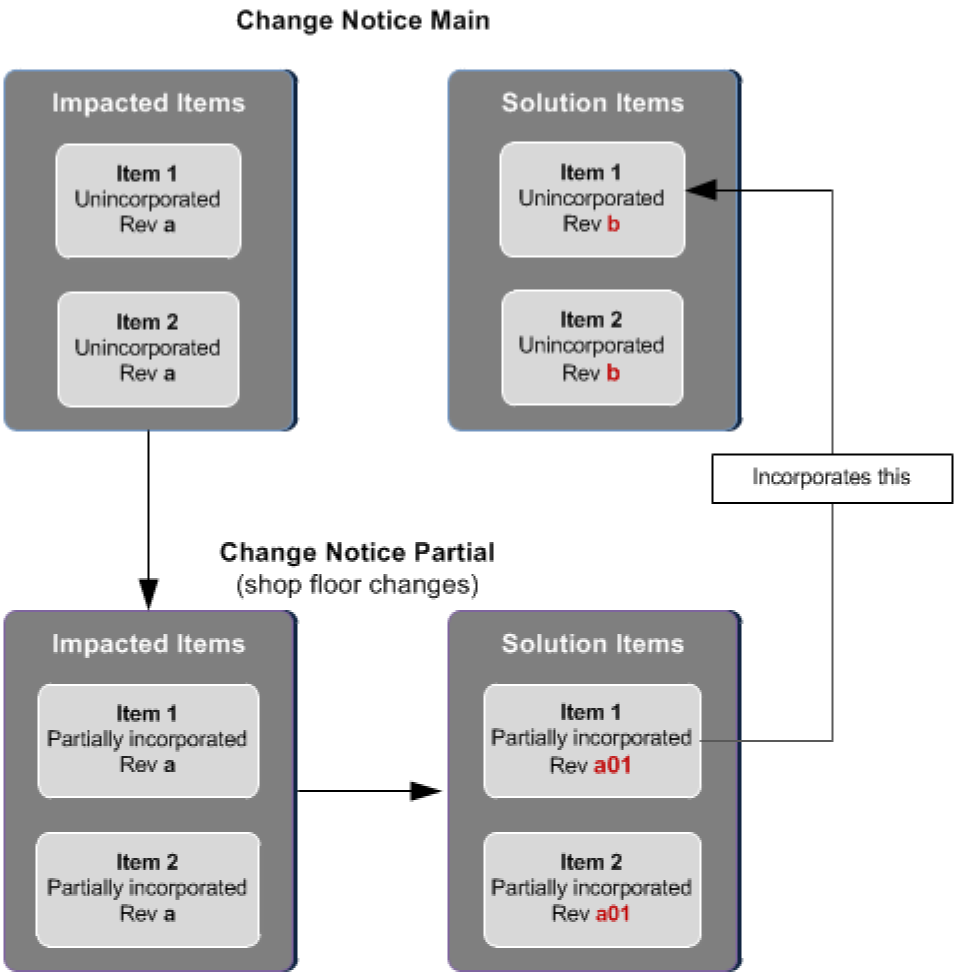


The change states of the ECNs are:

- **Change Notice Main** – **Open**, **Executing**, and **Approved**
- **Change Notice Partial** – **Closed**, **Complete**, and **Approved**

Step 3 – Incorporate changes from Item 1, Revision a01, into Revision b in Change Notice Main

The designer finishes the work in **Change Notice Main** on Revision b of the items. The designer wants to incorporate the change to **Item 1** in **Change Notice Partial** into Revision b of **Change Notice Main**, but not the change that was done to **Item2**. To indicate this, the designer creates an **Incorporates this** relation between **Change Notice Main** and Revision a01.

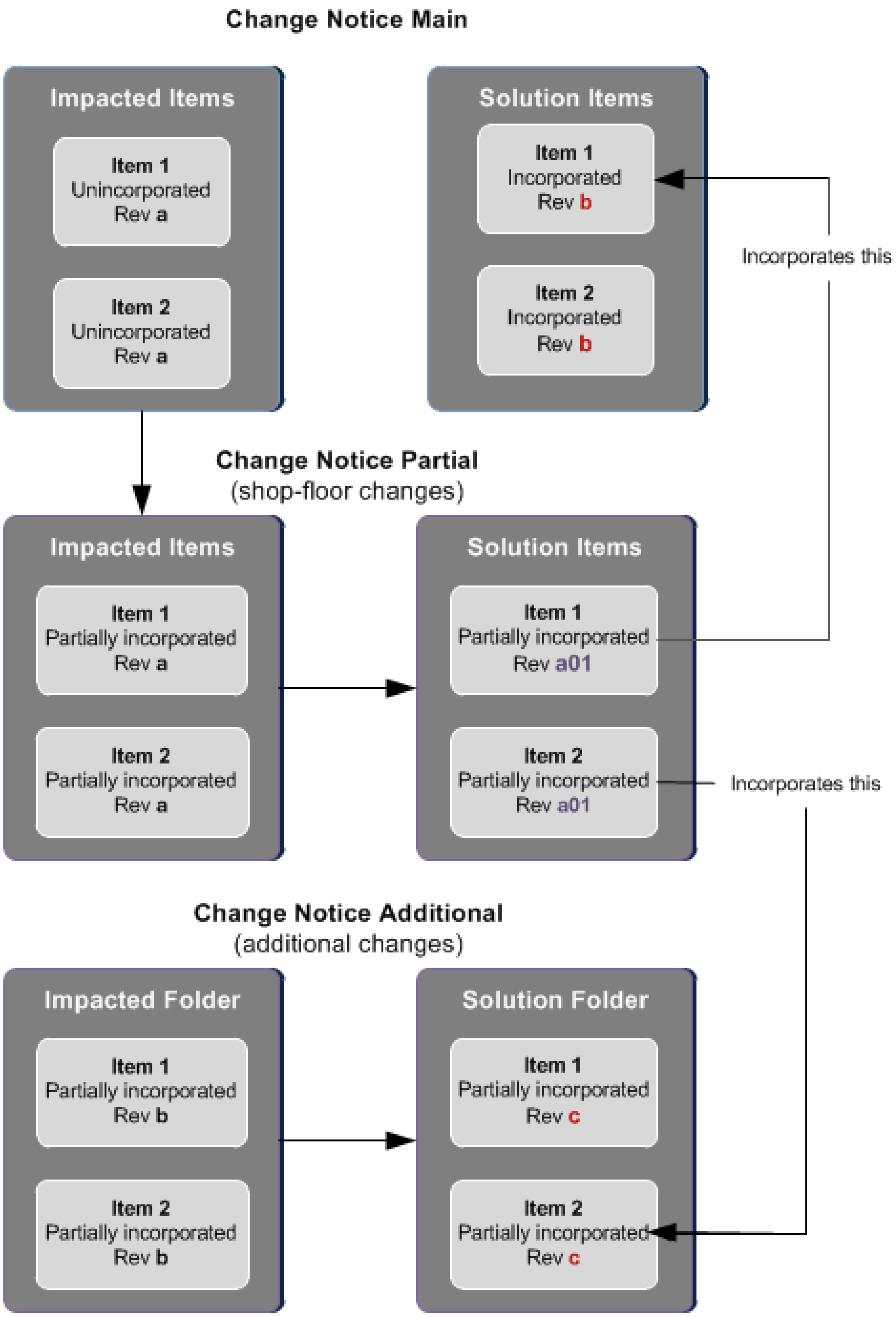

Change Notice Main
Impacted Items
Item 1 Unincorporated Rev a
Item 2 Unincorporated Rev a
Solution Items
Item 1 Unincorporated Rev b
Item 2 Unincorporated Rev b
Incorporates this
Change Notice Partial (shop floor changes)
Impacted Items
Item 1 Partially incorporated Rev a
Item 2 Partially incorporated Rev a
Solution Items
Item 1 Partially incorporated Rev a01
Item 2 Partially incorporated Rev a01

The change states of the ECNs are both Closed, Complete, and Approved.

Step 4 – Incorporate changes from Item 2, Revision a01, into Revision b in Change Notice Additional

Much later, more changes are recommended for **Item 1** and **Item 2**. Therefore, Company ABC creates a third change notice (**Change Notice Additional**). During that change, the designer implements the changes that were in **Change Notice Partial** for **Item 2** into Revision c.

At this point, all change notices are closed.


Change Notice Main
Impacted Items
Item 1 Unincorporated Rev a
Item 2 Unincorporated Rev a
Solution Items
Item 1 Incorporated Rev b
Item 2 Incorporated Rev b
Incorporates this
Change Notice Partial
(shop-floor changes)
Impacted Items
Item 1 Partially incorporated Rev a
Item 2 Partially incorporated Rev a
Solution Items
Item 1 Partially incorporated Rev a01
Item 2 Partially incorporated Rev a01
Incorporates this
Change Notice Additional
(additional changes)
Impacted Folder
Item 1 Partially incorporated Rev b
Item 2 Partially incorporated Rev b
Solution Folder
Item 1 Partially incorporated Rev c
Item 2 Partially incorporated Rev c

The change states of the ECNs are all Closed, Complete, and Approved.

Conditions that must be met to set Incorporation status

The following conditions must be met before you can set the Incorporation status of an item revision. By default, all impacted items in the **Impacted Items** folder of an ECN are set to **Unincorporated**.

In addition, when an ECN is closed through a workflow, all impacted items with a status of **Unincorporated** are automatically set to **Incorporated**.

Conditions required for a change to be partially incorporated

A change authorized by an ECN revision is considered partially incorporated in an item revision if the following are met:

- The item revision is a solution item of the ECN (in its **Solution Items** folder). If it is, then it meets the Business Modeler IDE condition **CMHasSolutionItem**.

  > Note:
  >
  > The **CMHasSolutionItem** condition is not supported by change revisions (ECR).

- The ECN has an impacted item in its **Impacted Items** folder and its Incorporation status is set to **Partially Incorporated**.

  The **CMHasImpactedItem** relation object has the ECN revision as its primary object and an item revision of the same item as its secondary object In addition, the value of the property **Cm0IncorporationStatus** on the **CMHasImpactedItem** relation object is set to **Partially Incorporated**.

Conditions required for fully incorporating a change

A change previously partially incorporated into an item revision, (for example, Revision a01) is considered fully incorporated into another item revision (for example, Revision b) of the same item if all the following are met:

- The ECN authorizing the item revision (in this example, Revision b) is a primary object of a **Cm0Incorporates** relation object and the previous revision (for example, Revision a01) is the secondary object.

- The ECN has an impacted item in its **Impacted Items** folder and its incorporation status is set to **Incorporated**.

  The **CMHasImpactedItem** relation object has the ECN as the primary object and an item revision of the same item as its secondary object. In addition, the value of the property **Cm0IncorporationStatus** on the **CMHasImpactedItem** relation object is set to **Incorporated**.

- The ECN revision authorizing the item revision (in this case, Revision b) is closed.

Conditions required for incorporating markups

An ECN incorporates a markup partially or fully in the same way it does item revisions, as explained in the two earlier sections. However, a markup is considered fully incorporated into all the solution items of the incorporating ECN that has the change partially incorporated into the same item.

> Example:
>
> A markup (**MU1**) is a solution item of an ECN (**CN1**) and the incorporation status of a few of the impacted items of **CN1** is set to **Partially Incorporated**.
>
> A designer later creates an **Incorporates** relation between a new ECN (**CN2**) and the markup **MU1**.
>
> Now CN1 and MU1 are considered fully incorporated into the solution items of **CN2**, which belong to the impacted items of **CN1** with the Incorporation status set to **Partially Incorporated**.

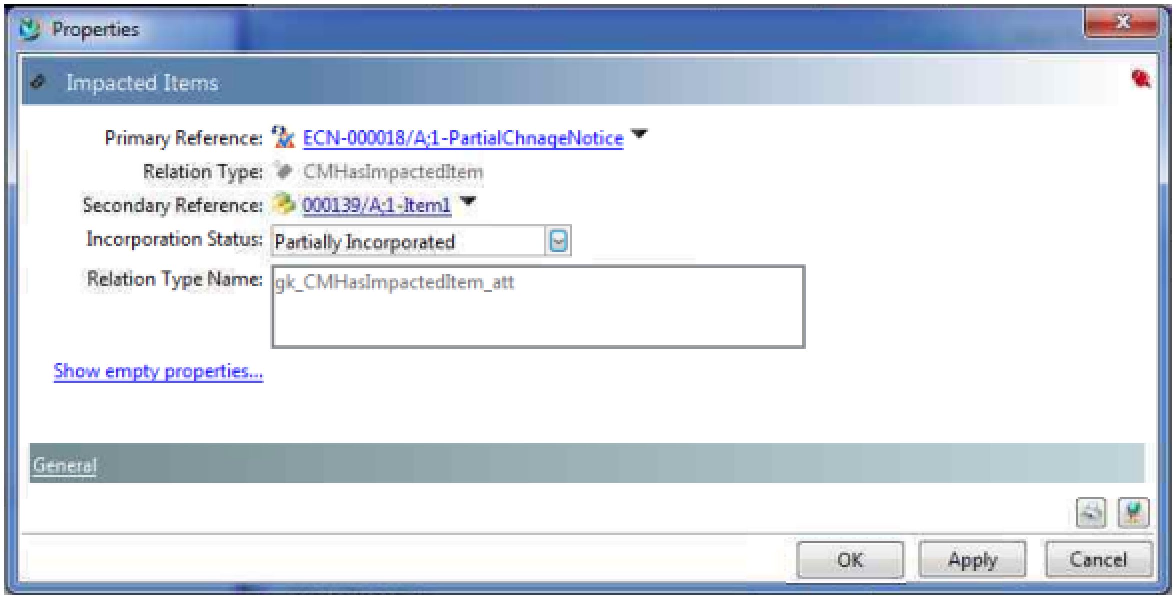
Set the Incorporation status of an object

Before you can set the Incorporation status of an item revision:

- The item revision must be related to an engineering change notice (ECN) as an item that it impacts.
- **Conditions must be met to set the Incorporation status of item revision**.

> Note:
>
> By default, the **Incorporates** relation is not displayed as a folder under the ECN revision.
>
> For information about displaying a folder with this relation under the ECN, see **Configure a folder to display Incorporation status**.

1. In My Teamcenter or Change Manager, right-click the item revision and choose **Properties on Relation**.
2. In the **Properties** dialog box, scroll to **Incorporation Status** and set it to one of the following:



Primary Reference: ECN-000018/A;1-PartialChnageNotice

Relation Type: CMHasImpactedItem

Secondary Reference: 000139/A;1-Item1

Incorporation Status: Partially Incorporated

Relation Type Name: gk_CMHasImpactedItem_att

- **Unincorporated**

  The change has not been incorporated.

- **Partially Incorporated**

  The change as defined has been partially incorporated. It may or may not already be fully incorporated by a subsequent change.

- **Incorporated**

  The change as defined has been fully incorporated. No subsequent changes are required to fully incorporate the defined change.

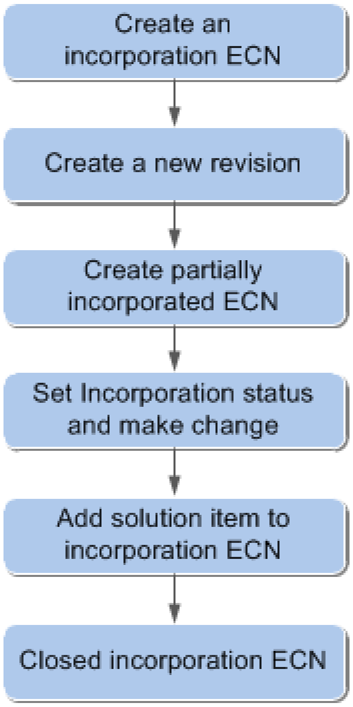
- **Cancelled**

  The change as defined is not incorporated in any way. This value indicates that an item has been identified as an impacted item, but the change as defined is not applied to that item. This status can be applied before a change solution definition or change execution has begun, or it can be updated after the change execution has begun.

3. Click **OK**.

General process for incorporating change items in an ECN

The following shows the general process for incorporating change items in an engineering change notice (ECN). It is followed by a summary of the steps.


Create an incorporation ECN
Create a new revision
Create partially incorporated ECN
Set Incorporation status and make change
Add solution item to incorporation ECN
Closed incorporation ECN

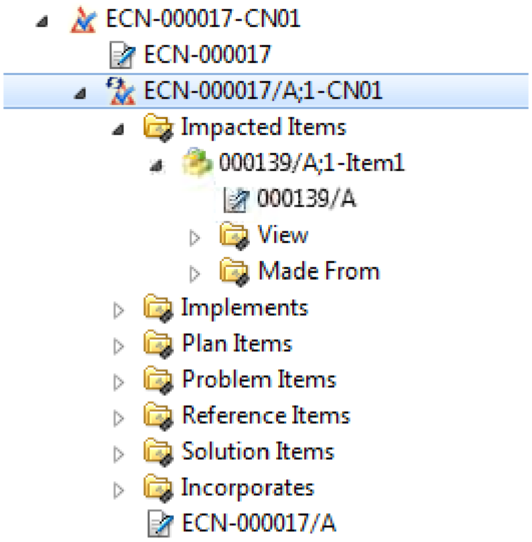
Note:

The examples use primary revisions. We recommend that you use secondary revisions when partially incorporating change items. For example, use A01 or A02 and not A, B, and C.

1. Create the ECN to completely incorporate the item revision.

   a. Create the ECN.

   b. Add the item revision to be incorporated to its **Impacted Items** folder.



```
◢ ECN-000017-CN01
    ECN-000017
  ◢ ECN-000017/A;1-CN01
    ◢ Impacted Items
      ◢ 000139/A;1-Item1
          000139/A
        ▷ View
        ▷ Made From
    ▷ Implements
    ▷ Plan Items
    ▷ Problem Items
    ▷ Reference Items
    ▷ Solution Items
    ▷ Incorporates
      ECN-000017/A
```

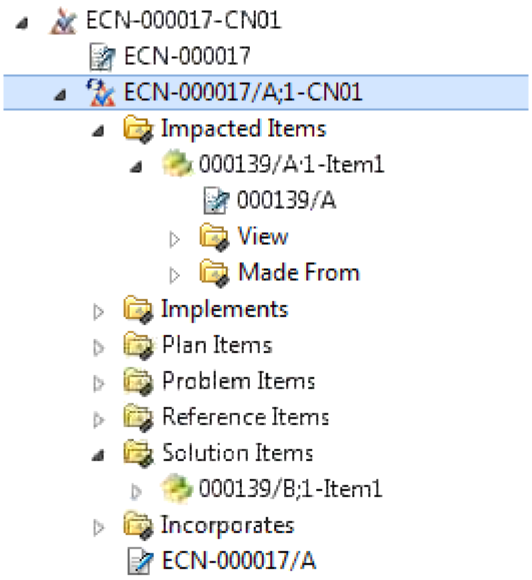
c. Assign participants and start a workflow.

2. Create the new item revision.

a. Use the **Revise Impacted Items(s)** command to create a new revision of the item revision and add it to the **Solutions Items** folder.

The following properties must be met:

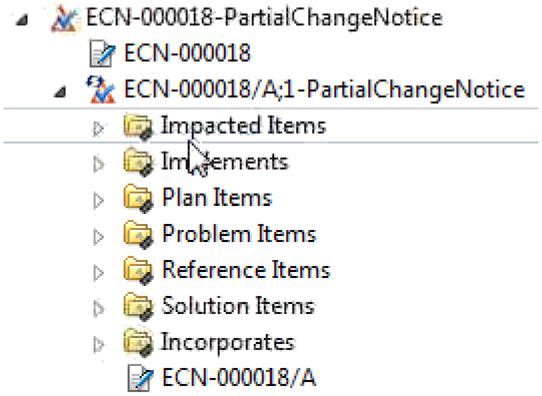
Assigned participant	**Closure/Disposition/Maturity property settings**
Analyst	**Open/Approved/Executing**



```
ECN-000017-CN01
    ECN-000017
    ECN-000017/A;1-CN01
        Impacted Items
            000139/A;1-Item1
                000139/A
                View
                Made From
        Implements
        Plan Items
        Problem Items
        Reference Items
        Solution Items
            000139/B;1-Item1
        Incorporates
        ECN-000017/A
```

3. Create the second ECN that is to incorporate the item revision partially (the partially incorporated ECN).

a. Create the partially incorporated ECN.

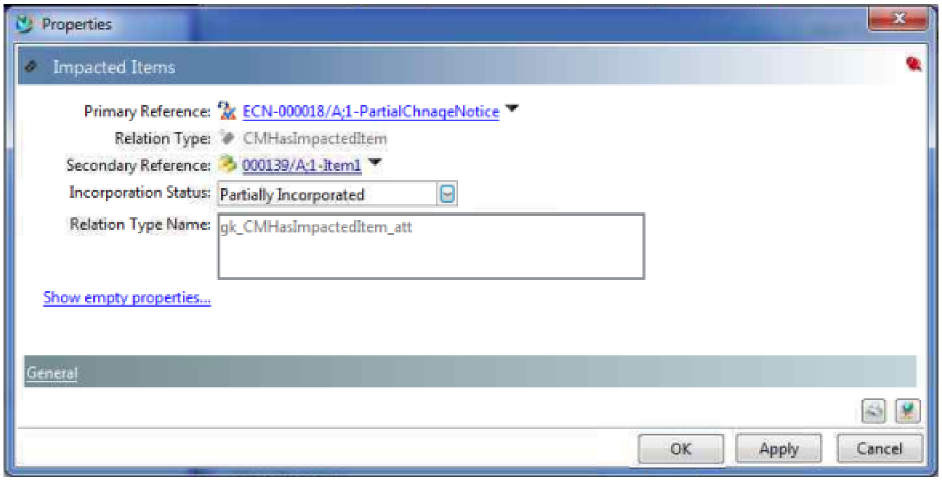


- ECN-000018-PartialChangeNotice
  - ECN-000018
  - ECN-000018/A;1-PartialChangeNotice
    - Impacted Items
    - Implements
    - Plan Items
    - Problem Items
    - Reference Items
    - Solution Items
    - Incorporates
    - ECN-000018/A

b. Assign participants and start a workflow.

4. Set the Incorporation status and make the change.

a. Use the **Properties on Relation** command to set the Incorporation status of the item revision to **Partially incorporated**



Properties

Impacted Items

Primary Reference: ECN-000018/A;1-PartialChnageNotice

Relation Type: CMHasImpactedItem

Secondary Reference: 000139/A;1-Item1

Incorporation Status: Partially Incorporated

Relation Type Name: gk_CMHasImpactedItem_att

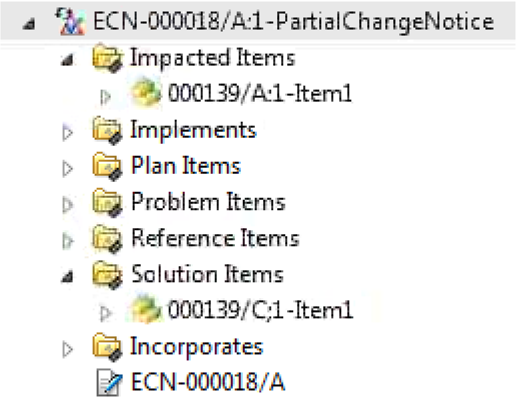
Show empty properties...

General

b. Use the **Revise Impacted Items(s)** command to create a new revision of the item revision and add it to the **Solutions Items** folder.

The following properties must be met:

Assigned participant	Closure/Disposition/Maturity property settings
Analyst	**Open/Approved/Executing**



- ECN-000018/A:1-PartialChangeNotice
  - Impacted Items
    - 000139/A:1-Item1
  - Implements
  - Plan Items
  - Problem Items
  - Reference Items
  - Solution Items
    - 000139/C;1-Item1
  - Incorporates
  - ECN-000018/A

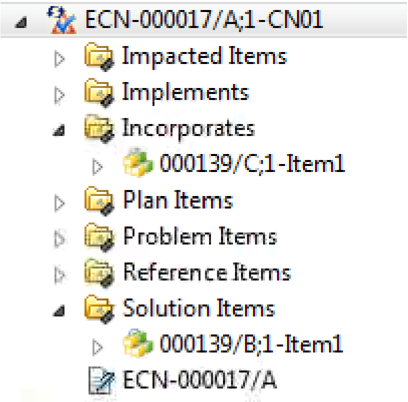
c. Make the change.

5. Add solution item to incorporation ECN.

   a. Add the item revision from the **Solutions Items** folder of the second ECN to the **Incorporates** folder of the first ECN that will incorporate all changes.

      The following properties must be set to add the item revision to the **Incorporates** folder:

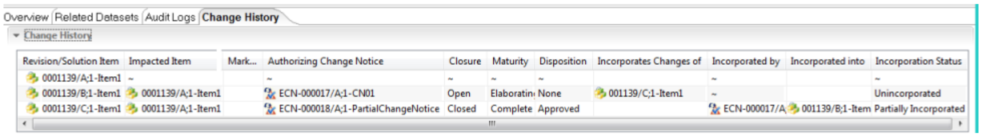
Assigned participant	Closure/Disposition/Maturity property settings
Analyst	**Open/Approved/Elaborating**



- ECN-000017/A;1-CN01
  - Impacted Items
  - Implements
  - Incorporates
    - 000139/C;1-Item1
  - Plan Items
  - Problem Items
  - Reference Items
  - Solution Items
    - 000139/B;1-Item1
  - ECN-000017/A

6. Close the incorporation ECN.

   a. Close the first ECN that incorporates the solution item. Closing the ECN automatically sets the Incorporated status of the solution item to **Incorporated** as shown in the **Change History** dashboard.



Revision/Solution Item	Impacted Item	Mark...	Authorizing Change Notice	Closure	Maturity	Disposition	Incorporates Changes of	Incorporated by	Incorporated into	Incorporation Status
0001139/A;1-Item1	~		~	~	~	~		~		~
0001139/B;1-Item1	0001139/A;1-Item1		ECN-000017/A;1-CN01	Open	Elaboratin	None	001139/C;1-Item1	~		Unincorporated
0001139/C;1-Item1	0001139/A;1-Item1		ECN-000018/A;1-PartialChangeNotice	Closed	Complete	Approved		ECN-000017/A	001139/B;1-Item	Partially Incorporated

The following preferences control defaults in change item revisions:

ChangeItemRevision_Cpd0SubsetDefaults_default_relation	
Description	When creating a change object in the context of a subset, specifies the relation the subset has to the created change item revision. The default is that the subset is a reference item.
Default	**CMReferences**
Default Protection Scope	User

ChangeItemRevision_Cpd0DesignElement_default_relation	
Description	When creating a change object in the context of a design element, specifies the relation the design element has to the created change item revision. The default is that the design element is an impacted item.
Default	**CMHasImpactedItem**
Default Protection Scope	User

ChangeItemRevision_Cpd0WorksetRevision_default_relation	
Description	When creating a change object in the context of a workset, specifies the relation the workset has to the created change item revision. The default is that the workset is a reference item.
Default	**CMReferences**
Default Protection Scope	User

ChangeItemRevision_DefaultChildProperties	
Description	Specifies the list of properties that can be displayed as children of a **ChangeItemRevision** node.
Default	**IMAN_master_form_rev** **IMAN_specification** **IMAN_requirement** **IMAN_manifestation** **IMAN_reference** **release_status_list** **TC_Attaches**

	CMImplements **CMImplementedBy** **CMHasImpactedItem** **CMHasProblemItem** **CMReferences** **CMHasSolutionItem** **CMHasWorkBreakdown**
Default Protection Scope	Site

ChangeItemRevision_ItemRevision_default_relation	
Description	When creating a change object in the context of an item revision, specifies the relation the item revision has to the created change item revision. The default is that the item revision is a problem item, as shown in the figure for a change request.
Default	**CMHasProblemItem**
Default Protection Scope	User

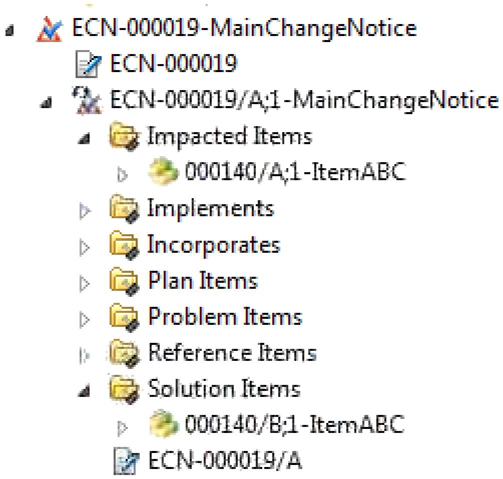
ChangeItemRevision_Ptn0Partition_default_relatio	
Description	When creating a change object in the context of a partition, specifies the relation the partition has to the created change item revision. The default is that the partition is an impacted item, as shown in the figure for a change request.
Default	**CMHasImpactedItem**
Default Protection Scope	User

General process for incorporating markups in an ECN

The following provides the general process for incorporating change items in an engineering change notice (ECN) when the solution item is a markup of the impacted item (they are different items; not revisions of each other).

> Note:
>
> The examples use primary revisions. We recommend that you use secondary revisions when partially incorporating change items. For example, use A01 or A02 and not A, B, and C.

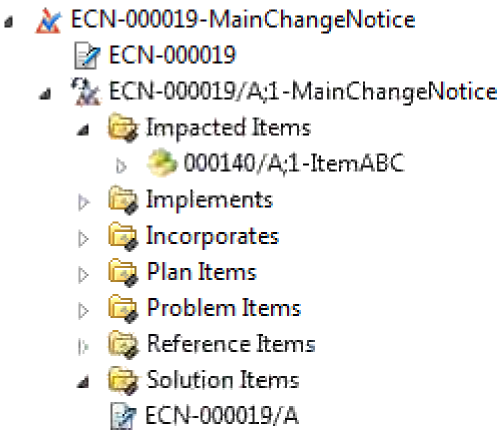
1. Create the ECN to completely incorporate the change.



```
ECN-000019-MainChangeNotice
    ECN-000019
    ECN-000019/A;1-MainChangeNotice
        Impacted Items
            000140/A;1-ItemABC
        Implements
        Incorporates
        Plan Items
        Problem Items
        Reference Items
        Solution Items
            000140/B;1-ItemABC
        ECN-000019/A
```

a. Create the ECN.

b. Add the item revision to be incorporated to its **Impacted Items** folder.



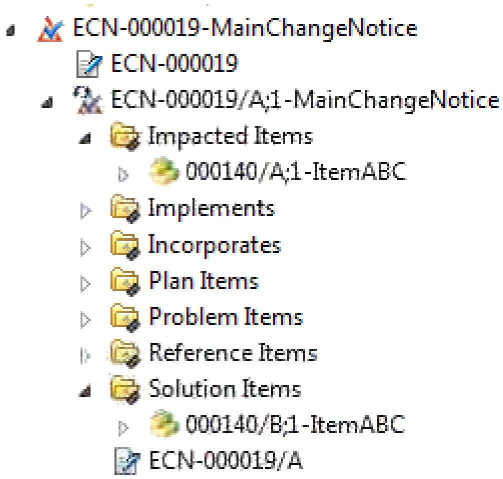
```
ECN-000019-MainChangeNotice
    ECN-000019
    ECN-000019/A;1-MainChangeNotice
        Impacted Items
            000140/A;1-ItemABC
        Implements
        Incorporates
        Plan Items
        Problem Items
        Reference Items
        Solution Items
        ECN-000019/A
```

c. Assign participants and start a workflow.

d. Use the **Revise Impacted Items(s)** command to create a new revision of the item revision and add it to the **Solutions Items** folder.

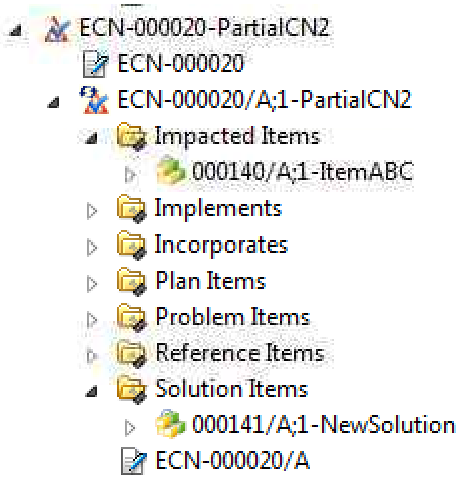
The following properties must be met:

Assigned participant	Closure/Disposition/Maturity property settings
Analyst	**Open/Approved/Executing**



- ECN-000019-MainChangeNotice
  - ECN-000019
  - ECN-000019/A;1-MainChangeNotice
    - Impacted Items
      - 000140/A;1-ItemABC
    - Implements
    - Incorporates
    - Plan Items
    - Problem Items
    - Reference Items
    - Solution Items
      - 000140/B;1-ItemABC
    - ECN-000019/A

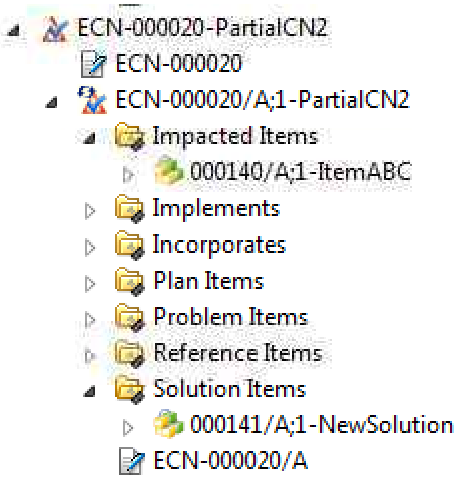
2. Create a second ECN that is to incorporate the item revision partially and make the partial change. The following shows the completed ECN with different impacted and solution items.



- ECN-000020-PartialCN2
  - ECN-000020
  - ECN-000020/A;1-PartialCN2
    - Impacted Items
      - 000140/A;1-ItemABC
    - Implements
    - Incorporates
    - Plan Items
    - Problem Items
    - Reference Items
    - Solution Items
      - 000141/A;1-NewSolution
    - ECN-000020/A

   a. Create the second ECN.

   b. Add the item revision to be partially incorporated to its **Impacted Items** and **Solutions Items** folders. In this case, they are different items that are unrelated.



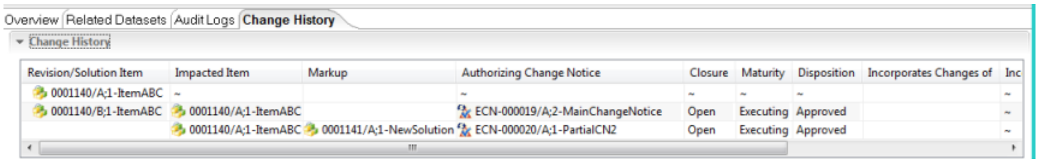
- ECN-000020-PartialCN2
  - ECN-000020
  - ECN-000020/A;1-PartialCN2
    - Impacted Items
      - 000140/A;1-ItemABC
    - Implements
    - Incorporates
    - Plan Items
    - Problem Items
    - Reference Items
    - Solution Items
      - 000141/A;1-NewSolution
    - ECN-000020/A

c. Assign participants and start a workflow.

d. Use the **Properties on Relation** command to set the Incorporation status of the item revision in the **Impacted Items** folder to **Partially incorporated**

For more information about setting the Incorporation status of a change item, see Set the Incorporation status of an object.

e. Use the **Relate Solution Item to Impacted Item** command to relate the solution item to the impacted item, which creates a link between them, as shown in the **Change History** dashboard.



Revision/Solution Item	Impacted Item	Markup	Authorizing Change Notice	Closure	Maturity	Disposition	Incorporates Changes of	Inc
0001140/A;1-ItemABC	~		~	~	~	~		~
0001140/B;1-ItemABC	0001140/A;1-ItemABC		ECN-000019/A;2-MainChangeNotice	Open	Executing	Approved		~
	0001140/A;1-ItemABC	0001141/A;1-NewSolution	ECN-000020/A;1-PartialCN2	Open	Executing	Approved		~

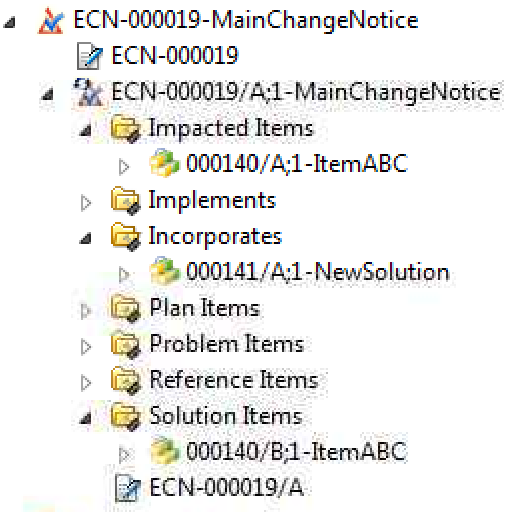
For more information about the **Change History** dashboard, see View Incorporation status and history.

f. Make the change and close the ECN.

3. Add the item revision from the **Solutions Items** folder of the second ECN to the **Incorporates** folder of the first ECN that will incorporate all changes.

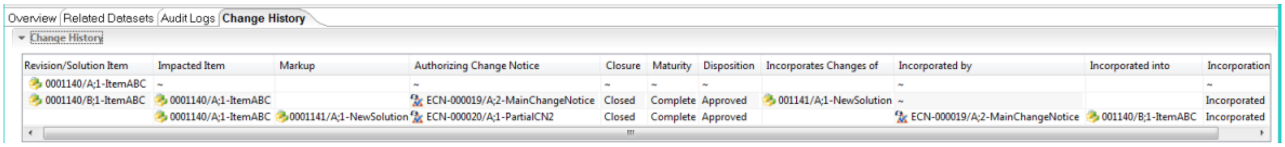
The following properties must be set to add the item revision to the **Incorporates** folder:

Assigned participant	Closure/Disposition/Maturity property settings
Analyst	**Open/Approved/Elaborating**



- ECN-000019-MainChangeNotice
  - ECN-000019
  - ECN-000019/A;1-MainChangeNotice
    - Impacted Items
      - 000140/A;1-ItemABC
    - Implements
    - Incorporates
      - 000141/A;1-NewSolution
    - Plan Items
    - Problem Items
    - Reference Items
    - Solution Items
      - 000140/B;1-ItemABC
    - ECN-000019/A

4. Close the first ECN that incorporates the solution item. Closing the ECN automatically sets the Incorporated status of the solution item to **Incorporated** as shown in the **Change History** dashboard.

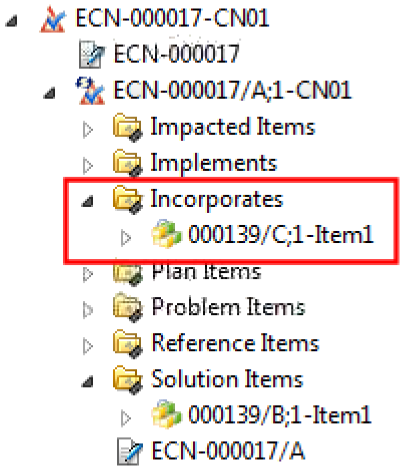


Revision/Solution Item	Impacted Item	Markup	Authorizing Change Notice	Closure	Maturity	Disposition	Incorporates Changes of	Incorporated by	Incorporated into	Incorporation
0001140/A;1-ItemABC	~		~	~	~	~		~		~
0001140/B;1-ItemABC	0001140/A;1-ItemABC		ECN-000019/A;2-MainChangeNotice	Closed	Complete	Approved	001141/A;1-NewSolution	~		Incorporated
	0001140/A;1-ItemABC	0001141/A;1-NewSolution	ECN-000020/A;1-PartialCN2	Closed	Complete	Approved		ECN-000019/A;2-MainChangeNotice	001140/B;1-ItemABC	Incorporated

For more information about the **Change History** dashboard, see **View Incorporation status and history**.

Configure a folder to display Incorporation status

By default, the Incorporation status **Incorporates** is not displayed as a folder under an engineering change notice (ECN) revision.



- ECN-000017-CN01
  - ECN-000017
  - ECN-000017/A;1-CN01
    - Impacted Items
    - Implements
    - Incorporates
      - 000139/C;1-Item1
    - Plan Items
    - Problem Items
    - Reference Items
    - Solution Items
      - 000139/B;1-Item1
    - ECN-000017/A

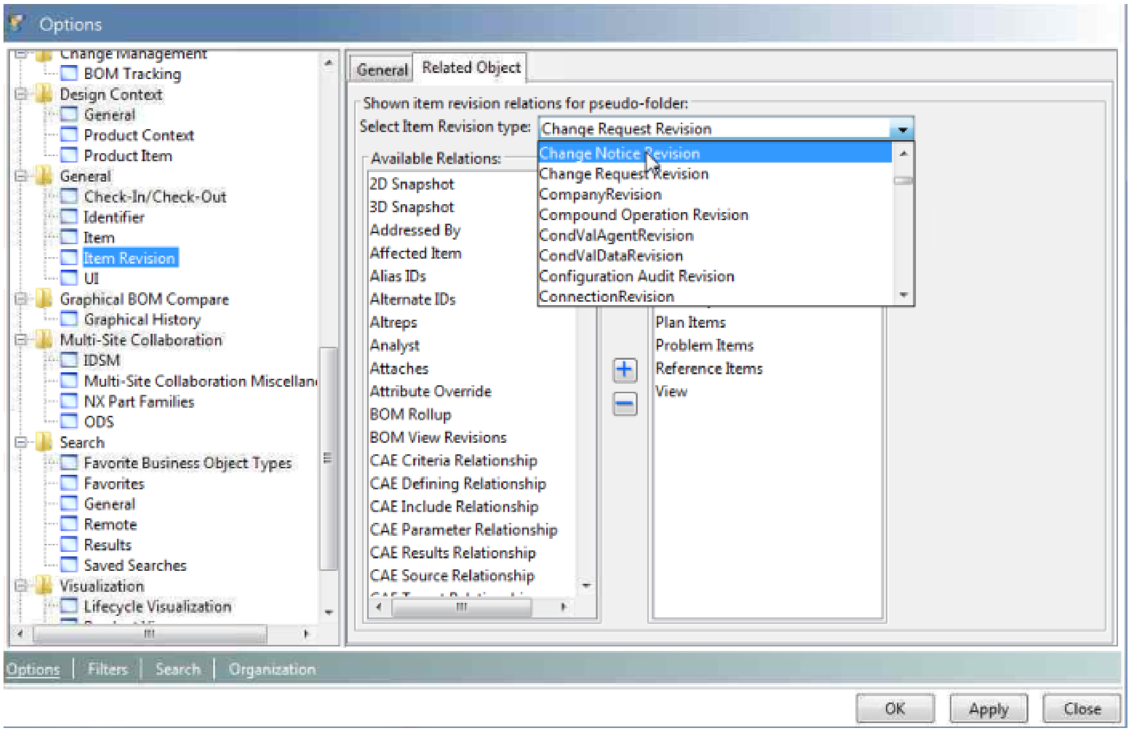
Use the **Options** dialog box to specify that the **Incorporates** property should be displayed for an ECN.

1. Choose **Edit→Options**.

2. In the list on the left, select **General** and then select **Item Revision**.

3. Click the **Related Object** tab.

4. Set **Select Item revision type** to **Change Notice Revision**.



5. In the **Available Relations** list, double-click **Incorporates**.

   Incorporates moves to the **Shown Relations** list.

6. Now click the **General** tab, and set up that the same relations be shown following Steps 4 and 5.

7. Click **OK**.

Viewing Incorporation status and history

View Incorporation status and history

You can view the incorporation status of an engineering change notice (ECN) or an item revision using the **Change History** dashboard of the **Summary** tab.

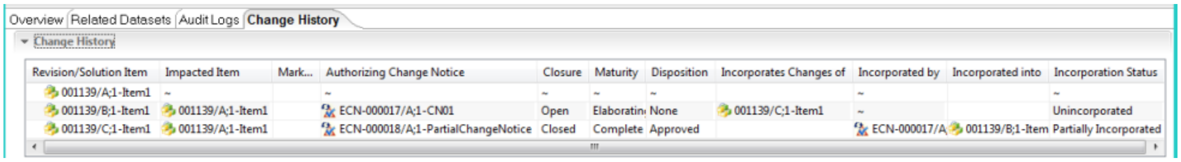
> Note:
>
> The Change History dashboard must be configured.

1. In My Teamcenter or Change Manager, select an item or ECN.

2. Click the **Summary** tab.

3. Click the **Change History** tab.

   The following shows the **Change History** dashboard when an item revision is selected.



Revision/Solution Item	Impacted Item	Mark...	Authorizing Change Notice	Closure	Maturity	Disposition	Incorporates Changes of	Incorporated by	Incorporated into	Incorporation Status
001139/A;1-Item1	~		~	~	~	~		~		~
001139/B;1-Item1	001139/A;1-Item1		ECN-000017/A;1-CN01	Open	Elaboratin	None	001139/C;1-Item1	~		Unincorporated
001139/C;1-Item1	001139/A;1-Item1		ECN-000018/A;1-PartialChangeNotice	Closed	Complete	Approved		ECN-000017/A	001139/B;1-Item	Partially Incorporated

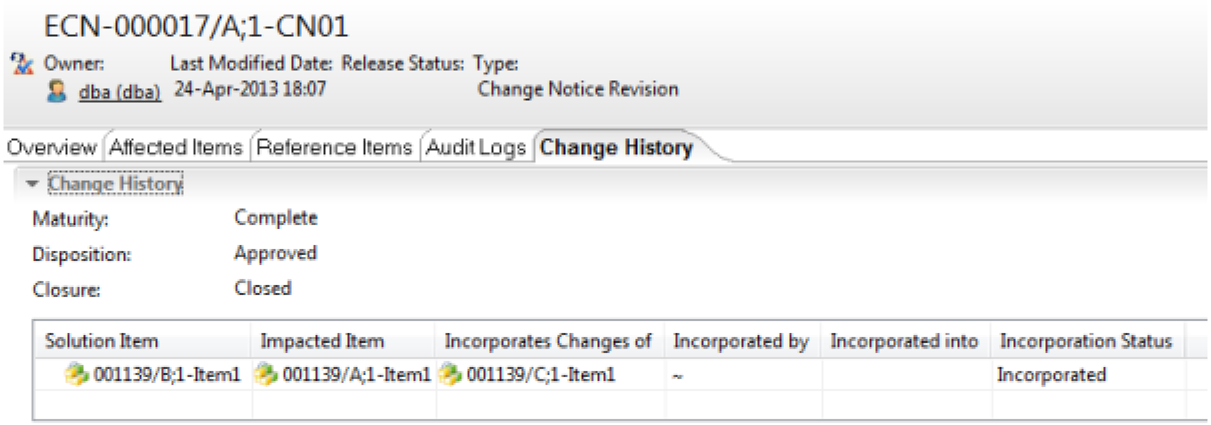
Each row in the dashboard represents one revision of the item or the ECN authorizing the revision (no two rows represent the same revision). The columns in the dashboard change depending on whether you selected to view the incorporation history of an item revision or ECN:

- Columns in the Change History dashboard for an item

- Columns in the Change History dashboard for an ECN

Understanding the columns in the Change History dashboard for an ECN

When you select to view the Incorporation status of an engineering change notice (ECN) in the **Change History** tab of the **Summary** tab, the **Change History** dashboard appears with the following columns.



ECN-000017/A;1-CN01

Owner: dba (dba)

Last Modified Date: 24-Apr-2013 18:07

Release Status:

Type: Change Notice Revision

Maturity: Complete

Disposition: Approved

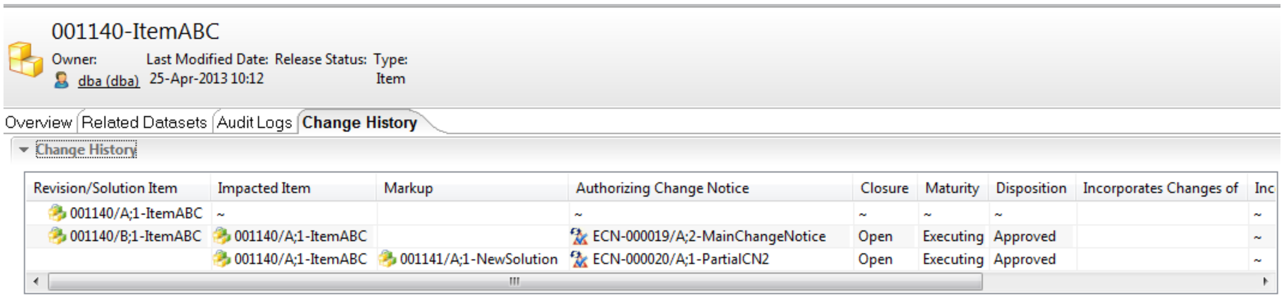
Closure: Closed

Solution Item	Impacted Item	Incorporates Changes of	Incorporated by	Incorporated into	Incorporation Status
001139/B;1-Item1	001139/A;1-Item1	001139/C;1-Item1	~		Incorporated

The column	Displays
Solution Item	The name of the solution item of the ECN.
Impacted Item	The name of the Impacted Item in the ECN.
Incorporates Changes of	The names of the item revisions or markups that the ECN is incorporating.
Incorporated by	The name of the ECN revision that is incorporating the object listed in **Solution Item** column.
Incorporated into	The Incorporation property of the ECN revision.
Incorporation Status	The Incorporation status of the impacted item, except if the Incorporation status is set to **Partially Incorporated**. Then, this indicates that another ECN has incorporated the change fully.

Understanding the columns in the Change History dashboard for an item

When you select to view the Incorporation status of an item in the **Change History** tab of the **Summary** tab, the **Change History** dashboard appears with the following columns.



001140-ItemABC

Owner: dba (dba)
Last Modified Date: 25-Apr-2013 10:12
Release Status:
Type: Item

Revision/Solution Item	Impacted Item	Markup	Authorizing Change Notice	Closure	Maturity	Disposition	Incorporates Changes of	Inc
001140/A;1-ItemABC	~		~	~	~	~		~
001140/B;1-ItemABC	001140/A;1-ItemABC		ECN-000019/A;2-MainChangeNotice	Open	Executing	Approved		~
	001140/A;1-ItemABC	001141/A;1-NewSolution	ECN-000020/A;1-PartialCN2	Open	Executing	Approved		~

The column	Displays
Revision/Solution Item	The ID and name of the solution item.
Impacted Item	The ID and name of the impacted item.
Markup	The name and ID of any markup.
Authorizing Change Notice	The name of the ECN revision.
Closure	The Closure of the ECN.
Maturity	The Maturity of the ECN revision.
Disposition	The Disposition of the ECN revision.

The column	Displays
Incorporates Changes of	IDs of the item revision or the name of the markup being incorporated by the ECN revision.
Incorporated by	The name of the incorporating ECN revision.
Incorporated into	The ID of the revision the change was incorporated into.
Incorporation Status of Change	Shows the Incorporation status set on the impacted item, except if the Incorporation status is set to **Partially Incorporated**. Then, this indicates that another ECN has incorporated the change fully.

Configure the Change History dashboard

1. Associate the **Change History** tab with more workspace objects.

   Change Manager associates the **Change History** tab with all standard workspace objects involved in a change management process, such as item revisions and change objects. If you want to display the **Change History** dashboard for other workspace objects, such as document revisions, you can modify the style sheet for the **Summary** view of the object by adding the following code.

   ```
       <page titleKey="tc_xrt_ChangeHistory" title="Change

   History" visibleWhen="{pref:CM_Change_History_Enable}==true">

           <section titleKey="tc_xrt_ChangeHistory"

   title="Change History" initialstate="expanded">

               <customPanel

   java="com.teamcenter.rac.cm.views.ItemChangeHistoryPanel"

   js="displayItemCustomPanel"  />

           </section>

       </page>
   ```

   If you have modified the style sheet of a subtype of **Item/ItemRevision**, the **Change History** would not appear in its **Summary** view, and you need to add the code shown.

2. Turn on and off the display of **Change History** dashboard.

   Use the **CM_Change_History_Enable** preference to turn on or off the display of the **Change History** dashboard.

3. Change the columns in the dashboard and their order.

   - **CM_Item_ChangeHistory_Columns**

     Changes the columns displayed for an item revision.

   - **CM_CNR_ChangeHistory_Columns**

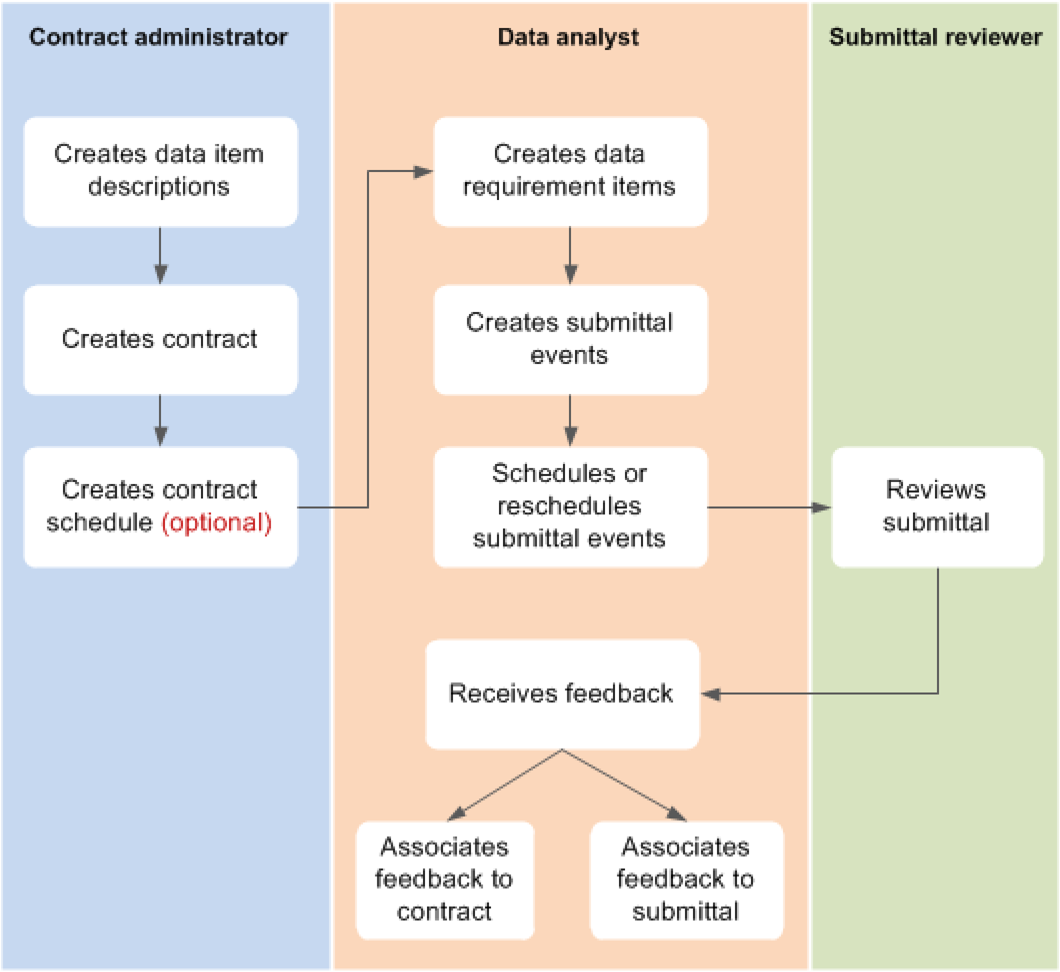
     Changes the columns displayed for an engineering change notice.

8. Managing contract data

Overview of contract data management

In the Aerospace and Defense industry, a contract is a structured procurement document that lists the milestones and schedule dates.

As part of contracts, the contractor must submit data deliverables such as reports and documents. Contract data management helps the contractor manage the creation, review, and delivery of these data deliverables.


Contract administrator
Data analyst
Submittal reviewer
Creates data item descriptions
Creates contract
Creates contract schedule (optional)
Creates data requirement items
Creates submittal events
Schedules or reschedules submittal events
Reviews submittal
Receives feedback
Associates feedback to contract
Associates feedback to submittal

A typical process for managing contract data is as follows:

1. Create data item descriptions

   Data item descriptions (DIDs) are standard forms that specify which data deliverables will be supplied as part of the contract. Contractors submit these forms as part of the response to a contract proposal.

You create DIDs in Teamcenter to keep track of the data deliverables that are promised to the customer.

A DID is associated with a DRI.

2. Create a contract

   A contract in Teamcenter represents the contract you are working on. It has schedules that contain tasks and milestones.

3. Create a contract schedule

   You can create a new schedule or use existing schedule templates.

4. Create data requirement items

   A DRI represents a single data deliverable that the contractor must supply to the customer.

   For each DRI, the contractor must submit documents or submittals at predefined intervals.

   A DRI is always associated with a contract.

5. Create submittal events

   Submittal events specify when and which submittals must be sent to the customer. Submittal events are defined in the DID.

6. Schedule a submittal delivery

   After creating submittal events, you can create a schedule for submittal delivery. When you create a schedule, the submittal items are created automatically.

7. Review submittals

   The submittal delivery schedule is based on a workflow template. The submittal tasks are routed for review and approval when you generate the submittal delivery schedule.

8. Send data deliverables to customers

   Data deliverables are the submittals that are sent to the customer in physical or electronic form.

9. Receive feedback

   You can store the feedback that you receive from the customer in a correspondence item. You can associate this feedback to the contract and the submittal.

Create a data item description

Data item descriptions (DIDs) define the data content, data format, preparation instructions, and intended use. Typically, a DID is created to represent the document that you submit to the customer, declaring the data deliverables you intend to deliver.

1. Choose **File→New→Item**.

2. In the **New Item** dialog box, select **Data Item Description** and then click **Next**.

3. Specify information for the new item:

DID Type	Specifies an attribute for classifying the document item.
Description	Describes the DID. Use a description that helps to distinguish the purpose of the DID.
Name	Specifies the identity of the DID and its revision.

4. Click **Next**.

5. Define additional item revision information as follows:

Document Author	Specifies the DID author.
Document Subject	Specifies the DID subject.
Document Title	Specifies the title of the DID.

6. Click **Finish**.

Associate documents with data item descriptions

You can associate documents that you sent to the customer by attaching the forms to the data item descriptions (DIDs).

1. Choose **File→New→Dataset**.

2. Right-click and copy the dataset.

3. Select the DID revision item and choose **Edit→Paste Special**.

4. In the **Paste Special** dialog box, choose the appropriate attribute and then click **OK**.

Create a contract

A contract is signed between a customer and a supplier. Each contract defines a set of required deliverables, such as a purchase order or other procurement documentation.

The administrator creates a contract object in Teamcenter, defines a master schedule with milestones, and associates this schedule with the contract. The fields marked with an asterisk are required.

1. From My Teamcenter, choose **File→New→Item**.

2. In the **New Item** dialog box, select **Contract** and click **Next**.

3. Define information for the contract:

   Contract Category

   Specifies the type of contract. The following options are available:

   - **CONTRACT**

     Represents any contract documents used for procurement.

   - **TWO**

     Represents a temporary work order.

   - **PO**

     Represents a purchase order.

   Description

   Describes the contract item. Use a description that helps to understand the purpose of the item.

   Name

   Specifies the identity of the contract.

   If naming rules are configured at your site, you can click the **Assign** button to automatically generate the contract ID and the revision.

4. Click **Next**.

   In the **Contract Revision** dialog box, define additional item revision information as follows:

   Comments

   Are used for providing remarks related to the contract.

   Contract Cost

   Specifies the total cost of the contract. It also includes the addition to the cost delta resulting from a modification.

Contract Cost Delta	Specifies the relationship between the current contract cost and the inherent future contract cost that may arise due to modification of the contract.
Contract Event schedule	Specifies the milestones for the contract.
Contract Pricing Model	Specifies the payment forms for the contract, for example, the cost plus a fixed fee.
Document Author	Specifies the name of the person who has drafted the contract.
Document Subject	Specifies the content of the contract.
Document Title	Specifies the title of the procurement document.
Effective Date	Specifies the date from which the contract is effective.
Office Primary Resp	Specifies the name of the person or a role responsible for the review and acceptance of the contract.
Period of Performance	Specifies the duration of the contract, which is the time period specified in the contract (in months).
Recurring Cost	Specifies whether the contract cost is a recurring cost.
SOW Affected	Specifies whether the statement of work for the contract is affected due to contract modifications.

5. Click **Next**.

6. Define the options for the contract and then click **Finish**.

Schedule contract events

1. Right-click the contract revision and choose **Add Contract Event Schedule**.

2. In the **Add Contract Event Schedule** dialog box, select a schedule template from the **Schedule Template** list to use an existing schedule. You can leave this field blank to add tasks to the schedule later.

    Note:

    To display all the templates, set the value of the **Cdm0UseCurrentProjSchTemplates** global constant to **False**.

    Set the value to **True** if you want to display only specific schedule templates related to the project.

Create a data requirement item

A data requirement item (DRI) represents a data deliverable that the contractor must supply to the customer. A contract deliverable can be submitted either by the customer or by the supplier.

1. Choose **File→New→Data Requirement Item**.

2. In the **Data Requirement Item** dialog box, select **Data Requirement Item** and click **Next**.

   Define the basic information for the new item as follows:

   ID / Revision - Name

   Specifies the identity of the DRI and the DRI revision. This is a mandatory field.

   > Note:
   >
   > If naming rules are configured at your site, you can click the **Assign** button to automatically generate the data requirement item ID and revision.

   Description

   Describes the DRI.

3. Click **Next**.

   Define additional DRIs as follows:

   Contract Line Item Number

   Specifies the section ID in the contract. This section describes how a specific task must be accomplished.

   Contract Reference

   Specifies the contract ID. The DRI is associated with the contract ID specified.

   Provide Contract Deliverable

   Specifies the type of contract data deliverable that is submitted by the customer or the supplier.

   Select **True** to provide the contract deliverable to the supplier for review.

   Select **False** to receive the contract deliverable from the supplier.

   Reference Citation

   Specifies the ID of specific reference content in the document.

   Submittal Document Category

   Specifies who delivers the submittals.

4. Click **Next**.

Note:

The following happens when you click **Next**, **Back**, or **Finish** and have configured multifield keys for contracts:

If there are multiple contracts matching the criteria for contracts you specified in the **Define additional item information** dialog box, a dialog box is displayed. It contains a list of contracts matching the criteria you specified for technical documents. To select a contract, double-click the row containing the relevant contract.

5. Define additional DRI revision properties as follows:

Addressee Refers to the reviewers of the contract.

Approval Code Specifies the approval code for the DRI revision.

Approval Required Specifies whether the customer must approve the data item.

Cert Required Specifies whether the submitted data item requires certification.

Comments Specifies any comments concerning the item.

DID Title Specifies the title of the document.

Data Management Due Offset Specifies the offset of the data management due date.

Data Requirement Item Subtitle Specifies the subtitle of the DRI.

Delivery Address Specifies the delivery address of the customer or the supplier.

Delivery Mechanism Specifies the mode of delivery for the contract.

Distribution Required Specifies whether any documentation related to the item must be distributed.

Effective Cutoff Date Specifies the last date for data collection.

Effective Cutoff Event Desc. Provides additional information about the last date for data collection.

Event List Specifies the event table for generating an event schedule.

Exhibit Name Specifies the ID of a particular data item in a data item description document.

Media Type Specifies the format of the data, for example, electrostatic or microfilm.

OPR Notif Due Offset Specifies the last date for creating the notification to the office of the primary reviewers.

Office Primary Response	Specifies the name of the contacts, groups, or role responsible for reviewing the data item.
Office Primary Response Notification Due Offset	Specifies the offset to the response due date. The notification will be sent based on the dates calculated based on this offset.
Response Due Offset	Specifies the offset to the response due date.
Shipping Document Reqd	Specifies whether DD Form 250 is required for the delivery of the data item.
Submit Flow Thru	Specifies whether a supplier's DRI must be submitted as a customer's DRI.
Submit Purpose	Specifies the purpose of the submitted data item.
Submit Quantity	Specifies the number of copies required for submission.

6. Click **Next**.

   Define the options and then click **Finish**.

Update a data requirement item

1. Navigate to the data requirement item (DRI) revision.

2. Right-click the DRI revision and choose **Check-In/Out→Check Out**.

3. In the **Check-Out** dialog box, fill in the relevant fields and then click **OK**.

4. Click the **Viewer** tab to display the DRI.

5. The **Event List** section specifies the event table for generating an event schedule. Based on the data defined in the event table, the submittal delivery schedule is generated.

   Update the **Event List** section as follows:

   Event List

   Click the **Add** button to add a row to define the following fields:

   - **Contract Schedule Tasks**

     Lists the contract tasks. Choose the contract task that you want to associate with the Data Requirement Item.

   - **Summary Task Name**

Denotes the submittal delivery event name.

- **Start Date**

  Denotes the event start date.

- **End Date**

  Denotes the event end date.

- **Offset**

  Denotes the offset used for calculating the submittal due date in order to calculate the first submittal delivery.

- **Relative To**

  Denotes the relation of the offset to either the start date or the end date.

- **Recurrence**

  Denotes frequency of the submittal delivery event.

- **Recurrence End Date**

  Specifies the last date for event schedule.

Process Template

Specifies the workflow template to use when generating submittals.

Task Duration Hours

Specifies the duration of the submittal task in hours.

Align Task

Specifies how to calculate the due dates of the submittal tasks.

Submittal Type

Specifies the type of submittal, for example, whether it is from a supplier or internal.

6. Click **Check In**.

Note:

- You must save any modification to the DRI before navigating to other objects.

Create submittal delivery events for a data requirement item

The supplier must submit documents on specified dates. You can define the submittal delivery events for the data requirement item (DRI) after creating the DRI. The data from the submittal delivery events is used to generate a preliminary submittal delivery schedule.

1. Click **Add** in the **Event List** section.

2. Add the following information in the event row:

   Contract Schedule Tasks
   Lists the tasks that are part of the contract schedule. Choose the contract task that you want to associate with the DRI.

   Summary Task Name
   Denotes the submittal delivery event name.

   Start Date
   Denotes the start date for the event.

   End Date
   Denotes the end date for the event.

   Offset
   Denotes the offset used for calculating the submittal due date for calculating the first submittal delivery.

   In some cases, even when the submittal is ready, you may need some extra time for submitting the deliverables. The offset in such cases is the time added to the deliverable submission date or the submittal due date.

   Relative To
   Denotes the relation of the offset to either the start date or the end date.

   Specifies whether the starting point for the submittal generation should be the start date or the end date of the event

   Recurrence
   Denotes the frequency of the submittal delivery event, for example, once a month or once in a quarter.

   Recurrence End Date
   Specifies the last date for the recurrence of the submittal event schedule.

3. Set scheduling properties for a submittal event:

   Process Template
   Specifies which workflow template to use to specify what needs to be done with the submittals once they are generated.

   Task Duration Hours
   Specifies the duration of the submittal task in hours.

   Align Task
   Specifies whether to align the submittal due date with either the start date or the end date of the submittal event.

	Align task specifies whether the submittal delivery starts on the date calculated using the submittal event attributes or whether it ends on the calculated date.
Submittal Type	Specifies the object type of the deliverable.

Generate a submittal delivery schedule

The data analyst modifies the data requirement item (DRI) revision to define the submittal delivery events. This data is then used by the system to generate a draft submittal delivery schedule.

The submittal is automatically created after the data analyst generates the submittal delivery schedule. You can then modify this autogenerated schedule, if required.

1. Right-click the DRI revision item and choose **Generate Submittal Delivery Schedule**.

If you have specified offsets for the data management due date, OPR notification due date, and the response due date in the DRI, the due dates are calculated based on these offsets.

Create submittals manually

You can manually create a submittal schedule. The submittal monitors and notifies the customer or the supplier regarding the pending tasks and what is required to be delivered for each task assigned to them. After you create a submittal, you must attach it to the submittal delivery schedule task.

1. From My Teamcenter, choose **File→New→Item**.

2. In the **New Item** dialog box, select **Submittal** and then click **Next**.

3. Specify the name of the submittal and click **Next**.

4. In the **Cdm0SubmittalMaster** panel, specify the following information:

Author	Specifies the name of the person, group, or the role who initiated the submittal.
Keywords	Specifies the keywords that can be used later to search for the submittal.
Subject	Specifies the subject of the submittal.
Title	Specifies the title of the submittal.

5. Click **Next**.

6. In the **SubmittalRevision** panel, specify the following information:

DRI Item Type	Specifies whether the data item requirement is from a customer or a supplier.
Disposition	Specifies the unique identifier for DD form 250, which is prepared for contract deliveries.
Disposition Date	Specifies the disposition or the incorporation date of a change or a submittal.
Document Author	Identifies the author of the DRI.
Document Subject	Specifies the subject of the DRI.
Document Title	Specifies the title of the DRI.
Office Primary Resp	Specifies the person, group, or the role responsible for reviewing and acknowledging the data item submittal.
Received Date	Specifies the date on which an item is received from the supplier or the date on which the item is sent to a customer.
Resubmit Comments	Specifies the customer's comments during resubmittal.
Resubmit Due Date	Specifies the due date of sending the contract data item to the customer.
Resubmit Required	Specifies whether the data item must be submitted again.
Shipping Document Required	Specifies the ID of DD Form 250 that is prepared to support contract deliveries.
Submit Due Date	Specifies the due date of the contract data item to the customer.
Submittal Type	Specifies the data item submittal stage, for example, draft or final.

7. Click **Finish**.

Associate manually created submittals with schedule deliverables

You can relate the data that is to be delivered to the submittal through Schedule Manager. You must first associate the submitted deliverable to a schedule and then initiate a task.

1. Open Schedule Manager to locate the submittal object.

2. Select the submittal object and then choose **Schedule→Schedule Deliverables**.

3. In the **Schedule Deliverables** dialog box, click the **Add** button to define the following:

   Name

Specifies the name of the schedule deliverable.

Type

Select **Submittal**.

Deliverable

Specifies the submittal object that is associated with a Data Requirement Item (DRI).

Click the button to search for the deliverable.

4. Click **OK**.

5. Expand the submittal object and select the task object.

6. Choose **Schedule→Task Deliverables**.

7. In the **Task Deliverables** dialog box, click the **Add** button to define the following:

   Schedule Deliverable

   Specifies the name of the schedule deliverable.

   Submit Type

   Specifies the type of task.

   Target

   Select this to attach the deliverable as a target attachment to the task workflow.

   Reference

   Select this to attach the deliverable as a reference attachment to the task workflow.

   Do Not Submit

   Select this when no deliverable is attached to the task workflow.

   Deliverable

   Specifies the submittal object that is associated with a DRI.

   Click the button to search for the deliverable.

8. Click **OK**.

9. Double-click the task deliverable object to define the workflow process.

10. In the **Properties** dialog box, edit the schedule tasks to define the workflow process.

    You can select a workflow template from the **Workflow Template** list.

11. Click **OK**.

Data Requirement Item rescheduling rules

You can use Data Requirement Item (DRI) rescheduling to update the dates in the DRI event list and then generate the submittal delivery schedule again.

You can reschedule the DRI at the *contract level* or at the *DRI level*. Rescheduling at a contract level updates all the impacted DRI event lists.

When rescheduling at the DRI level, you can choose the DRI events to reschedule.

The following rules apply when you reschedule a DRI:

- The submittal tasks that are complete or are in progress will not be rescheduled.

- The submittal tasks with a start date earlier than the current date will not be rescheduled.

- The submittal tasks that are yet to start and with a start date in the future will be rescheduled.

- If the start date of the submittal tasks predates the current date, the tasks will be rescheduled to start on the current date.

- If the event list in the DRI is deleted, all the submittal tasks that are yet to start will be deleted.

- If the DRI event list is not mapped to the contract schedule event list, all the submittal tasks yet to start will be deleted.

- If you manually change the DRI schedule in Schedule Manager, the corresponding event list will not be updated.

If the value of the **Cdm0UseFixedSubmittalDueDates** preferences is **true**, the submittal due dates will be calculated based on the company working time. For example, if a submittal due date falls on a holiday, the due date is adjusted to the previous working day. If the adjusted due date is also a holiday, the due date is moved to the next available working day.

For example, consider a monthly frequency for the submittal. If the starting submittal due date is January 1, the next submittal due date is February 1. If February 1 is a holiday, January 31 is chosen as the submittal due date. If January 31 is a holiday, the next available working day in February is chosen as the due date.

The **Generate Submittal Delivery Schedule** functionality will now update the submittal delivery schedule if the DRI event list is changed.

Reschedule a data requirement item

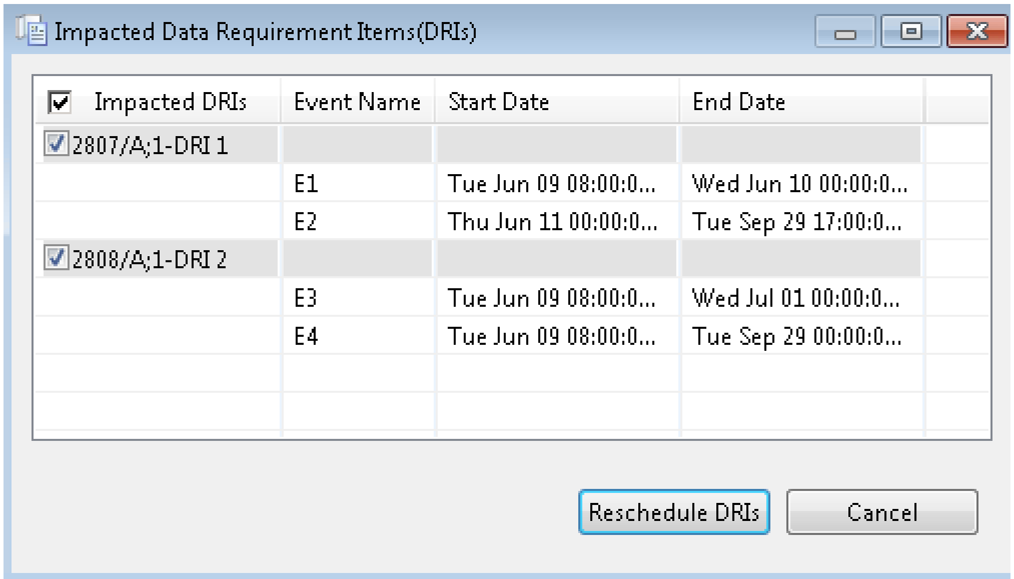
You can reschedule a data requirement item (DRI) at the DRI level or at the contract level. Rescheduling at the contract level provides you the option of rescheduling all the DRIs in the contract.

The Generate Submittal Delivery Schedule functionality reschedules the submittal delivery schedule if the DRI event list has changed.

To reschedule DRIs at the contract level, do the following:

1. Right-click the contract revision and choose **Reschedule Data Requirement Items**.

2. In the **Impacted Data Requirement Items(DRIs)** dialog box, choose the DRIs you want to reschedule.



☑ Impacted DRIs	Event Name	Start Date	End Date
☑ 2807/A;1-DRI 1			
	E1	Tue Jun 09 08:00:0...	Wed Jun 10 00:00:0...
	E2	Thu Jun 11 00:00:0...	Tue Sep 29 17:00:0...
☑ 2808/A;1-DRI 2			
	E3	Tue Jun 09 08:00:0...	Wed Jul 01 00:00:0...
	E4	Tue Jun 09 08:00:0...	Tue Sep 29 00:00:0...

3. Click **Reschedule DRIs**.

4. Click **OK** in the **Rescheduling Summary** dialog box.

The DRIs are rescheduled as per the DRI rescheduling rules.

Create a correspondence

The data analyst creates a correspondence in Teamcenter and attaches the feedback documents to the correspondence object.

1. From My Teamcenter, choose **File→New→Item**.

2. In the **New Item** dialog box, select **Correspondence** and click **Next**.

3. Define basic information for the correspondence as follows:

    Name Identifies the correspondence.

    Description Describes the correspondence item. Use a description that helps you distinguish the purpose of the correspondence item.

    Correspondence Type Specifies a category of similar technical documents, for example, assembly drawings, wire list, or material specifications.

4. Click **Next**.

5. In the **CorspondenceRevision** dialog box, define additional information as follows:

    Category Specifies the correspondence category, for example, memo.

    Correspondence Direction Specifies whether the correspondence is incoming or outgoing.

    Document Author Identifies the name of the person or the role who initiated the correspondence.

    Document Subject Specifies the subject of the correspondence.

    Document Title Specifies the title of the correspondence.

    Priority Specifies the priority level for accomplishing the task.

    Received Org Name Specifies the name of the organization receiving the correspondence.

    Received or Sent Date Specifies the date on which the correspondence was received or sent from the supplier or the customer.

    Reference Specifies the list of references.

    Reply Requested Date Specifies the date by which a reply is expected.

    Reply Required Specifies whether a reply is required or not.

    Responsible Party Specifies the person or roles responsible for replying.

6. Click **Finish**.

Associate the correspondence revision with a submittal revision

Communication related to the scheduled tasks must be associated with a submittal revision.

1. Right-click the correspondence revision item and choose **Copy**.
2. Navigate to the submittal revision item.
3. Select the submittal revision and choose **Edit→Paste Special**.
4. In the **Paste Special** dialog box, select **Correspondences** from the **Add As:** list.
5. Click **OK**.

Associate a correspondence with a contract revision

All communication related to the contract must be associated with a correspondence revision.

1. Right-click the contract revision and choose **Copy**.
2. Select the correspondence revision and choose **Edit→Paste Special**.
3. In the **Paste Special** dialog box, select **Contracts** from the **Add As:** list.
4. Click **OK**.

9. Managing stock materials

Basic concepts about stock materials

In the Aerospace and Defense industry many parts are made from stock materials such as bar stock, tubing stock, and sheet stock. You can manage stock materials in Teamcenter by:

- Creating a library of stock materials.

- Making the stock material the preferred stock material of a program.

- Assigning the stock material to a part.

The Aerospace and Defense industry typically manages stock materials as follows:

- The Teamcenter administrator creates a classification hierarchy or stock material library using the Classification application.

- The materials manager creates stock materials and adds them to the stock material library.

- The program administrator specifies the stock materials that are preferred to the program. By doing this, only approved stock materials can be used in the program.

- The design engineer creates a part and associates the part with the stock material and specifies the dimensional properties of the stock material.

Install stock materials

1. Start Teamcenter Environment Manager (TEM).

2. Select the appropriate options until you reach the **Select Features** panel.

3. In the **Select Features** panel, expand the **Extensions** navigation tree.

4. Under **Enterprise Knowledge Foundation**, select **Stock Materials** to install the stock material management functionality.

5. Click **Next**.

6. Select the appropriate options until you reach the **Business Modeler IDE Templates** panel.

7. In the **Business Modeler IDE Templates** panel, select **Stock Materials**.

8. Click **Next** and select the appropriate options until you complete the installation.

9. After completing the TEM installation, install the Web tier and deploy the web applications.

Configure the display of the Made From folder

To add the **Made From** relation to an object and add the **Made From** folder to an object:

1. In My Teamcenter, choose **Edit→Options**.

   The **Options** dialog box is displayed.

2. Choose **General→Item Revision**.

3. Click the **General** tab and do the following:

   a. From the **Item Revision type** list, select the object that must have the **Made From** relation, for example, **ItemRevision** or **Part Revision**.

   b. From the **Available Relations** list, choose **Made From** and add it to the **Shown Relations** list.

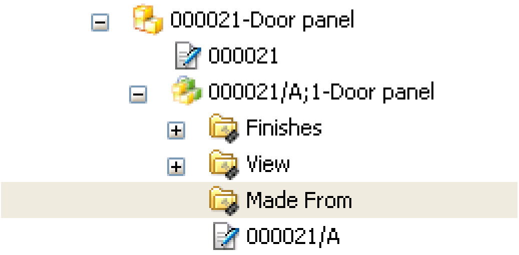
4. Click the **Related Object** tab and do the following:

   a. From the **Item Revision type** list, select the object which must have the **Made From** folder, for example, **ItemRevision** or **Part Revision**.

   b. From the **Available Relations** list, choose **Made From** and add it to the **Shown Relations** list.

5. Click **OK**.

The **Made From** relation and folder is added to the object.



Creating a classification library for stock materials

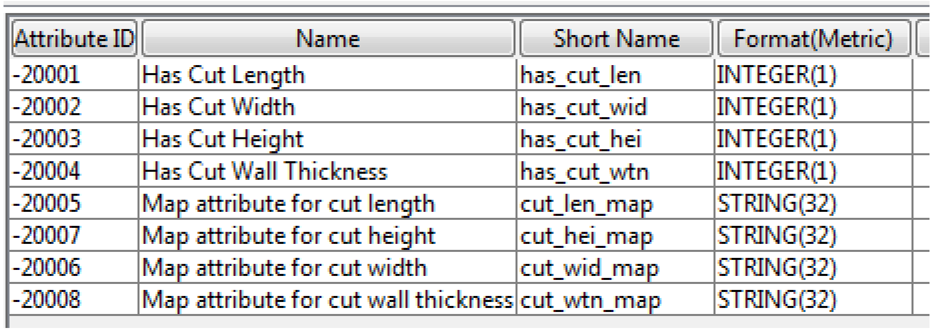
Use the Classification Admin application to create a stock material library.

- In Classification Admin, create a class.

Ensure that the parent class is an abstract class and that the child classes are storage classes.

- To add attributes to the class, click **Add Attributes**.

  The following classes are available by default:



Attribute ID	Name	Short Name	Format(Metric)
-20001	Has Cut Length	has_cut_len	INTEGER(1)
-20002	Has Cut Width	has_cut_wid	INTEGER(1)
-20003	Has Cut Height	has_cut_hei	INTEGER(1)
-20004	Has Cut Wall Thickness	has_cut_wtn	INTEGER(1)
-20005	Map attribute for cut length	cut_len_map	STRING(32)
-20007	Map attribute for cut height	cut_hei_map	STRING(32)
-20006	Map attribute for cut width	cut_wid_map	STRING(32)
-20008	Map attribute for cut wall thickness	cut_wtn_map	STRING(32)

- You can also add custom attributes using the **Dictionary** feature.

Create stock materials

1. From My Teamcenter, choose **File→New→Item**.

2. Select **Stock Material** from the **New Item** dialog box.

   Click **Next**.

3. In the **Object Create Information** dialog box enter the following information:

   Configuration Item Specifies if the stock material is a configuration item.

   Description Describes the stock material.

   ID Specifies the stock material ID.

   (Optional) Click the **Assign** button to auto assign the ID.

   Name Specifies the name of the stock material.

   Unit of Measure Specifies the metric for the stock material.

4. Click **Next**.

   Specify additional information about the stock material in the following dialogs.

5. Click **Finish**.

Add stock materials to the stock material library

1. Right-click the stock material and choose **Send To→Classification**.

2. To classify the object, click **Yes** in the **Classify Object** dialog box.

   The object ID is displayed in the **Properties** pane.

3. Browse the hierarchy tree to locate the stock material class that best matches the characteristics of the stock material you are classifying.

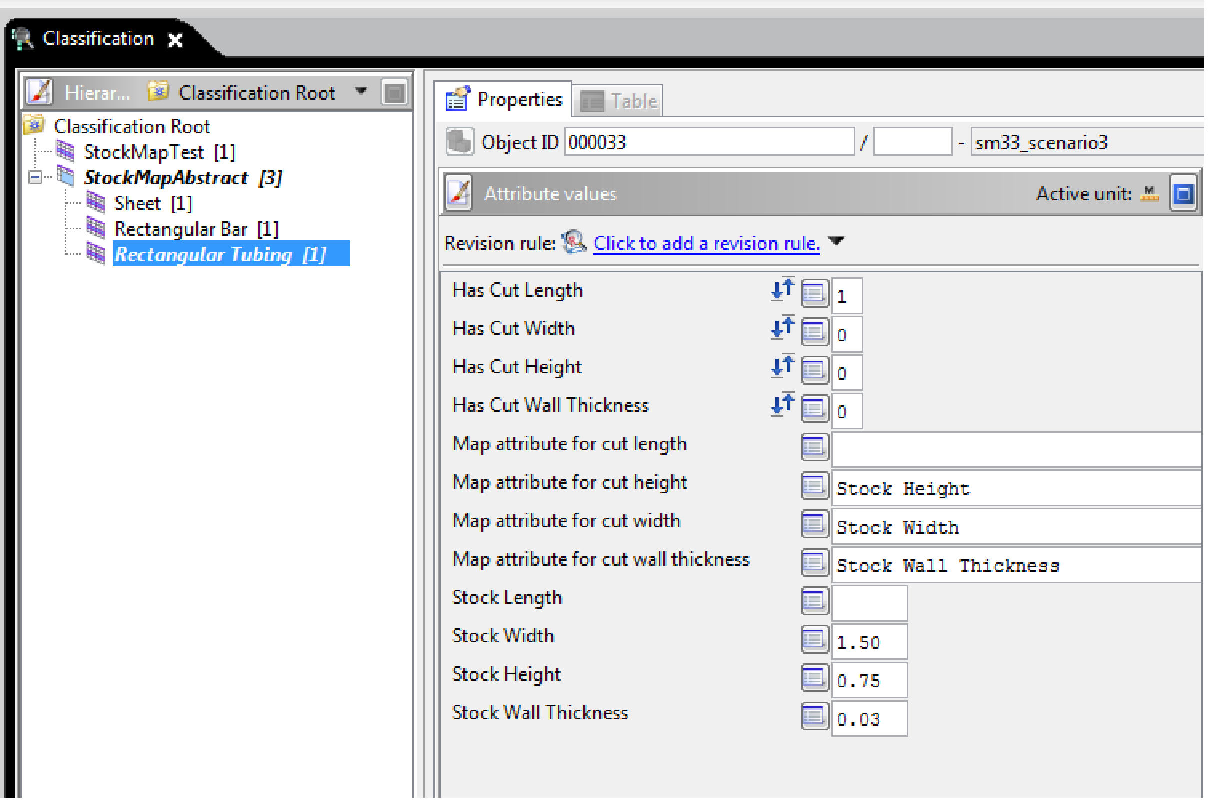
4. Right-click the storage class and choose **Select**. You can also double-click the storage class to display the attributes.

   The attributes associated with the selected stock material library are displayed in the **Properties** pane.

5. Type values for the stock material attributes in the **Attribute values** section.

   You can only enter **Attribute values** if you have added **Attribute values** for the classification class.

6. You can map the default attributes to your custom attributes as follows.



Object ID 000033 / - sm33_scenario3

Has Cut Length	1
Has Cut Width	0
Has Cut Height	0
Has Cut Wall Thickness	0
Map attribute for cut length	
Map attribute for cut height	Stock Height
Map attribute for cut width	Stock Width
Map attribute for cut wall thickness	Stock Wall Thickness
Stock Length	
Stock Width	1.50
Stock Height	0.75
Stock Wall Thickness	0.03

   a. If you want to provide a fixed value of **0.75** to the cut height attribute and ensure that the attribute is not editable in the **Make From** dialog box, update the attributes as follows:

      - **Has Cut Height**: Assign the value **0**. This value ensures that the cut height attribute is not editable in the **Make From** dialog box.

      - **Map attribute for cut height**: Assign the value **Stock Height** to this attribute to map the **Has Cut Height** attribute with the custom attribute **Stock Height**.

      - For the **Stock Height** attribute assign a value of **0.75**.

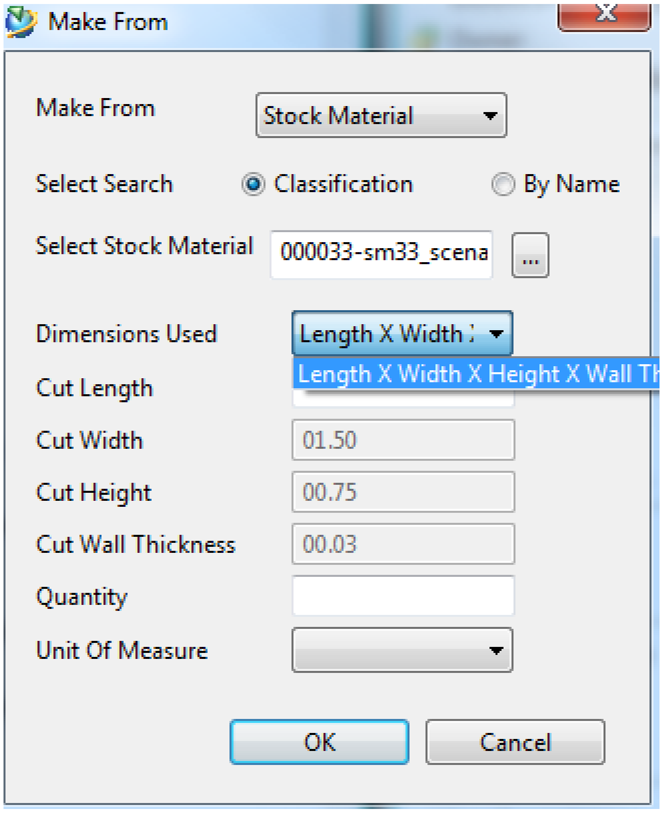
   b. If you want users to provide the cut length value in the **Make From** dialog box, update the attributes as follows:

      - **Has Cut Length**: Assign the value **1**. This value ensures that the cut length attribute is editable in the **Make From** dialog box.

      - It is not necessary to assign a value to the **Map attribute for cut length** attribute.

7. Click the **Save** button on the toolbar to add the stock material to the stock material library.

The following **Make From** dialog box is displayed when the cut attributes are successfully mapped to the custom attributes.



Make From

Make From: Stock Material

Select Search: ● Classification ○ By Name

Select Stock Material: 000033-sm33_scena ...

Dimensions Used: Length X Width :

Length X Width X Height X Wall Th

Cut Length:

Cut Width: 01.50

Cut Height: 00.75

Cut Wall Thickness: 00.03

Quantity:

Unit Of Measure:

As described in the previous graphic, you can edit only the **Cut Length** attribute and not the others.

Specify a preferred stock material for a program

1. In My Teamcenter, select the stock material and choose **Copy**.

2. In the Program application, expand the program where you want to add the stock material.

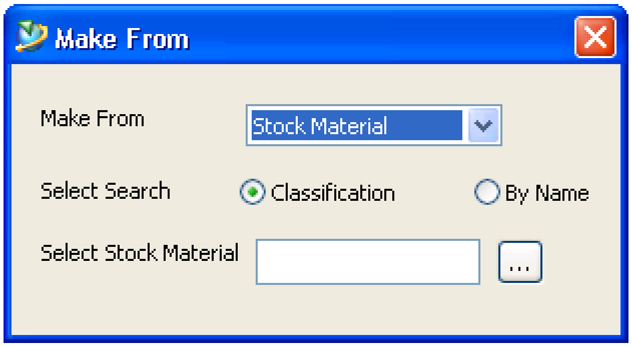
3. Select the **Preferred Items** folder and choose **Paste**.

The stock material is now the preferred stock material of the program.

Assign stock material or standard part to a part

1. In My Teamcenter, right-click the part revision to which you want to add the stock material, and choose **Make From**.

   The My Teamcenter application shows the **Make From** dialog box.



Make From: Stock Material

Select Search: ● Classification ○ By Name

Select Stock Material:

2. Select **Stock Material** or **Part** from the **Make From** list.

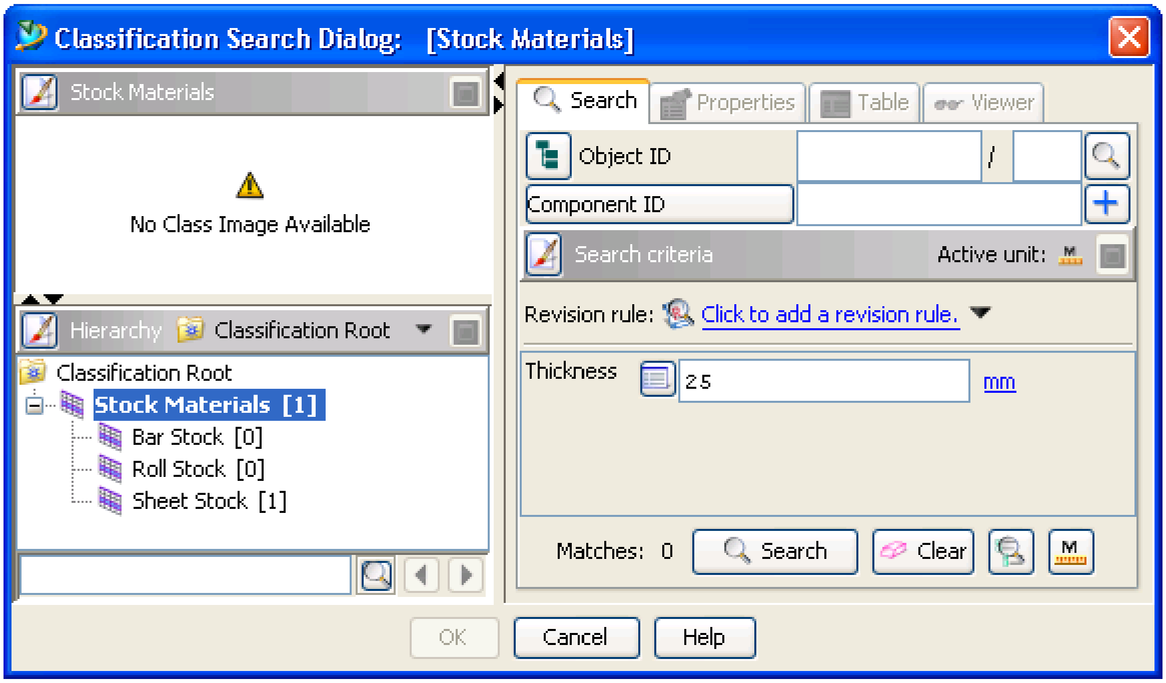
3. Click any one of the following options from **Select Search**:

   a. **Classification**: To search for stock materials in the classification hierarchy.

   b. **By Name**: To search for stock materials by name.

4. Click the button next to the **Select Stock Material** box.

5. If you select the **Classification** option, the **Classification Search** dialog box appears.



Thickness 2.5 mm

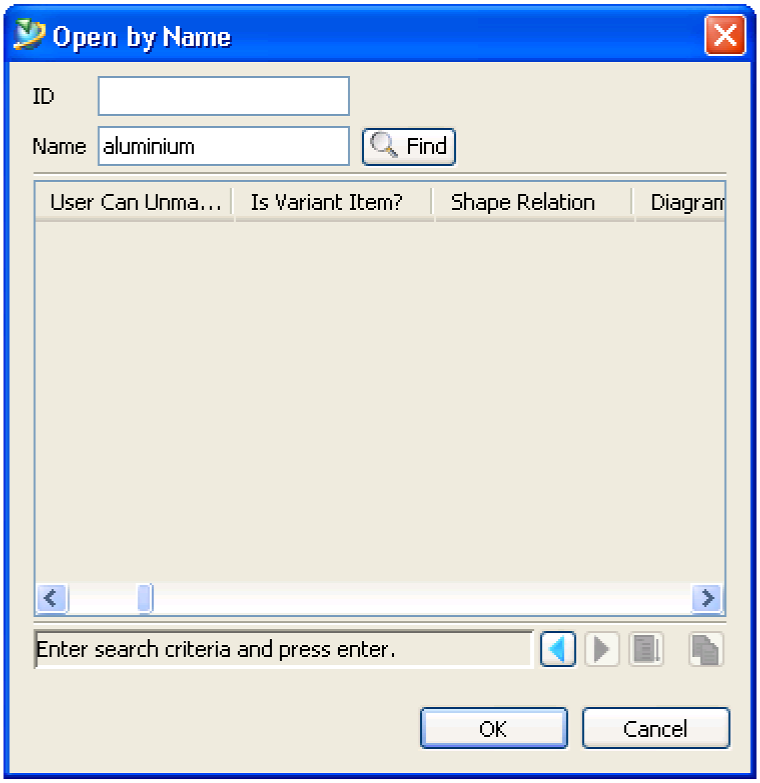
a. Type the search criteria in the search fields and click **Search**.

The search results appear in the **Table** tab.

b. Select the stock material you want and click **OK**.

The stock material appears in the **Select Stock Material** box.

6. If you select the **By Name** option, the **Open By Name** dialog box appears.



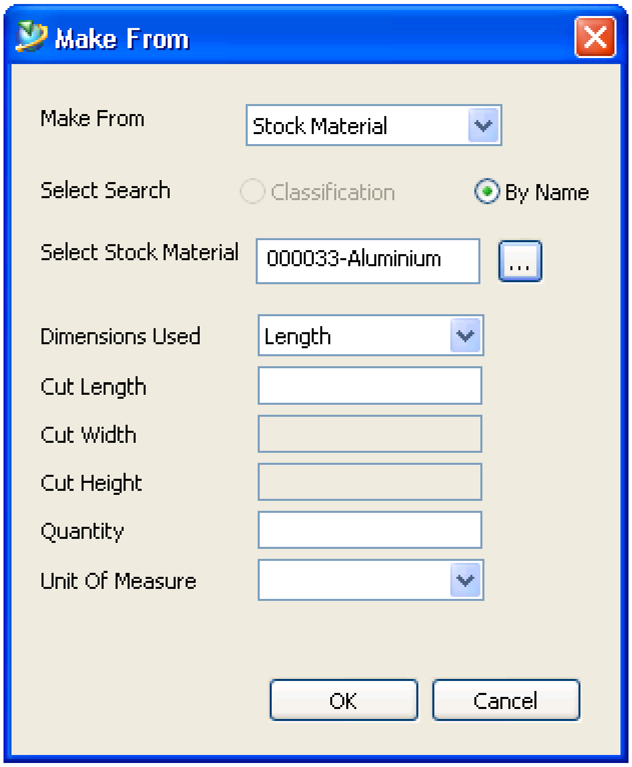
Name: aluminium

Enter search criteria and press enter.

a. Type the name of stock material in the **Name** box and click the **Find** button.

b. Select the stock material you want and click **OK**.

The stock material appears in the **Select Stock Material** box.



Make From: Stock Material

Select Search: ○ Classification ● By Name

Select Stock Material: 000033-Aluminium

Dimensions Used: Length

Cut Length

Cut Width

Cut Height

Quantity

Unit Of Measure

7. In the **Make From** dialog box, type the values for the following dimensions:

   a. **Cut Length**

   b. **Cut Thickness**

   c. **Cut Width**

   d. **Quantity**

8. Type the quantity of the stock material required in the **Quantity** box.

9. Select the unit of measure for the stock material from the **Unit of Measure** list.

10. Click **OK**.

The stock material appears in the **Made From** pseudo folder of the part revision.

The **Make From** list supports only **Stock Material** and **Part**.

The custom properties you add to the **Make From** relation are not available when you assign a stock material using the **Make From** dialog box. To get these custom properties, assign stock materials using the copy-paste special functionality. The custom properties are shown in the **Properties on Relation** dialog box.

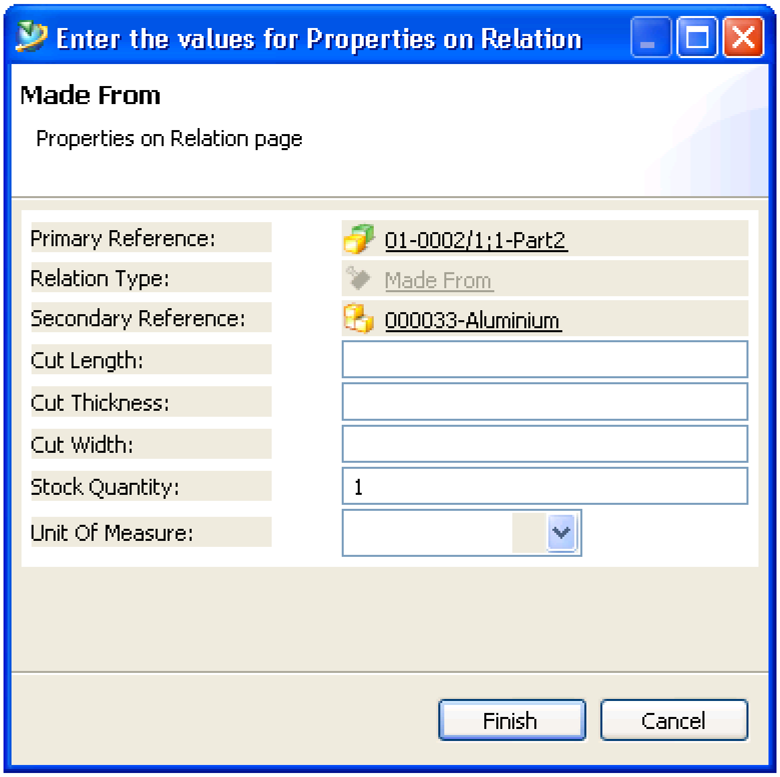
Assign stock materials to parts using the copy functionality

Assign stock materials to parts using the copy-paste functionality

1. Right-click the stock material and choose **Copy**.

2. Right-click the **Made From** folder of the part revision and choose **Paste**.

   The **Made From** dialog box appears.



Enter the values for Properties on Relation

Made From

Properties on Relation page

Primary Reference:	01-0002/1;1-Part2
Relation Type:	Made From
Secondary Reference:	000033-Aluminium
Cut Length:	
Cut Thickness:	
Cut Width:	
Stock Quantity:	1
Unit Of Measure:	

3. In the **Made From** dialog box, type the values for the following dimensions:

   a. **Cut Length**

   b. **Cut Thickness**

   c. **Cut Width**

   d. **Stock Quantity**

4. Select the unit of measure for the stock material from the **Unit of Measure** list.

5. Click **Finish**.

Assign stock materials to parts, using the copy and paste special functionality

1. Right-click the stock material and choose **Copy**.

2. Select the part revision and choose **Edit**→**Paste Special**.

3. From the **Paste Special** dialog, select **Made From**.

   The **Made From** dialog box appears.

4. In the **Made From** dialog box, type the values for the following dimensions:

   a. **Cut Length**

   b. **Cut Thickness**

   c. **Cut Width**

   d. **Stock Quantity**

5. Select the unit of measure for the stock material from the **Unit of Measure** list.

6. Click **Finish**.

Assign stock materials to parts, using the drag drop functionality

1. Drag the stock material to the **Made From** folder of the part revision.

   The **Made From** dialog box appears.

2. In the **Made From** dialog box, type the values for the following dimensions:

   a. **Cut Length**

   b. **Cut Thickness**

   c. **Cut Width**

   d. **Stock Quantity**

3. Select the unit of measure for the stock material from the **Unit of Measure** list.

Teamcenter adds the stock material to the **Made From** folder of the part revision.

10. Managing work packages

Basic concepts about work packages

A *work package* or *package* is typically any collection of CAD files and documentation that an outsourcing partner requires to build, test, or maintain a component or subassembly of a larger product.

A *package* serves as a revisable collection of product information and can be used in a variety of contexts.

The Aerospace and Defense industry typically uses work packages as follows:

- The Designer creates a work package and specifies the work associated with the work package.
- The Designer adds objects to work packages. The objects can have static or dynamic relationship with the work package.

  Objects that have static relationship are generally reference items and do not change over time, for example, a design document.

  Objects that have a dynamic relationship always show the latest revision.
- The Designer submits the work package to a workflow.
- When the work associated with the work package is complete, it is released with a maturity status indicating the completion of work.

  You can use a workflow or change management functionalities to release the work package.

Install work packages

1. Start Teamcenter Environment Manager (TEM).
2. Select the appropriate options until you reach the **Select Features** panel.
3. In the **Select Features** panel, expand the **Extensions** navigation tree.
4. Under **Enterprise Knowledge Foundation**, select **Work Packages** to install the stock material management functionality.
5. Click **Next**.
6. Select the appropriate options until you reach the **Business Modeler IDE Templates** panel.

7. In the **Business Modeler IDE Templates** panel, select **Work Packages**.

8. Click **Next** and select the appropriate options until you complete the installation.

9. After completing the TEM installation, install the Web tier and deploy the web applications.

Create a work package

1. From My Teamcenter, choose **File→New→Item**.

2. Select **Work Package** from the **New Item** dialog box.

   Click **Next**.

3. In the **Object Create Information** dialog box, enter the following information:

Configuration Item	Specifies if the work package is a configuration item.
Description	Describes the work package.
ID	Specifies the ID of the work package. Click the **Assign** button to auto assign the ID.
Name	Specifies the name of the work package.
Unit of Measure	Specifies the metric for the work package.
Work Package Complexity	Specifies the level of difficulty for executing the work package.
Work Package Security	Specifies the security level of the work package.
Work Package Type	Specifies if the work package is executed internally or by a supplier.
Work Package Vendor	Specifies the name of the vendor executing the work package.

4. Click **Next**.

   Specify additional information about the work package in the following dialogs.

5. Click **Finish**.

Add objects to a work package

Objects can have a static or dynamic relationship with the work package.

Static relations show objects with the original revision in the work package. To create a static relationship, add the object to the **Static Contents** folder of the work package.

Dynamic relations always show objects with the latest revision. When you revise the objects outside the work package, the work package will show the latest revision. To create a dynamic relationship, add the object to the **Dynamic Contents** folder.

You can add objects to the work package in the following ways.

Drag the object to the work package

Drag the object to the **Static Contents** or the **Dynamic Contents** folder of he work package.

Copy and paste the object to the work package

Copy and paste the object to the **Static Contents** or the **Dynamic Contents** folder of the work package.

Use copy and paste special functionality to add objects to the work package

1. Right-click the object and choose **Copy**.
2. Select the work package and choose **Edit→Paste Special**.
3. From the **Paste Special** dialog box, choose **Static Contents** to create a static relationship or **Dynamic Contents** to create a dynamic relationship.

11. Managing finishes

Using finishes

A finish represents a finishing process on a part. It may be used to improve appearance, adhesion, corrosion resistance, tarnish resistance, chemical resistance, wear resistance, remove burrs, and so on. For example, you can apply a finishing process on a part by cleaning, priming, and painting it.

You can group the finishes together to create a finish group. You can specify the order of execution of finishes in the finish group. For example, you can create a finish group called Preserve that contains finishes named Clean, Prime, and Paint.

The following is an example process of how finishes are used:

- The finish manager creates a library of finish and finish groups, using the Classification application.

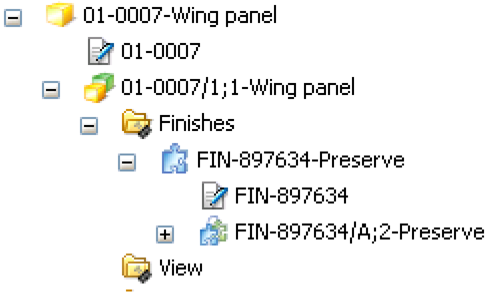
- The finish manager creates a finish.

- The finish manager creates a finish group.

- The design engineer applies the finish and finish group to a part.

- The design engineer applies the finish or finish group to a part occurrence.

- The manufacturing engineer views the finishes and performs the finish job on the part.



```
⊟ 01-0007-Wing panel
      01-0007
  ⊟ 01-0007/1;1-Wing panel
    ⊟ Finishes
      ⊟ FIN-897634-Preserve
            FIN-897634
        ⊞ FIN-897634/A;2-Preserve
      View
```

Install finishes

1. Start Teamcenter Environment Manager (TEM).

2. Select the appropriate options until you reach the **Select Features** panel.

3. In the **Select Features** panel, expand the **Extensions** navigation tree.

4. Under **Enterprise Knowledge Foundation**, select **Finish Management** to install the finish management functionality.

5. Click **Next**.

6. Select the appropriate options until you reach the **Business Modeler IDE Templates** panel.

7. In the **Business Modeler IDE Templates** panel, select **Finish Management**.

8. Click **Next** and select the appropriate options until you complete the installation.

9. After completing the TEM installation, install the Web tier and deploy the web applications.

Create a library of finish and finish groups

In Classification Admin application, create a class.

Ensure that you clear the **Abstract** option.

Create a finish

1. From My Teamcenter, choose **File→New→Item**.

2. Select **Finish** from the **New Item** dialog box.

   Click **Next**.

3. In the **Object Create Information** dialog box, enter the following information:

Finish ID	Specifies the ID of the finish.
Revision	Specifies the revision ID of the finish.
Name	Specifies the name of the finish.
Description	Describes the finish.

4. Click **Next**.

   Specify additional information about the finish in the following dialogs.

5. Click **Finish**.

Create a finish group

> Note:
>
> Copy the finishes that you want to add to the finish group before creating it.

1. From My Teamcenter, choose **File→New→Item**.

2. Select **Finish Group** from the **New Item** dialog box.

   Click **Next**.

3. In the **Object Create Information** dialog box, enter the following information:

   Finish Group ID Specifies the ID of the finish group.

   Revision Specifies the revision ID of the finish group.

   Name Specifies the name of the finish group.

   Description Describes the finish group.

   Finish Items Specifies the finishes to be associated with the finish group.

   To add finishes, ensure that you copy the finishes to the clipboard.

   a. Click the **Expand to modify** button ✏.

   b. Click the **Add objects from Clipboard** button +.

      Teamcenter adds the finish objects to the **Finish Items** list.

4. Click **Next**.

   Specify additional information about the work package in the following dialogs.

5. Click **Finish**.

Add finishes to a finish group

After creating a finish group, you can add finishes to the finish group as follows:

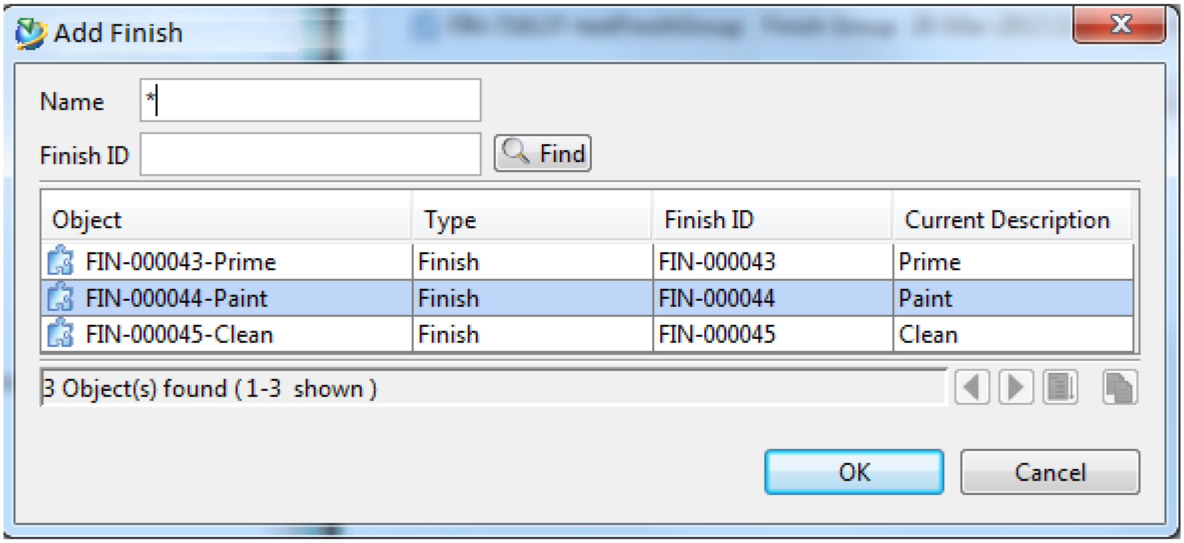
1. Right-click a finish group revision and choose **Check-In/Out→Check Out**

2. Click **Yes** in the **Check-Out** dialog box to check out the finish group.

3. Select the Summary view of the finish revision and choose the **Finishes** tab.

4. Click **Add Finish** button to open the **Add Finish** dialog.

5. Search for the finish in the **Add Finish** dialog box and click **Find**.



Name: *

Finish ID:

Object	Type	Finish ID	Current Description
FIN-000043-Prime	Finish	FIN-000043	Prime
FIN-000044-Paint	Finish	FIN-000044	Paint
FIN-000045-Clean	Finish	FIN-000045	Clean

3 Object(s) found (1-3 shown)

6. Select the finishes you want and click **OK**.

   The finishes are added in the **Finishes** section.

7. To reorder the sequence of finishes in the **Finish Sequence** section:

   a. Click the **Expand to modify** button ✏.

   b. Select the finish and move it up or down the sequence by using the **Move selected objects up** ▲ or **Move selected objects up** ▼ buttons.

8. Right-click the finish group revision and choose **Check-In/Out→Check In**.

9. Click the **Yes** button in the **Check-In** dialog box.

The finishes are now added to the finish group.

Apply a finish or a finish group to an item revision

1. Right-click an item revision and choose **Assign Finish**.

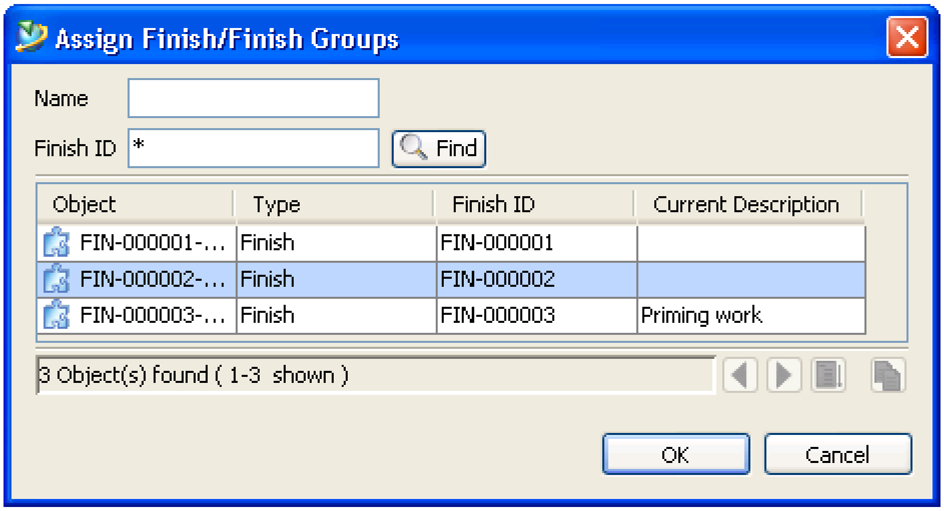
   If the value set in the **UsingAssignFinishDialog** preference is **TRUE**, you see the **Assign Finish/ Finish Groups** dialog box and if the value is **FALSE**, you see the **Classification Search** dialog box.

2. If you see the **Assign Finish/Finish Groups** dialog box, type the name of the finish or finish group in the **Name** box.

   OR

   Type the ID of the finish or finish group in the **Finish ID** box.

   Click **Find**.



Name:

Finish ID: *

Object	Type	Finish ID	Current Description
FIN-000001-…	Finish	FIN-000001	
FIN-000002-…	Finish	FIN-000002	
FIN-000003-…	Finish	FIN-000003	Priming work

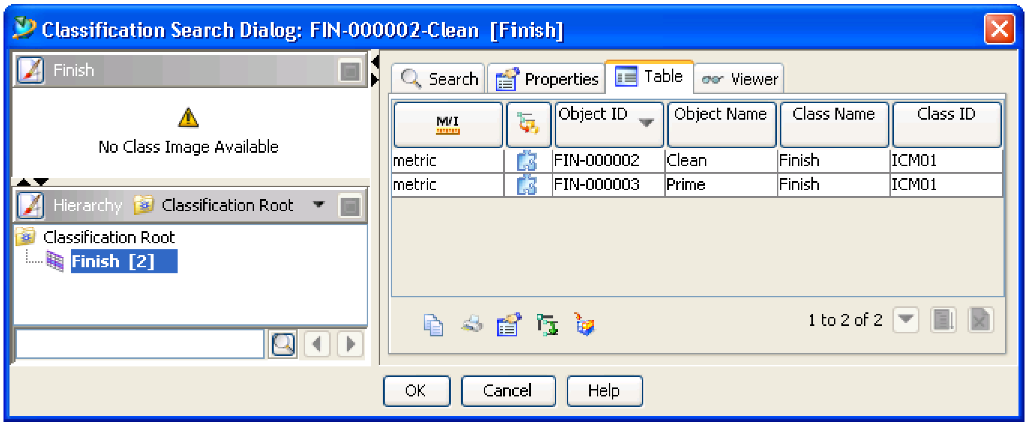
3 Object(s) found (1-3 shown)

3. Select the finishes you want and click **OK**.

   The selected finish and finish groups are added to the **Finishes** folder of the item revision.

4. If you see the **Classification Search** dialog box, type the search criteria in the search fields and click **Search**.

   The search results appear in the **Table** tab.



M/I		Object ID	Object Name	Class Name	Class ID
metric		FIN-000002	Clean	Finish	ICM01
metric		FIN-000003	Prime	Finish	ICM01

5. Select the finishes you want and click **OK**.

The selected finish and finish groups are added to the **Finishes** folder of the item revision.

Apply a finish to an item revision using the copy functionality

Drag the finish or finish group to the item revision

- Drag the finish or finish group to the **Finishes** folder of the item revision.

Copy and paste the finish or finish group to the item revision

- Copy and paste the finish or finish group to the **Finishes** folder of the item revision.

Use the copy and paste special functionality to add finish or finish group to the item revision

- Right-click the finish or finish group and choose **Copy**.
- Select the item revision and choose **Edit→Paste Special**.
- From the **Paste Special** dialog box, choose **Finishes**.

  The finish or finish group is added to the **Finishes** folder of the item revision.

Apply a finish or a finish group to an item occurrence

Ensure that trace links are enabled.

1. Open a product structure in Structure Manager and right-click the item occurrence and choose **Assign Finish to Occurrence**.

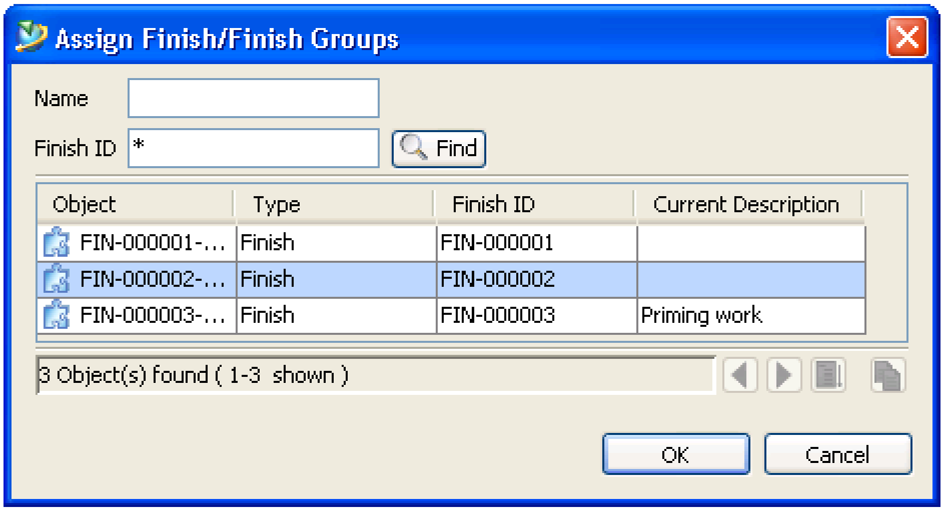
   If the value set in the **UsingAssignFinishDialog** preference is **TRUE**, you see the **Assign Finish/ Finish Groups** dialog box and if the value is **FALSE**, you see the **Classification Search** dialog box.

2. If you see the **Assign Finish/Finish Groups** dialog box, type the name of the finish or finish group in the **Name** box.

   OR

   Type the ID of the finish or finish group in the **Finish ID** box.

   Click **Find**.



Name:

Finish ID: *

Object	Type	Finish ID	Current Description
FIN-000001-...	Finish	FIN-000001	
FIN-000002-...	Finish	FIN-000002	
FIN-000003-...	Finish	FIN-000003	Priming work

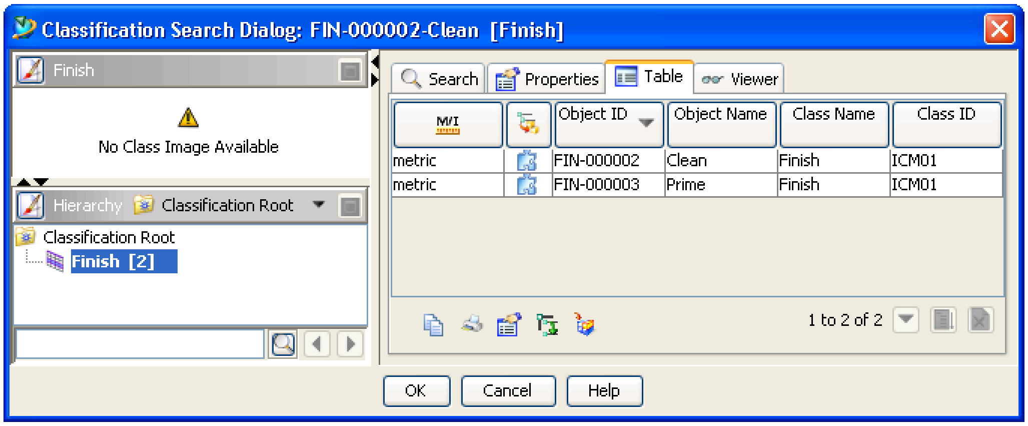
3 Object(s) found (1-3 shown)

3. Select the finishes you want and click **OK**.

   The selected finish and finish groups are associated to the selected occurrence and can be seen in the **Attachments** tab of the occurrence.

4. If you see the **Classification Search** dialog box, type the search criteria in the search fields and click **Search**.

   The search results appear in the **Table** tab.



M/I		Object ID	Object Name	Class Name	Class ID
metric		FIN-000002	Clean	Finish	ICM01
metric		FIN-000003	Prime	Finish	ICM01

1 to 2 of 2

5. Select the finishes you want and click **OK**.

   The selected finish and finish groups are associated to the selected occurrence and can be seen in the **Attachments** tab of the occurrence.

A finish is now applied to an item occurrence.

12. Managing configuration audits

What are configuration audits

Configuration audits helps validate if the functional and physical requirements of a product meet the requirements specified in the product configuration documentation.

Configuration audit helps to:

- Ensure that the product design provides the agreed-to performance.
- Validate the consistency between a product and its associated configuration documentation.
- Determine that adequate processes are in place to support production.

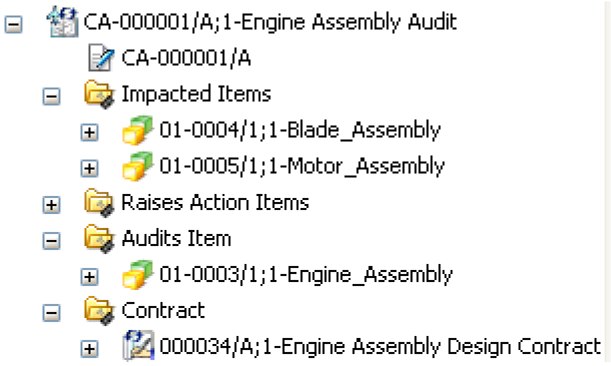
You can perform different types of configuration audits such as Functional Configuration Audit (FCA) and Physical Configuration Audits (PCA).

The findings of the configuration audit results in action items. The action items represent the request for actions raised during the audit.

The following is an example process of how configuration audits are performed:

- The compliance auditor performs a configuration audit on a product or deliverable and creates a configuration audit item in Teamcenter.

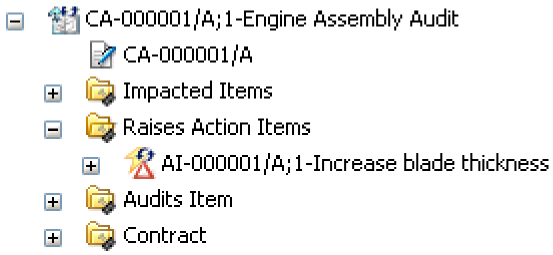
  The auditor adds the objects being audited to the **Audits Item** folder of the configuration audit and adds the objects affected by the audit to the **Impacted Items** folder of the configuration audit. Example: If an engine assembly is audited, it is added to the **Audits Item** folder and the subassemblies of the engine such as motor assembly and blade assembly are added to the **Impacted Items** folder.

  - CA-000001/A;1-Engine Assembly Audit
    - CA-000001/A
    - Impacted Items
      - 01-0004/1;1-Blade_Assembly
      - 01-0005/1;1-Motor_Assembly
    - Raises Action Items
    - Audits Item
      - 01-0003/1;1-Engine_Assembly
    - Contract
      - 000034/A;1-Engine Assembly Design Contract

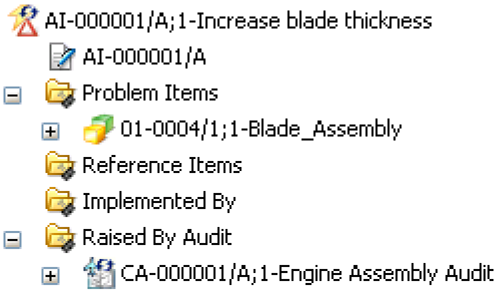
  Based on the findings of the audit, the auditor suggests action items.

- The change manager creates action items and associates the action item with the configuration audit.



```
⊟ CA-000001/A;1-Engine Assembly Audit
      CA-000001/A
   ⊞ Impacted Items
   ⊟ Raises Action Items
      ⊞ AI-000001/A;1-Increase blade thickness
   ⊞ Audits Item
   ⊞ Contract
```

- The change manager also associates the action item with the affected item. Example: The affected item, the blade assembly is added to the **Problem Items** folder of the action item.



```
AI-000001/A;1-Increase blade thickness
   AI-000001/A
⊟ Problem Items
   ⊞ 01-0004/1;1-Blade_Assembly
   Reference Items
   Implemented By
⊟ Raised By Audit
   ⊞ CA-000001/A;1-Engine Assembly Audit
```

- The change manager associates the action item to a change request and then sends the change request through a change management workflow process.

  If the action item is resolved using a formal change process, the items in the folders of the action item are propagated to the change request by using the Derive functionality.

Set up configuration audits

Set the following business object constants:

- **Cm0AuditsItemCreCondition**

  Specifies the condition to use for checking the status of the configuration audit before the creation of an **Audits Items** relation.

- **Cm0HasContractCreCondition**

  Specifies the condition to use for checking the status of the configuration audit before the creation of a **Has Contract** relation.

- **Cm0RaisesActionItemsCreCondition**

  Specifies the condition to use for checking if the closure status of the configuration audit is open and the **Audits Item** relationship is established before the creation of the **Raises Action Item** relation.

Create a configuration audit

1. From My Teamcenter, choose **File→New→Audit**.

2. Select **Configuration Audit** from the **New Item** dialog box.

   Click **Next**.

3. In the **New Audit** dialog box, enter the following information.

   Audit Number Specifies the ID for the configuration audit.

   Revision Specifies the revision ID for the configuration audit.

   Synopsis Specifies the name of the configuration audit.

   Description Describes the configuration audit.

   Audit Type Specifies the type of audit. You can choose from the following audit types:

   - **PCA**: Physical Configuration Audit
   - **FCA**: Functional Configuration Audit
   - **CDR**: Critical Design Review
   - **PDR**: Preliminary Design Review
   - **TRR**: Test Readiness Review

   Process Date Specifies the date on which the configuration audit is conducted.

   Comments Specifies any other information about the configuration audit.

   Work Package Type Specifies if the work package is executed internally or by a supplier.

   Work Package Vendor Specifies the name of the vendor executing the work package.

4. Click **Finish**.

After creating a configuration audit, add the object being audited to the **Audits Item** folder of the configuration audit item. You can add objects to the **Audits Item** folder using the copy-paste method.

You can also initiate a change management workflow on the configuration audit.

Create an action item

> Note:
>
> Ensure that the object being audited is added to the **Audits Item** folder of the configuration audit item. If you do not do this, the action items will not be added to the **Raises Action Items** folder of the configuration audit item.
>
> Use the copy-paste method to add the object being audited to the **Audits Item** folder.

1. Right-click a configuration audit revision and choose **Raise Action Item**.

2. Select **Action Item** from the **Add New Action Item** dialog box.

   Click **Next**.

3. In the **Add New Action Item** dialog box, add the following information.

Action Item Number	Specifies the ID for the action item.
Revision	Specifies the revision ID for the action item.
Synopsis	Specifies the name of the action item.
Description	Describes the action item.
Action Type	Specifies the type of action item. You can choose from the following action types: • **Audit**: Physical Configuration Audit • **CCR**: Configuration Change Request • **RFA**: Request for Action • **RFI**: Request for Information
Comments	Specifies any other information about the action item.
Impacts	Specifies the impacted item.
Office Primary Responsible	Specifies the person or office responsible for the action item.
Zones	Specifies the product zones affected by the action item.

4. Click **Finish**.

The action item is added to the **Raises Action Items** folder of the configuration audit.

Associate related objects with a configuration audit or an action item

To associate objects to a configuration audit, either copy or drag the object to the relevant folder of the configuration audit. You can add objects to the following folders:

- **Impacted Items**

  Add objects that are impacted by the audit.

- **Audits Item**

  Add objects that are being audited.

- **Contract**

  Add contract documents for the objects being audited.

To associate objects to an action item, either copy or drag the object to the relevant folder of the action item revision. You can add objects to the following folders:

- **Problem Items**

  Add part revisions that the action item is meant to address.

- **Reference Items**

  Add any Teamcenter object, including datasets, that reference related information.

- **Implemented By**

  Add change object revisions that are referenced by the action item.

- **Raised by Audit**

  Contains the configuration audit item.

13. Searching for Aerospace and Defense items

You can use the Teamcenter search functionality to search for Aerospace and Defense items.

In addition to Teamcenter search, you can use the following Aerospace and Defense search forms:

Search form	Description
Find ADSDesign	Returns design items according to the criteria you specify.
Find ADSDrawing	Returns drawing items according to the criteria you specify.
Find ADSPart	Returns part items according to the criteria you specify.
FindADSTechDocument	Returns technical document items according to the criteria you specify.
Find All Change Notice Revisions	Returns all change notice revisions. You do not specify criteria for this query.
Find Organization	Returns the list of organizations. You do not specify criteria for this query.

The **GroupBasedProjects** search form in the standard Teamcenter installation can be used with programs. This query returns the list of programs that a specified group is working on.

Note:

Standard change item searches do not return the Aerospace and Defense change items. You must configure search forms to work with Aerospace and Defense change items.

A. Aerospace and Defense Training Program

Overview of the Aerospace and Defense training program

The **Training** program is a default program that is created on your system when you choose to install the **Aerospace and Defense Training Program** template along with the **Aerospace and Defense Foundation** template. The training program helps you learn how to create and organize necessary data in the context of a program.

> Note:
>
> The **Aerospace and Defense Training Program** template is dependent on the **Aerospace and Defense Foundation** template.

Install the training program

1. Access the **Teamcenter Environment Manager** (TEM) installer.

2. In the **Solutions** pane, select **Aerospace and Defense**.

3. In the **Select Features** pane, select **Vendor Management**, **Aerospace and Defense Foundation**, and **Aerospace and Defense Training Program**.

   > Note:
   >
   > The **Aerospace and Defense Training Program** template is dependent on the **Aerospace and Defense Foundation** template. You must select both the features to install the training program.

4. If this is a new installation, enter the directory in which you want to install Teamcenter in the **Installation Directory** box. Specify a directory that does not exist. Teamcenter Environment Manager (TEM) creates the directory you specify.

   On successful installation, two training programs are created in the database.

Program ID	Description
Training	Provides an example of how to create and organize data for a document-centric program.
Training-Part Centric	Provides an example of how to create and organize data for a part-centric program.

Training program reference

Training program reference

Aerospace and Defense Training Program provides a preconfigured program with rules and lists of values that are saved in the **adstrainingprogram_template.xml** file.

The following components are included in the template:

- Conditions
- Naming rules
- Revision naming rules
- Lists of values

Programs

Programs provide a mechanism for organizing data and implementing access control based on program membership. Data assigned to programs can be searched for and viewed in the context of the program and can be distributed across multiple sites.

To work with programs, you must perform additional configurations like assigning users to programs, setting a program as the default program of the user, setting the program security and son on.

The following concepts apply to programs:

- Only privileged team members can assign data to programs.
- Data can be assigned to or removed from programs manually or when the data item is created, and items can be assigned to more than one program.
- Propagation rules define the associated data that is implicitly assigned to a program when a primary item is assigned to the program.
- All items in a complete product structure can be assigned to a program using the **update_project_bom** utility.

The **Training** program is provided by the **Aerospace and Defense Training Program** template.

Training

Description	This is the default program that is provided by the **Aerospace and Defense Training Program** template. The program is document-centric with program-

	level security enabled. The program is preconfigured with conditions, naming rules, revision naming rules, and lists of values.
Program type	Document-centric
Program-level security	Enabled
Notes	None

Training-Part Centric

Description	This is the default program that is provided by the **Aerospace and Defense Training Program** template. The program is part-centric with program-level security enabled. The program is preconfigured with conditions, naming rules, revision naming rules, and lists of values.
Program type	Part-centric
Program-level security	Enabled
Notes	None

Conditions

Conditions

Conditions are conditional statements that are run with rules; they resolve to true or false. Conditions can be used to evaluate objects or user sessions to deliver only certain results.

ADSTrngCondition

Description	This condition is configured for the **Training** and **Training-Part Centric** programs.
Expression	**o.project_name = "Training”** **o** is the parameter name and the parameter type is **UserSession**.
Notes	The condition evaluates to **true** when the active program for the user session is the **Training** or **Training-Part Centric** program; otherwise, the condition evaluates to **false**.
Where used	This condition is used in naming rules, revision naming rules, and lists of values.

AnDCMTrainingProgramCondition

Description	This condition is configured to make the ADS Training Program data model applicable to change objects.
Expression	**o.project_name = "Training”** **o** is the parameter name and the parameter type is **UserSession**.
Notes	The condition evaluates to **true** when the active program for the user session is the **Training** program; otherwise, the condition evaluates to **false**. The condition is provided by the **Aerospace and Defense Change Management** template.
Where used	This condition is used in naming rules, revision naming rules, and lists of values.

Naming rules

Naming rules

Naming rules provide a way for applying custom naming conventions to items, item revisions, identifiers, datasets, forms, projects, and work contexts. In addition, naming rules can be used to define patterns for automatically generating IDs when creating objects. A naming rule consists of multiple naming rule patterns and a counter.

TrngNamingRule is a naming rule provided by the **Aerospace and Defense Training Program** template and is attached to the **Training** and **Training-Part Centric** programs.

TrngNamingRule

Description	This naming rule is attached to the **item_id** property of all ADS business objects. The ADSTrngCondition condition is evaluated to determine whether the naming rule is applied. The naming rule is attached to the business object property when the condition evaluates to **true**.
Business objects	The naming rule is attached to the following business objects: • **ADSTechDocument** • **ADSPart** • **ADSDesign** • **ADSDrawing**

Business object property

item_id

Pattern

“01–”NNNN

“02–”NNNN

“01–”NNNN”-”NNN

Counter

Enabled.

Condition

ADSTrngCondition

Notes

For the **“01–”NNNN** pattern, the initial value is **01–0000** and the maximum value is **01–9999**.

For the **“02–”NNNN** pattern, the initial value is **02–0000** and the maximum value is **02–9999**.

For the **“01–”NNNN”-”NNN** pattern, the initial value is **01–0000–000** and the maximum value is **01–9999–999**.

Adc0TrngNamingRule

Description

This naming rule is attached to the **item_id** property of all change objects. The AnDCMTrainingProgramCondition condition is evaluated to determine whether the naming rule is applied. The naming rule is attached to the business object property when the condition evaluates to **true**.

Business objects

The naming rule is attached to the following business objects:

- **Adc0ChangeNotice**
- **Adc0ChangeRqst**
- **Adc0DevRqst**
- **ProblemReport**

Business object property

item_id

Pattern

“01–”NNNN

“02–”NNNN

Counter

Enabled.

Condition	**AnDCMTrainingProgramCondition**
Notes	For the **“01–”NNNN** pattern, the initial value is **01–0000** and the maximum value is **01–9999**. For the **“02–”NNNN** pattern, the initial value is **02–0000** and the maximum value is **02–9999**. This rule is included in the **Aerospace and Defense Change Management** template.

Revision naming rules

Revision naming rules

Revision naming rules make it easy to maintain the revision scheme for business objects. Each time an object is revised, the revision ID increments to indicate that the object has been revised and the new revision supersedes the previous revision.

TrngRevNamingRule is a revision naming rule provided by the **Aerospace and Defense Training Program** template and is attached to the **Training** and **Training-Part Centric** programs.

TrngRevNamingRule

Description	This revision naming rule is attached to the **item_revision_id** property of all ADS business objects. The ADSTrngCondition condition is evaluated to determine whether the revision naming rule is applied. The revision naming rule is attached to the business object property when the condition evaluates to **true**.
Business objects	The revision naming rule is attached to the following business objects: • **ADSTechDocument** • **ADSPart** • **ADSDesign** • **ADSDrawing**
Business object property	**item_revision_id**
Exclude skip letters	True
Condition	**ADSTrngCondition**
Initial revision details	The initial revision type is numeric and the starting value is **1**.

Secondary revision details	The secondary revision type is alphabetic and the starting value is **-**.
Supplemental revision details	The supplemental revision format is **NextRevLetterFixedTwoDigitsZeroFill**.
Notes	None

Adc0TrngRevNamingRule

Description	This revision naming rule is attached to the **item_revision_id** property of all the ADS business objects. The AnDCMTrainingProgramCondition condition is evaluated to determine whether the revision naming rule is applied. The revision naming rule is attached to the business object property when the condition evaluates to **true**.
Business objects	The revision naming rule is attached to the following business objects: • **Adc0ChangeNoticeRevision** • **Adc0ChangeRqstRevision** • **Adc0DevRqstRevision** • **ProblemReportRevision**
Business object property	**item_revision_id**
Exclude skip letters	**True**
Condition	**AnDCMTrainingProgramCondition**
Initial revision details	The initial revision type is numeric and the starting value is **1**.
Secondary revision details	The secondary revision type is alphabetic and the starting value is **-**.
Supplemental revision details	The supplemental revision format is **NextRevLetterFixedTwoDigitsZeroFill**.
Notes	This rule is included in the **Aerospace and Defense Change Management** template.

Lists of values

Lists of values

Lists of values (LOVs) ensure consistent data entry in Teamcenter. The list entries are created either by referencing existing data or by entering custom site data. After it is created, the LOV is implemented throughout the interface by attaching the LOV to one or more properties. Creating LOVs and implementing them throughout Teamcenter can greatly improve productivity at your site and help prevent incorrect user entries.

This section describes the LOVs that are provided by the **Aerospace and Defense Training Program** template and attached to the **Training** and **Training-Part Centric** programs.

Adc0Change Class

Description	Designates the government's or customer's classification to a change. For example, **I** or **II**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeNoticeRevision.Adc0change_class** **Adc0ChangeRqstRevision.Adc0change_class** **Adc0DevRqstRevision.Adc0change_class**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**I** **II**
Valid values	Accepts string as a value. It must be a valid classification.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Change Category

Description	Designates the specific category of change within the change classification.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeRqstRevision.Adc0change_category** **Adc0ChangeNoticeRevision.Adc0category**

	Adc0DevRqstRevision.Adc0change_category
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**1** **2** **A** **B** **C** **D** **E** **M**
Valid values	Accepts string as a value.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0CN Types

Description	Specifies acronyms for the types of change documentation used on a program for a change notice.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeNotice.Adc0change_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**ADCN** **AMR** **DCN** **DCR** **DRN** **ICN** **PCD** **PMD** **REDLINE** **RR** **VAR**

Valid values	Accepts string as a value. It must be a valid acronym that specifies the type of change documentation.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0CR Types

Description	Specifies the change request types. For example, development change or engineering change process.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeRqst.Adc0change_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**CA** **CRBD** **DEV** **ECP** **IPCD** **RFD** **RFW**
Valid values	Accepts string as a value. It must be a valid change request type.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0DR Types

Description	Specifies acronyms for the types of change documentation used on a program for a deviation request, for example, **RFD** or **RFW**.
Type	**ListOfValuesString**
Usage	Exhaustive.

Business object property	**Cm0DevRqst.Cm0change_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**RFD** Request for Deviation **RFW** Request for Waiver
Valid values	Accepts string as a value. It must be a valid acronym that specifies the type of change documentation.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Production Systems

Description	Displays the list of systems or subsystems for an item. A product system denotes a group of parts that perform a common function. For example, **Electrical**, **Environmental Control**, and **Weapons**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0DevRqstRevision.Adc0product_systems** **Adc0ChangeRqstRevision.Adc0product_systems** **Adc0ChangeNoticeRevision.Adc0product_systems**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**Electrical** **Environmental Control** **Hydraulic** **Fuel** **Weapons**
Valid values	Accepts string as a value. It must be a valid system or subsystem.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Quotation Type

Description	Specifies the quotation type for the change impact, for example, **Supplier ROM**, **Buyer Estimate**, or **Supplier Firm**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeRqstRevision.Adc0quotation_type** **Adc0DevRqstRevision.Adc0quotation_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**NTE** **ROM**
Valid values	Accepts string as a value. It must be a valid quotation type.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Task Category

Description	Specifies the various task categories, for example, **Agenda Event**, **Planning**, and **QA**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0Task.Adc0task_category**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**Agenda Event** **Engineering** **ILS** **Major Subcontr** **Manufacturing** **Non-Std Event** **Part** **Planning**

	QA
	Subcontract
Valid values	Accepts string as a value. It must be a valid task category.
Notes	This LOV is included in the **adschangemanagement** template.

ADS Trng Categories

Description	Specifies the valid values for the ADS document category. This is a cascading list.	
Type	**ListOfValuesString**	
Usage	Exhaustive.	
Business object property	**ADSTechDocument.adscategory** **ADSPart.adssource_tdoc_category** **ADSDesign.adssource_tdoc_category** **ADSDrawing.adssource_tdoc_category**	
Business object condition	**ADSTrngCondition**	
Default values	**ALOO**	
	DL	Document list
	DOC	Document
	DWG	Drawing
	MFG	
	PL	Parts list
	SDL	
	SDRL	Supplier definition requirements list
	SOW	Statement of work
	SPEC	Specification
Valid values	Any valid document category.	
Notes	Each value contains a sub-LOV.	

ADS Trng DOC Categories

Description	Sub-LOV that specifies the list of values if the ADS category is document.
Type	**ListOfValuesString**
Usage	Exhaustive.
Default values	**Dwg Design Data** **Interface Control Document** **Text Report** **Weapon System Specification**
Valid values	Any valid drawing.
Notes	None.

ADS Trng DWG Categories

Description	Sub-LOV that specifies the list of values if the ADS category is drawing.
Type	**ListOfValuesString**
Usage	Exhaustive.
Default values	**Assembly Drawing** **Design Drawing** **Detail Drawing** **Tube Drawing** **Zone Drawing**
Valid values	Any valid drawing.
Notes	None.

ADS Trng PL Categories

Description	Sub-LOV that specifies the list of values if the ADS category is part.
Type	**ListOfValuesString**
Usage	Exhaustive.
Default values	**Assembly Drawing** **Detail Drawing** **Tube Drawing** **Zone Drawing**
Valid values	Any valid drawing.
Notes	None.

ADS Trng SDRL Categories

Description	Sub-LOV that specifies the list of values if the ADS category is SDRL.
Type	**ListOfValuesString**
Usage	Exhaustive.
Default values	**Requirement List**
Valid values	Any valid document type.
Notes	None.

ADS Trng Design Categories

Description	Specifies valid values for the ADS design category.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**ADSDesign.adsdesign_category**
Business object condition	**ADSTrngCondition**
Default values	**Program** **Design** **Domestic** **Supplier** **GFE**
Valid values	Any valid design category.
Notes	None.

ADS Trng Dist Code

Description	Specifies valid values for the distribution code.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**ADSTechDocument.adsdistribution_code** **ADSPart.adsdistribution_code** **ADSDesign.adsdistribution_code** **ADSDrawing.adsdistribution_code**

Business object condition	**ADSTrngCondition**
Default values	**A01**
Valid values	Any valid distribution code.
Notes	None.

ADS Trng Part Categories

Description	Specifies valid values for the ADS part category.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**ADSPart.adspart_category**
Business object condition	**ADSTrngCondition**
Default values	**Program** **Design** **Domestic** **Supplier** **GFE**
Valid values	Any valid part category.
Notes	None.

B. Aerospace and Defense data model

Aerospace and Defense data model description

The following structure is used to describe the objects:

Object type Describes the data model of the object that represents the parent class and the subclass of the object.

Object type number Describes the attributes of the object that provides it a unique identifier.

Object type master Describes the class model of the object that represents the attributes of the object.

Preconditions and postactions Describes the preconditions and postactions that should be defined during the creation of the object/component.

Template information Describes the template in which the functionality of the object is available.

List of values Describes the program-independent list of values for the master form and the revision master form.

Object type revision Describes the item revision corresponding to an object.

Preconditions and postactions

Extension point Describes the application extensions that allow for the configuration of applications using a decision table. Application extensions can be used to configure business logic on the server, a Teamcenter rich client application (such as My Teamcenter), or any application. You can use application extensions for anything that calls an input and output, from user interface changes, such as icons and colors, to actions.

Operation name Describes the operation to be placed on the business object.

Extension rule name Determines when an application extension point is used and defines inputs and outputs. When the input is matched, the rule engine returns the output to the application that called the extension point.

Parameters Describes the parameters to be passed to the extension.

Description Describes the application extension point.

ADSTechDocument

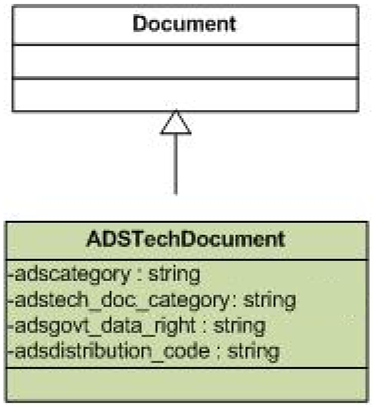
ADSTechDocument

ADSTechDocument is a subtype of the **Document** item type and represents the technical document used in document-centric programs.

ADSTechDocument is a primary business object that represents a variety of printed or digital entities that contain technical information about some aspect of the product or a component of the product. These may include documents such as technical reports, drawings, procurement specifications, parts lists, interface control documents, and schematics.

Technical documents serve as the source documents for creating parts and assemblies. All parts must be associated with a technical document using an appropriate relationship.

The following figure illustrates the **ADSTechDocument** data model.


Document
ADSTechDocument
-adscategory : string
-adstech_doc_category: string
-adsgovt_data_right : string
-adsdistribution_code : string

ADSTechDocument attributes

ADSTechDocument items are uniquely identified in the system by an **item_id** property. The **item_id** property denotes the technical document number.

The **ADSTechDocument** item is provided by the **adsfoundation** template.

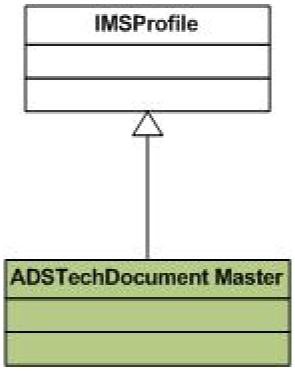
The following table describes the attributes of the **ADSTechDocument** class.

Attribute name	Storage type	Description
adscategory	**String [32]**	Classifies document items, for example, **DOC**, **DWG**, **PL**, and **SPEC**.

Attribute name	Storage type	Description
		The attribute values must be configured based on the individual program. There are no default values.
adstech_doc_category	**String [32]**	Designates a category of similar technical documents, for example, **Dwg Design Data**, **Weapon System Specification**, **Assembly Drawing**, **Design Drawing**, **Detail Drawing**, and **Requirements List**. The attribute values must be configured based on the individual program. The valid values for this attribute are dependent on the value specified for the **category** attribute.
adsgovt_data_right	**String [32]**	Specifies the nature of the government technical data rights-in-data for a document or software. The attribute values must be configured based on the individual program. There are no default values.
adsdistribution_code	**String [32]**	Specifies the distribution statement that is affixed to the document or viewable file to indicate the authorized circulation or dissemination of the information contained within the item.

ADSTechDocument Master

ADSTechDocument Master is the storage class of the item master form corresponding to the **ADSTechDocument** item. The following figure illustrates the schema definition and attributes of the **ADSTechDocument Master** class.



Preconditions and postactions on ADSTechDocument

The following table describes the preconditions that are defined on the **ADSTechDocument** business object.

Extension rule name	Operation name	Parameters
authorizationCheckOnSaveAs Checks the user permission while saving an **ADSTechDocument**.	**ITEM_create_from_rev**	**ADSTechDocAuthority**
checkAdsAttachment Checks whether or not the selected source document is associated with other items during deletion.	**IMAN_delete**	**ADSTechDocAuthority**
validateOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value is set to true and the current user group belongs to an organization, the workspace object is created.	**ITEM_create** **ITEM_create_from_rev**	
validateImport Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. Data is imported when the value of the constants is true and the data belongs to the same organization at the both the sites.	**TIE_deserialize**	
authorizationCheckOnRevise Checks the user permission during the revision of **ADSTechDocument Revision**.	**ITEM_copy_rev**	

The following table describes the postactions that are defined on the **ADSTechDocument** business object.

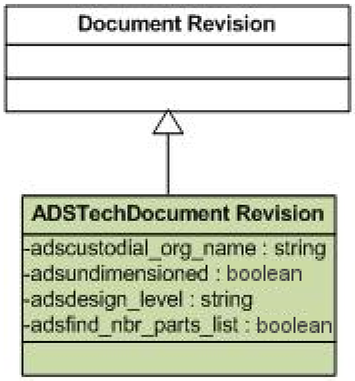
Extension rule name	Operation name	Parameters
autoAssignToProject Assigns the object to the current project.	**ITEM_create** **IMAN_import** **ITEM_create_from_rev**	
setOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**ITEM_create** **ITEM_create_from_rev**	

ADSTechDocument Revision

ADSTechDocument Revision is a subtype of the **Document Revision** business object and represents revisions corresponding to a technical document.

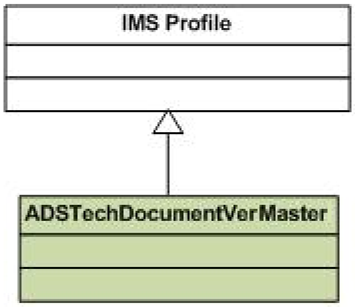
ADSTechDocument Revision is a primary business object.

The following figure illustrates the **ADSTechDocument Revision** business object data model.


Document Revision
ADSTechDocument Revision
-adscustodial_org_name : string
-adsundimensioned : boolean
-adsdesign_level : string
-adsfind_nbr_parts_list : boolean

ADSTechDocumentVerMaster

ADSTechDocumentVerMaster is the storage class of the item revision master form corresponding to the **ADSTechDocumentRevision** business object. The following figure illustrates the schema definition and attributes of the **ADSTechDocumentVerMaster** class.



ADSPart

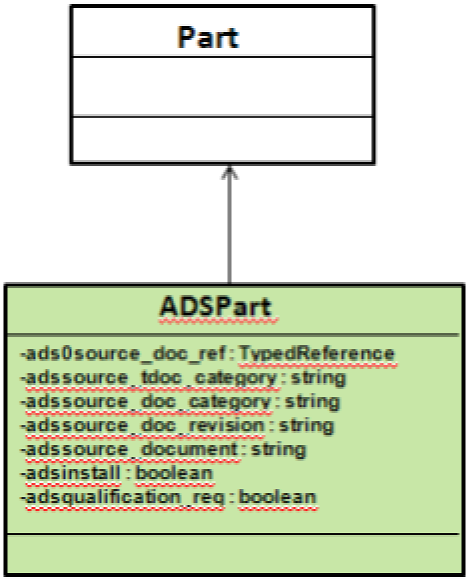
ADSPart

ADSPart is a subtype of the **Part** item type, which represents a component of a product. It is a primary business object.

> Note:
>
> The terms part and component are used interchangeably.

In document-centric programs, **ADSPart** objects are created on a source technical document and are related to the technical document through an **ADS_Lists_Parts** relationship.

The following figure illustrates the **ADSPart** data model.


Part
ADSPart
-ads0source_doc_ref : TypedReference
-adssource_tdoc_category : string
-adssource_doc_category : string
-adssource_doc_revision : string
-adssource_document : string
-adsinstall : boolean
-adsqualification_req : boolean

ADSPart attributes

ADSPart items are uniquely identified in the system by an **item_id** property. The **item_id** property denotes the ADS part number.

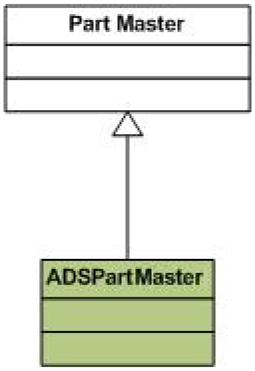
The **ADSPart** item is provided by the **adsfoundation** template.

The following table describes the attributes of the **ADSPart** class.

Attribute name	Storage type	Description
ads0source_doc_ref	**TypedReference**	Specifies the technical document.
adssource_document	**String [128]**	The unique identifier for the document the part is created on. **ADS source document** is a text field that can contain up to 80 characters.
adssource_doc_revision	**String [32]**	Specifies the revision of the ADS source document.
adssource_doc_category	**String [32]**	A user-defined attribute for classifying document items.
adssource_tdoc_category	**String [32]**	Designates a category of similar technical documents, for example, assembly drawing, wire list, or material specification.
adssource_doc_org	**String [32]**	Specifies organization name of source document.
adsqualification_req	**Boolean**	Indicates whether or not a part requires qualification testing before it can be installed in the product.
adsinstallation	**Boolean**	Indicates whether the assembly object is an installation.

ADSPartMaster

ADSPartMaster is the storage class of the item master form corresponding to the **ADSPart** item. The following figure illustrates the schema definition and attributes of the **ADSPartMaster** class.



Preconditions and postactions on ADSPart

The following table describes preconditions for the **ADSPart** object.

Extension rule name	Operation name	Parameters
authorizationCheckOnSaveAs Checks the user permission while saving an ADS part.	**ITEM_create_from_rev**	
DCPCheckMandatoryValuesSaveas Checks if the mandatory values are entered for a document-centric program.	**ITEM_create_from_rev**	**ADSComponentAuthority**
validateOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value is set to true and the current user group belongs to an organization, the workspace object is created.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
validateImport Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business	**TIE_deserialize**	

Extension rule name	Operation name	Parameters
constant. Data is imported when the value of the constants is true and the data belongs to the same organization at the both the sites.		
checkTechDocPrivilege Checks if the ADS part has access to the technical document.	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
authorizationCheckOnRevise Checks the user permission during the revision of an ADS part.	**ITEM_copy_rev**	
checkLatest_Released	**ITEM_copy_rev_to_existing** **ITEM_create_rev**	

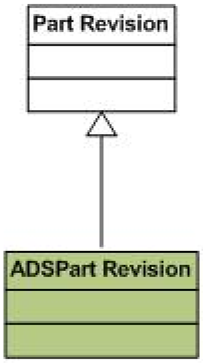
The following table describes postactions for the **ADSPart** object.

Extension rule name	Operation name	Parameters
cmpSrcDocRelationOnSaveAs Creates a relation between ADS part and the source document when the part is saved.	**ITEM_create_from_rev**	**ADSComponentAuthority**
cmpSrcDocRelationOnCreate Creates a relation between ADS part and the source document while creation of a part.	**ITEM_create**	**ADSComponentAuthority**
autoAssignToProject Assigns the object to the current project.	**IMAN_import** **ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing**	

Extension rule name	Operation name	Parameters
	ITEM_create_rev	
setOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
cmpSrcDocRelationOnRevise Creates a relation between ADS part and the source document when the part is revised.	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
Ads0AssociateChangeNotice	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	

ADSPart Revision

ADSPart Revision is a subtype of the **Part Revision** business object and represents revisions corresponding to an **ADSPart**. The following figure illustrates the **ADSPart Revision** business object data model.



ADSPartVerMaster

ADSPartVerMaster is the storage class of the item revision master form corresponding to the **ADSPartVerMaster** business object.

ADSDesign

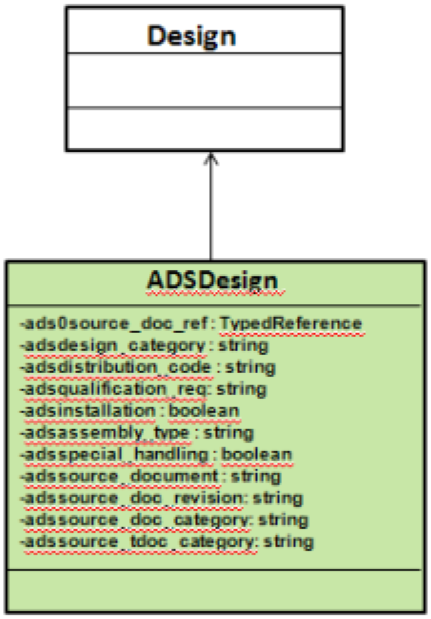
ADSDesign

ADSDesign is a subtype of the **Design** item type that represents the geometric data of a component or assembly.

ADSDesign is a primary business object.

In document-centric programs, **ADSDesign** are created on a source technical document and are related to the technical document through an **ADS_Lists_Parts** relationship.

The following figure illustrates the **ADSDesign** data model.


Design
ADSDesign
-ads0source_doc_ref : TypedReference
-adsdesign_category : string
-adsdistribution_code : string
-adsqualification_req: string
-adsinstallation : boolean
-adsassembly_type : string
-adsspecial_handling : boolean
-adssource_document : string
-adssource_doc_revision: string
-adssource_doc_category: string
-adssource_tdoc_category: string

Preconditions and postactions on ADSDesign

The following table describes preconditions for the **ADSDesign** object.

Extension rule name	Operation name	Parameters
DCPCheckMandatoryValuesSaveas Checks if the mandatory values are entered for a document-centric program.	**ITEM_create_from_rev**	**ADSComponentAuthority**
authorizationCheckOnSaveAs Checks the user permission while saving an ADSDesign.	**ITEM_create_from_rev**	
validateOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. If the value is set to true and the current user group belongs to an organization, the workspace object is created.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
validateImport Checks the value of the **TcSetOwningOrganization**	**TIE_deserialize**	

Extension rule name	Operation name	Parameters
global constant and the **AutoAssignOwningOrg** business constant. Data is imported when the value of the constants is true and the data belongs to the same organization at the both the sites.		
authorizationCheckOnRevise Checks the user permission during the revision of an ADSDesign.	**ITEM_copy_rev**	
checkTechDocPrivilege Checks if ADSDesign has access to the technical document.	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
checkLatest_Released	**ITEM_copy_rev_to_existing** **ITEM_create_rev**	

The following table describes postactions for the **ADSDesign** object.

Extension rule name	Operation name	Parameters
cmpSrcDocRelationOnSaveAs Creates a relation between ADSDesign and the source document when the design is saved.	**ITEM_create_from_rev**	
cmpSrcDocRelationOnCreate Creates a relation between ADSDesign and the source document when the design is created.	**ITEM_create**	
autoAssignToProject Assigns the object to the current project.	**IMAN_import** **ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing**	

Extension rule name	Operation name	Parameters
	ITEM_create_rev	
setOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
cmpSrcDocRelationOnRevise Creates a relation between ADSDesign and the source document when the design is revised.	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
Ads0AssociateChangeNotice	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	

ADSDesign attributes

ADSDesign objects are uniquely identified in the system by an **item_id** property.

The **item_id** property denotes the ADS design number.

The **ADSDesign** item is provided by the **adsfoundation** template.

The following table describes the attributes of the **ADSDesign** object.

Attribute name	Storage type	Description
ads0source_doc_ref	**TypedReference**	Specifies the technical document.
adsdesign_category	**String [32]**	Designates the category of the design for the program.
adsassembly_type	**String [32]**	Identifies the type of assembly for the

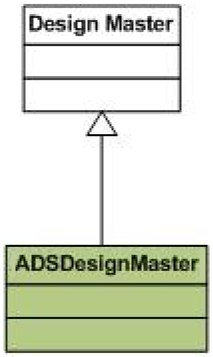
Attribute name	Storage type	Description
		part. Valid values are **Separable** and **Inseparable** An inseparable assembly part cannot be disassembled without causing physical harm to at least one of the constituent parts in the assembly.
adsdistribution_code	**String [32]**	Specifies the distribution statement that is affixed to the document or viewable file to indicate the authorized circulation or dissemination of the information contained within the item.
adsspecial_handling	**Boolean**	Indicates whether or not a part requires special handling.
adsqualification_req	**Boolean**	Indicates whether or not a part requires qualification testing before it can be installed in the product.
adsinstallation	**Boolean**	Indicates whether the assembly object is an installation.
adssource_doc_revision	**String [32]**	Specifies the revision of the ADS source document.
adssource_doc_category	**String [32]**	A user-defined attribute for classifying document items.
adssource_tdoc_category	**String [32]**	Designates a category of similar technical documents, for example, assembly drawing, wire list, or material specification.
adssource_doc_org	**String [32]**	Specifies the organization name of source document.
adssource_document	**String [128]**	The unique identifier of the document on

Attribute name	Storage type	Description
		which the part is created.

ADSDesignMaster

ADSDesignMaster is the storage class of the item master form corresponding to the **ADSDesign** item.

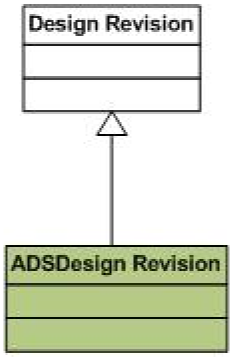
The following figure illustrates the schema definition and attributes of the **ADSDesignMaster** class.



ADSDesign Revision

ADSDesign Revision is a subtype of **Design Revision** business object and represents the revisions corresponding to **ADSDesign**.

The following figure illustrates the **ADSDesign Revision** data model.



Users must manually associate **ADSDesign Revision** with the corresponding **ADSPart Revision** using the **TC_Is_Represented_By** relation type.

ADSDesignVerMaster

ADSDesignVerMaster is the storage class of the item revision master form corresponding to the **ADSDesign Revision** business object.

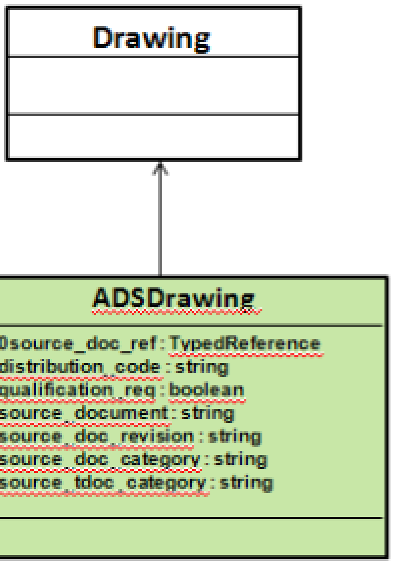
ADSDrawing

ADSDrawing

ADSDrawing is a subtype of the **Drawing** item type. In document-centric programs, **ADSDrawing** are created in the context of a source technical document.

These are referred to as *assembly drawings*. There can be one or more drawings for a given technical document, and a single drawing can detail out one or more assemblies or parts created with that technical document as their source document. The assemblies created on a given source document that have minor differences in their product structure configuration are detailed out on a single drawing.

The following figure illustrates the **ADSDrawing** data model.


Drawing
ADSDrawing
-ads0source_doc_ref : TypedReference
-adsdistribution_code : string
-adsqualification_req : boolean
-adssource_document : string
-adssource_doc_revision : string
-adssource_doc_category : string
-adssource_tdoc_category : string

Preconditions and postactions

The following table describes preconditions for the **ADSDrawing** object.

Extension rule name	Operation name	Parameters
DCPCheckMandatoryValuesSaveas	ITEM_create_from_rev	ADSDrawingAuthority

Extension rule name	Operation name	Parameters
Checks if the mandatory values are entered for a document-centric program.		
validateOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. If the value is set to true and the current user group belongs to an organization, the workspace object is created.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
authorizationCheckOnSaveAs Checks the user permission while saving an **ADSDrawing**.	**ITEM_create_from_rev**	
validateImport Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. Data is imported when the value of the constants is true and the data belongs to the same organization at the both the sites.	**TIE_deserialize**	
authorizationCheckOnRevise Checks the user permission during the revision of an **ADSDrawing**.	**ITEM_copy_rev**	
checkTechDocPrivilege Checks if ADS drawing has access to the technical document.	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
checkLatest_Released	**ITEM_copy_rev_to_existing** **ITEM_create_rev**	

The following table describes postactions for the **ADSDrawing** object.

Extension rule name	Operation name	Parameters
cmpSrcDocRelationOnSaveAs Creates a relation between ADS drawing and the source document when the drawing is saved.	**ITEM_create_from_rev**	**ADSDrawingAuthority**
cmpSrcDocRelationOnCreate Creates a relation between ADS drawing and the source document when the drawing is created.	**ITEM_create**	**ADSDrawingAuthority**
autoAssignToProject	**ITEM_import** **ITEM_create**	

Extension rule name	Operation name	Parameters
Assigns the object to the current project.	**ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
setOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
Ads0AssociateChangeNotice	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
cmpSrcDocRelationOnRevise Creates a relation between ADS drawing and the source document when the drawing is revised.	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	

ADSDrawing attributes

ADSDrawing objects are uniquely identified in the system by an **item_id** property. The **item_id** property denotes the ADS drawing number.

The **ADSDrawing** business object is provided by the **adsfoundation** template.

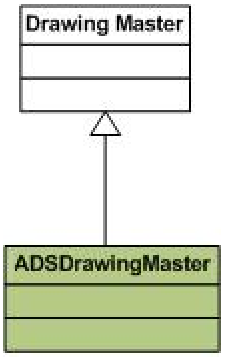
The following table describes the attributes of the **ADSDrawing** object.

Attribute name	Storage type	Description
ads0source_doc_ref	**TypedReference**	Specifies the technical document.
adsdistribution_code	**String [32]**	Specifies the distribution statement that is affixed to the document or viewable file to indicate the authorized circulation or dissemination of the

Attribute name	Storage type	Description
		information contained within the item.
adsqualification_req	**Boolean**	Indicates whether or not a part requires qualification testing before it can be installed in the product.
adssource_document	**String [128]**	The unique identifier for the document on which the part is created.
adssource_doc_revision	**String [32]**	Revision of the source document.
adssource_doc_category	**String [32]**	User-defined attribute used to classify **Document** items.
adssource_tdoc_category	**String [32]**	Designates a category of similar technical documents, for example, **assembly drawing**, **wire list**, or **material specification**.
adssource_doc_org	**Typed reference [32]**	Specifies organization name of the source document.

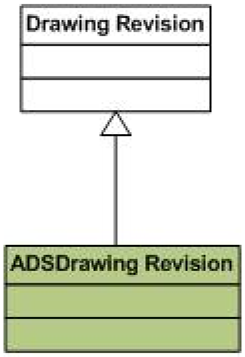
ADSDrawingMaster

ADSDrawingMaster is the storage class of the item master form corresponding to the **ADSDrawing** item. The following figure illustrates the schema definition and attributes of the **ADSDrawingMaster** class.



ADSDrawing Revision

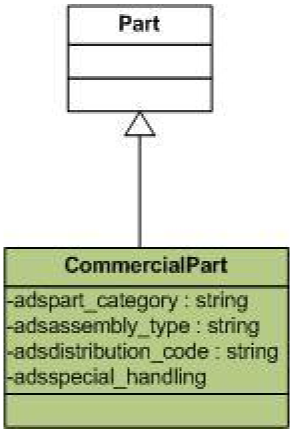
ADSDrawing Revision is a subtype of the **Drawing Revision** business object and represents revisions corresponding to an ADS drawing. The following figure illustrates the **ADSDrawing Revision** business object data model.



CommercialPart

CommercialPart

CommercialPart is a subtype of the **Part** item type. It represents the common-use parts that have been identified as standard design by a company, an industry, or the military. Standard parts are associated to programs with the **TC_Program_Preferred_Parts** relation. The following figure illustrates the **CommercialPart** data model.


Part
CommercialPart
-adspart_category : string
-adsassembly_type : string
-adsdistribution_code : string
-adsspecial_handling

CommercialPart attributes

CommercialPart objects are uniquely identified in the system by the **item_id**, property.

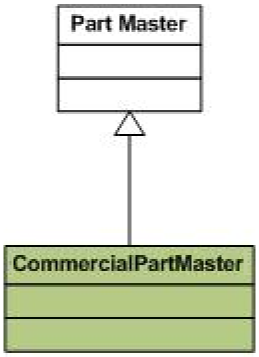
The **CommercialPart** business object is provided by the **vendormanagement** template.

The following table describes the attributes of the **CommercialPart** object.

Attribute name	Storage type	Description
adspart_category	**String [32]**	Designates the category of part for the program, for example, supplier part, government furnished equipment, or military standard part.
adsassembly_type	**String [32]**	Identifies the type of assembly for the part. Valid values are **Separable** and **Inseparable**. An inseparable assembly part cannot be disassembled without causing physical harm to at least one of the constituent parts in the assembly.
adsspecial_handling	**String [32]**	Indicates whether or not a part requires special handling.
adsdistribution_code	**String [32]**	

CommercialPartMaster

CommercialPartMaster is the storage class of the item master form corresponding to the **CommercialPart** item. The following figure illustrates the schema definition and attributes of the **CommercialPartMaster** class.



Preconditions and postactions for CommercialPart

The following table describes preconditions for the **CommercialPart** object.

Extension rule name	Operation name	Parameters
validateImport Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. Data is imported when the value of the constants is true and the data belongs to the same organization at the both the sites.	**TIE_deserialize**	
validateOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. If the value is set to true and the current user group belongs to an organization, the workspace object is created.	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
checkLatest_Released	**ITEM_copy_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	

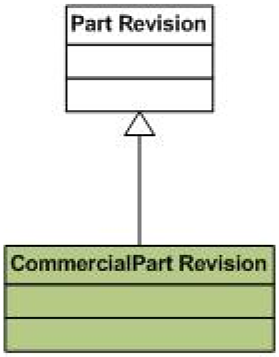
The following table describes postactions for the **CommercialPart** object.

Extension rule name	Operation name	Parameters
autoAssignToProject Assigns the object to the current project.	**ITEM_create** **IMAN_import** **ITEM_create_from_rev** **ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	NA
setOrgOnCreation Checks the value of the **TcSetOwningOrganization** global constant and the	**ITEM_create** **ITEM_create_from_rev** **ITEM_copy_rev**	

Extension rule name	Operation name	Parameters
AutoAssignOwningOrg business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	
Ads0AssociateChangeNotice	**ITEM_copy_rev** **ITEM_baseline_rev** **ITEM_copy_rev_to_existing** **ITEM_create_rev**	

CommercialPart Revision

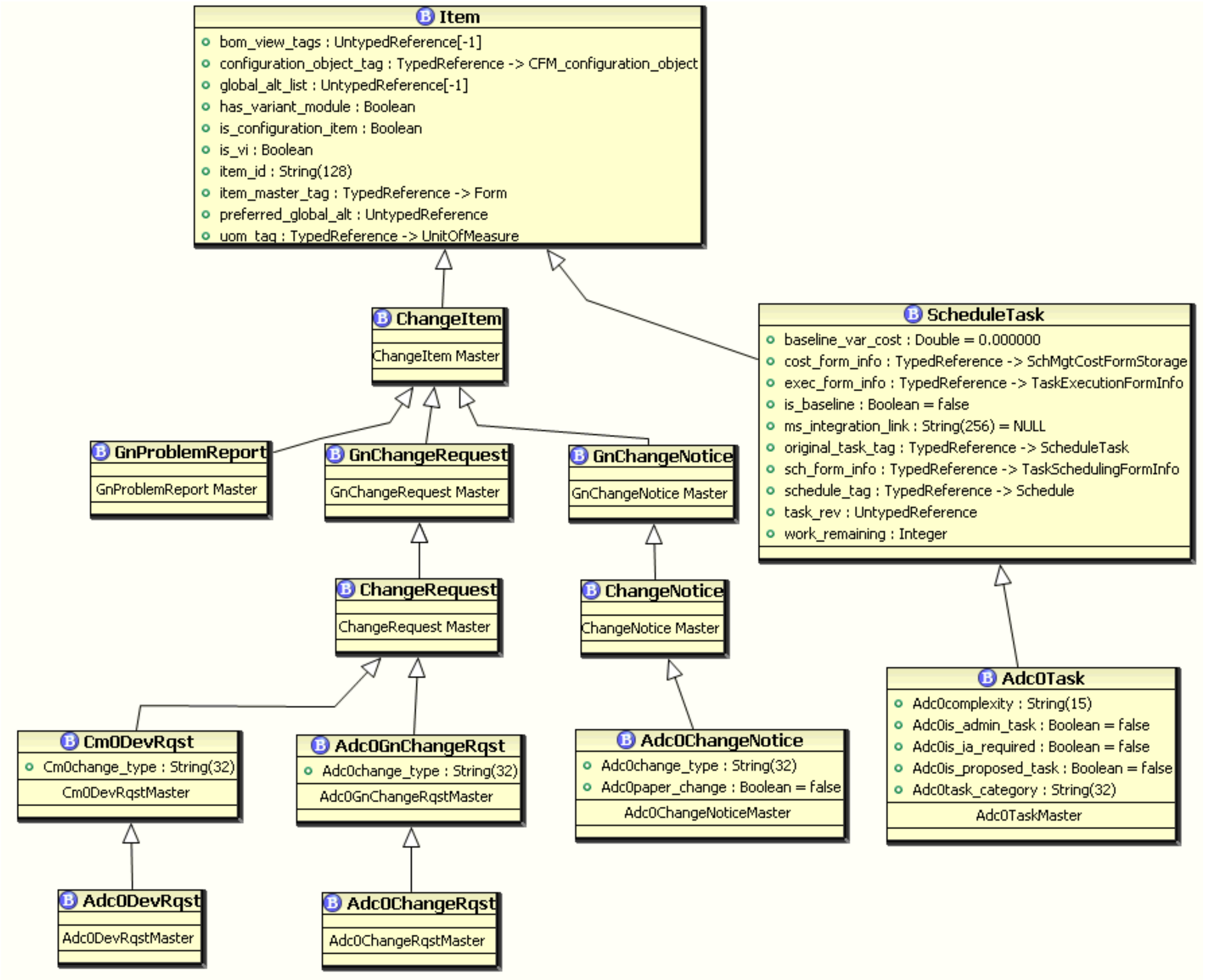
CommercialPart Revision is a subtype of the **Part Revision** business object and represents revisions corresponding to an commercial part. The following figure illustrates the **CommercialPart Revision** business object data model.



Aerospace and Defense change management

Aerospace and Defense template

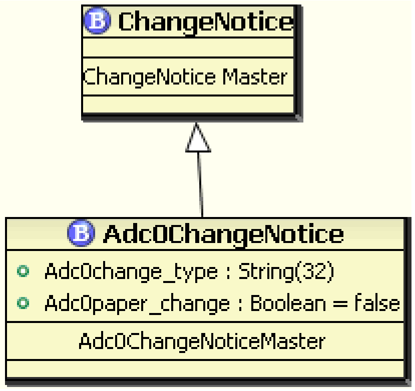
The **adschangemanagement** template includes the classes shown in the following diagram.


Item
bom_view_tags : UntypedReference[-1]
configuration_object_tag : TypedReference -> CFM_configuration_object
global_alt_list : UntypedReference[-1]
has_variant_module : Boolean
is_configuration_item : Boolean
is_vi : Boolean
item_id : String(128)
item_master_tag : TypedReference -> Form
preferred_global_alt : UntypedReference
uom_tag : TypedReference -> UnitOfMeasure
ChangeItem
ChangeItem Master
ScheduleTask
baseline_var_cost : Double = 0.000000
cost_form_info : TypedReference -> SchMgtCostFormStorage
exec_form_info : TypedReference -> TaskExecutionFormInfo
is_baseline : Boolean = false
ms_integration_link : String(256) = NULL
original_task_tag : TypedReference -> ScheduleTask
sch_form_info : TypedReference -> TaskSchedulingFormInfo
schedule_tag : TypedReference -> Schedule
task_rev : UntypedReference
work_remaining : Integer
GnProblemReport
GnProblemReport Master
GnChangeRequest
GnChangeRequest Master
GnChangeNotice
GnChangeNotice Master
ChangeRequest
ChangeRequest Master
ChangeNotice
ChangeNotice Master
Adc0Task
Adc0complexity : String(15)
Adc0is_admin_task : Boolean = false
Adc0is_ia_required : Boolean = false
Adc0is_proposed_task : Boolean = false
Adc0task_category : String(32)
Adc0TaskMaster
Cm0DevRqst
Cm0change_type : String(32)
Cm0DevRqstMaster
Adc0GnChangeRqst
Adc0change_type : String(32)
Adc0GnChangeRqstMaster
Adc0ChangeNotice
Adc0change_type : String(32)
Adc0paper_change : Boolean = false
Adc0ChangeNoticeMaster
Adc0DevRqst
Adc0DevRqstMaster
Adc0ChangeRqst
Adc0ChangeRqstMaster

Adc0ChangeNotice

Adc0ChangeNotice is a subtype of the **ChangeNotice** item type.

The following figure illustrates the **Adc0ChangeNotice** data model.



Adc0ChangeNotice attributes

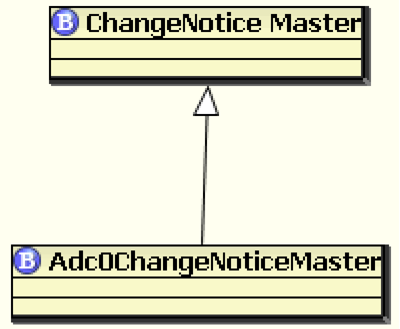
The **Adc0ChangeNotice** item is provided by the **adschangemanagement** template.

The following table describes the attributes of the **Adc0ChangeNotice** class.

Attribute name	Storage type	Description
Adc0change_type	**String [32]**	List of values that specifies the type of change documentation for the program, for example, **ADCN**, **DCN**, and **Redline**.
Adc0paper_change	**Boolean**	If true, change is incorporated. If false, change is not incorporated.

Adc0ChangeNoticeMaster

Adc0ChangeNoticeMaster is the storage class of the item master form corresponding to the **Adc0ChangeNotice** item. The following figure illustrates the schema definition and attributes of the **Adc0ChangeNoticeMaster** class.



Preconditions and postactions on Adc0ChangeNotice

The following table describes preconditions for the **Adc0ChangeNotice** object.

Extension rule name	Description	Operation name
validateOrgOnCreation	Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value is set to true and the current user	**ITEM_create** **ITEM_create_from_rev**

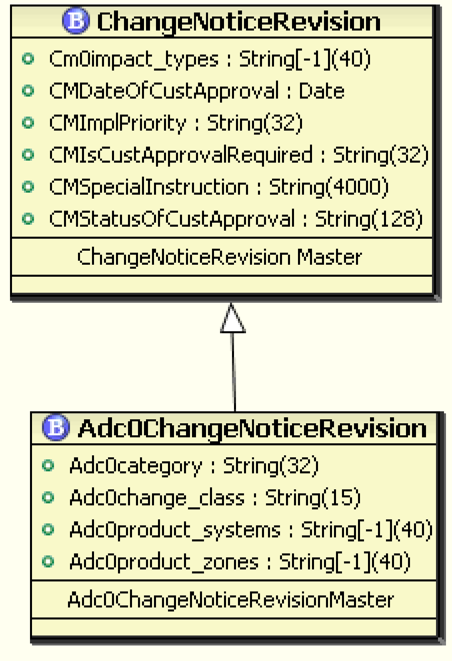
Extension rule name	Description	Operation name
	group belongs to an organization, the workspace object is created.	
validateImport	Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. Data is imported when the value of the constants is true and the data belongs to the same organization at the both the sites.	**TIE_deserialize**

The following table describes postactions for the **Adc0ChangeNotice** object.

Extension rule name	Description	Operation name
autoAssignToProject	Assigns the object to the current project.	**IMAN_import** **ITEM_create** **ITEM_create_from_rev**
setOrgOnCreation	Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**ITEM_create** **ITEM_create_from_rev**
setOrgOnImport	Checks the value of the **TcSetOwningOrganization** global constant and the **AutoAssignOwningOrg** business constant. When the value of the constants is true and the current group belongs to an organization, the **owning_organization** attribute value of the object is automatically updated.	**TIE_deserialize**

Adc0ChangeNoticeRevision

Adc0ChangeNoticeRevision is a subtype of the **ChangeNoticeRevision** business object and represents revisions corresponding to an **Adc0ChangeNotice**. The following figure illustrates the **Adc0ChangeNoticeRevision** business object data model.


ChangeNoticeRevision
Cm0impact_types : String[-1](40)
CMDateOfCustApproval : Date
CMImplPriority : String(32)
CMIsCustApprovalRequired : String(32)
CMSpecialInstruction : String(4000)
CMStatusOfCustApproval : String(128)
ChangeNoticeRevision Master
Adc0ChangeNoticeRevision
Adc0category : String(32)
Adc0change_class : String(15)
Adc0product_systems : String[-1](40)
Adc0product_zones : String[-1](40)
Adc0ChangeNoticeRevisionMaster

Adc0ChangeNoticeRevision attributes

The **Adc0ChangeNoticeRevision** item is provided by the **adschangemanagement** template.

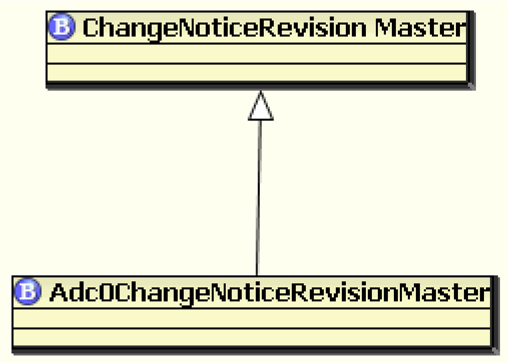
The following table describes the attributes of the **Adc0ChangeNoticeRevision** class.

Attribute name	Storage type	Description
Adc0category	**String [32]**	Designates the specific category of change within the change classification.
Adc0change_class	**String [15]**	List of values that designates the government's or customer's classification to a change, for example, **I** or **II**.
Adc0product_systems	**String [40]**	List of values that designates the systems or subsystems for an item. A product system denotes a group of parts that perform a common function, for example, **Electrical**, **Environmental**

Attribute name	Storage type	Description
		Control, and **Weapons**.
Adc0product_zones	**String [40]**	Designates the zones for an item. A zone denotes an area or section of the product used in managing multiple system interfaces and structural interferences. For example, **Baggage Compartment**, **Electrical**, and **Plumbing**.

Adc0ChangeNoticeRevMaster

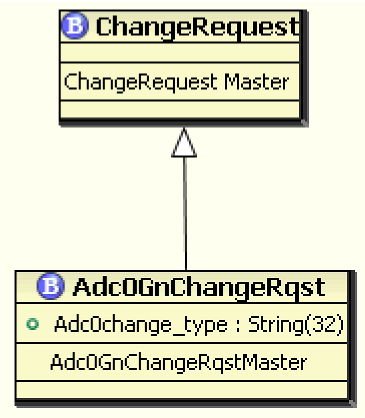
Adc0ChangeNoticeRevMaster is the storage class of the item revision master form.



Adc0GnChangeRqst

Adc0GnChangeRqst is a subtype of the **ChangeRequest** item type.

The following figure illustrates the **Adc0GnChangeRqst** data model.



Adc0GnChangeRqst attributes

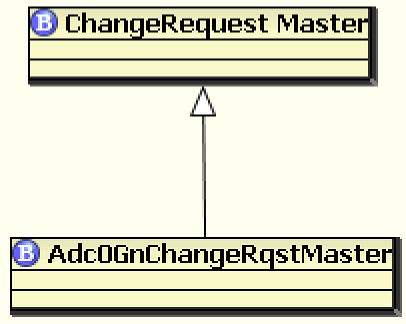
The **Adc0GnChangeRqst** item is provided by the **adschangemanagement** template.

The following table describes the attributes of the **Adc0GnChangeRqst** class.

Attribute name	Storage type	Description
Adc0change_type	**String [32]**	List of values that specifies the type of change documentation for the program, for example, **ADCN**, **DCN**, and **Redline**.

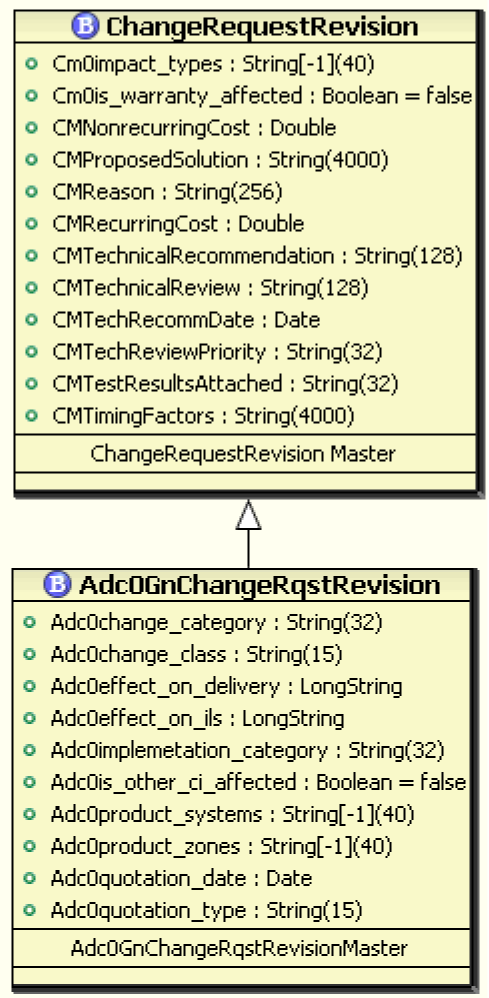
Adc0GnChangeRqstMaster

Adc0GnChangeRqstMaster is the storage class of the item master form corresponding to the **Adc0GnChangeRqst** item. The following figure illustrates the schema definition and attributes of the **Adc0GnChangeRqstMaster** class.



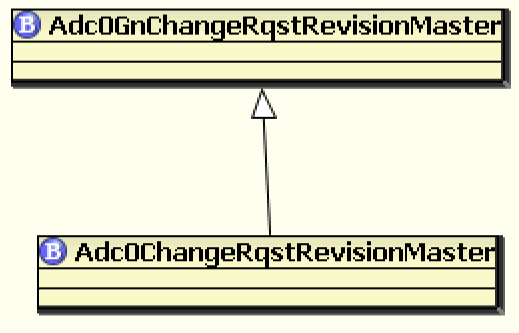
Adc0GnChangeRqstRevision

Adc0GnChangeRqstRevision is a subtype of the **ChangeRequestRevision** business object and represents revisions corresponding to an **Adc0GnChangeRqst**. The following figure illustrates the **Adc0GnChangeRqstRevision** business object data model.


ChangeRequestRevision
Cm0impact_types : String[-1](40)
Cm0is_warranty_affected : Boolean = false
CMNonrecurringCost : Double
CMProposedSolution : String(4000)
CMReason : String(256)
CMRecurringCost : Double
CMTechnicalRecommendation : String(128)
CMTechnicalReview : String(128)
CMTechRecommDate : Date
CMTechReviewPriority : String(32)
CMTestResultsAttached : String(32)
CMTimingFactors : String(4000)
ChangeRequestRevision Master
Adc0GnChangeRqstRevision
Adc0change_category : String(32)
Adc0change_class : String(15)
Adc0effect_on_delivery : LongString
Adc0effect_on_ils : LongString
Adc0implemetation_category : String(32)
Adc0is_other_ci_affected : Boolean = false
Adc0product_systems : String[-1](40)
Adc0product_zones : String[-1](40)
Adc0quotation_date : Date
Adc0quotation_type : String(15)
Adc0GnChangeRqstRevisionMaster

Adc0GnChangeRqstRevMaster

Adc0GnChangeRqstRevMaster is the storage class of the item revision master form.



Adc0GnChangeRqstRevision attributes

The **Adc0GnChangeRqstRevision** item is provided by the **adschangemanagement** template.

The following table describes the attributes of the **Adc0GnChangeRqstRevision** class.

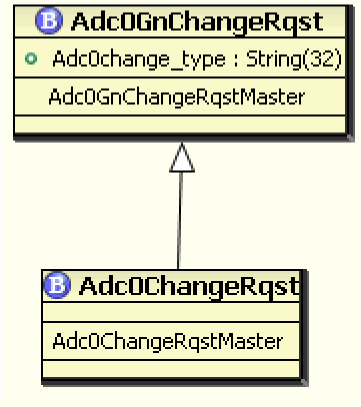
Attribute name	Storage type	Description
Adc0category	**String [32]**	Designates the specific category of change within the change classification.
Adc0change_class	**String [15]**	List of values that designates the government's or customer's classification to a change, for example, **I** or **II**.
Adc0effect_on_delivery	**LongString**	Details the effects of the change on the delivery schedule.
Adc0effect_on_ils	**LongString**	Details the effects of the change on affected parts, materials, and software.
Adc0implementation_category	**String [32]**	Designates the specific category of the change implementation, for example, **Prior to first flight** or **Post flight test**.
Adc0is_other_ci_affected	**Boolean**	If true, other systems or configuration items are affected. If false, no other systems or configuration items are affected by this change.

Attribute name	Storage type	Description
Adc0product_systems	**String [40]**	List of values that designates the systems or subsystems for an item. A product system denotes a group of parts that perform a common function, for example, **Electrical**, **Environmental Control**, and **Weapons**.
Adc0product_zones	**String [40]**	Designates the zones for an item. A zone denotes an area or section of the product used in managing multiple system interfaces and structural interferences, for example, **Baggage Compartment**, **Electrical**, and **Plumbing**.
Adc0quotation_date	**Date**	Designates the date the quotation type was assigned.
Adc0quotation_type	**String [15]**	Designates the quotation type for the change impact. For example, **Supplier ROM**, **Buyer Estimate**, or **Supplier Firm**.

Adc0ChangeRqst

Adc0ChangeRqst is a subtype of the **Adc0GnChangeRqst** item type.

The following figure illustrates the **Adc0ChangeRqst** data model.



Adc0ChangeRqst attributes

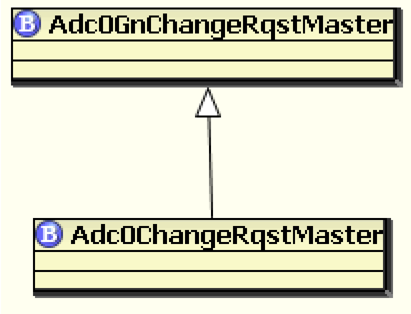
The **Adc0ChangeRqst** item is provided by the **adschangemanagement** template.

The following table describes the attributes of the **AAdc0ChangeRqst** class.

Attribute name	Storage type	Description
Adc0change_type	**String [32]**	List of values that specifies the type of change documentation for the program, for example, **ADCN**, **DCN**, and **Redline**.

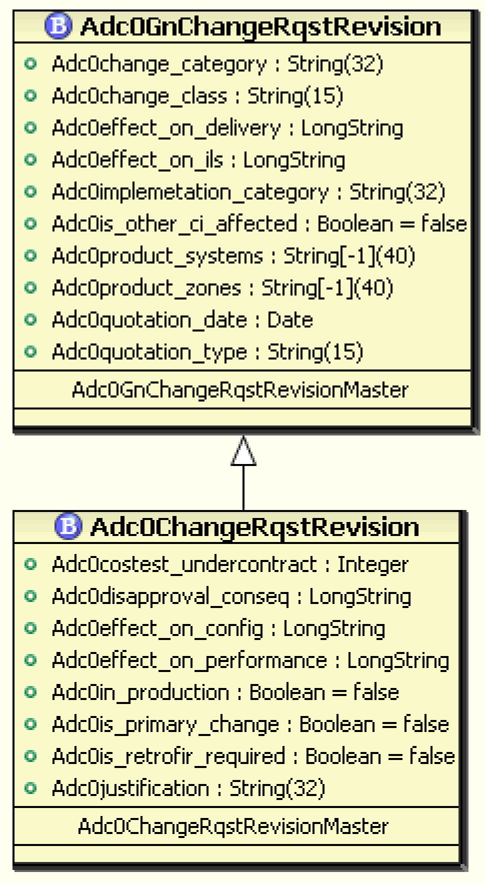
Adc0ChangeRqstMaster

Adc0ChangeRqstMaster is the storage class of the item master form. The following figure illustrates the schema definition and attributes of the **Adc0ChangeRqstMaster** class.



Adc0ChangeRqstRevision

Adc0ChangeRqstRevision is a subtype of the **Adc0GnChangeRqstRevision** business object and represents revisions corresponding to an **Adc0ChangeRqstRevision**. The following figure illustrates the **Adc0ChangeRqstRevision** business object data model.


Adc0GnChangeRqstRevision
Adc0change_category : String(32)
Adc0change_class : String(15)
Adc0effect_on_delivery : LongString
Adc0effect_on_ils : LongString
Adc0implemetation_category : String(32)
Adc0is_other_ci_affected : Boolean = false
Adc0product_systems : String[-1](40)
Adc0product_zones : String[-1](40)
Adc0quotation_date : Date
Adc0quotation_type : String(15)
Adc0GnChangeRqstRevisionMaster
Adc0ChangeRqstRevision
Adc0costest_undercontract : Integer
Adc0disapproval_conseq : LongString
Adc0effect_on_config : LongString
Adc0effect_on_performance : LongString
Adc0in_production : Boolean = false
Adc0is_primary_change : Boolean = false
Adc0is_retrofir_required : Boolean = false
Adc0justification : String(32)
Adc0ChangeRqstRevisionMaster

Adc0ChangeRqstRevision attributes

The **Adc0ChangeRqstRevision** item is provided by the **adschangemanagement** template.

The following table describes the attributes of the **Adc0ChangeRqstRevision** class.

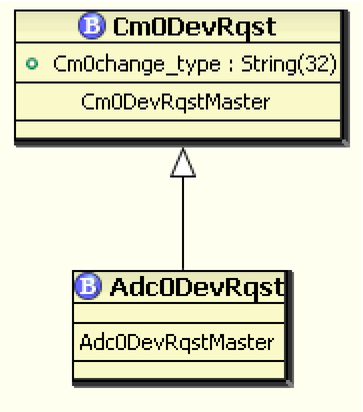
Attribute name	Storage type	Description
Adc0costest_undercontract	**Integer**	Designates the estimated cost under contract for approving and implementing the change request. The number is expressed in

Attribute name	Storage type	Description
		currency, for example, dollars or euros.
Adc0disapproval_conseq	**LongString**	Details the consequences if the change is not approved.
Adc0effect_on_config	**LongString**	Details the effect on the configuration specification for the change.
Adc0effect_on_performance	**LongString**	Details the effect on the product performance for the change.
Adc0in_production	**Boolean**	If true, the change has been implemented to production. If false, the change is not yet in production.
Adc0is_primary_change	**Boolean**	If true, this is the primary change with related changes. If false, this is a secondary or subsidiary change.
Adc0is_retrofit_required	**Boolean**	If true, the change recommends retrofit of existing units. If false, no retrofit is required.
Adc0justification	**String [32]**	Details the justification or reason for the preparation of the change.

Adc0DevRqst

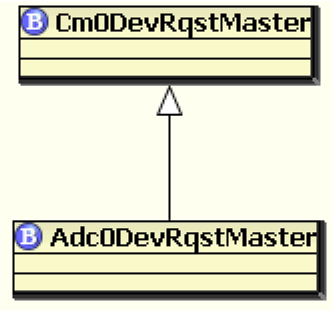
Adc0DevRqst is a subtype of the **CMm0ChangeRqst** item type.

The following figure illustrates the **Adc0DevRqst** data model.



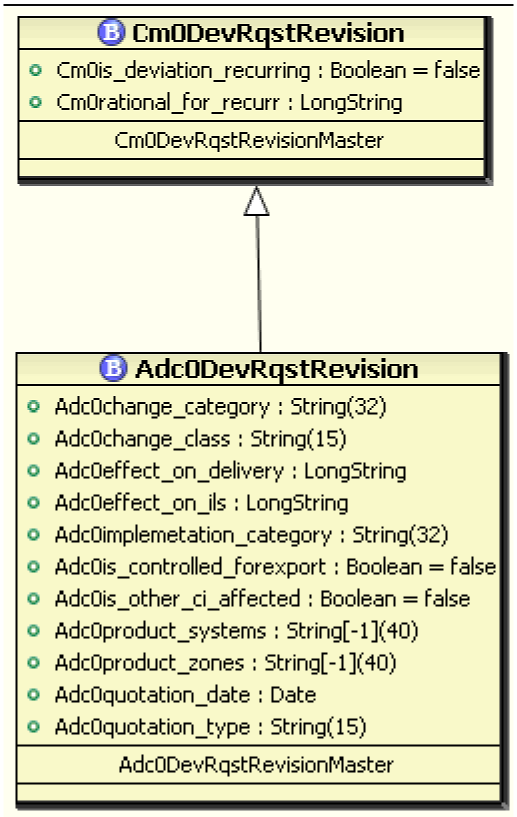
Adc0DevRqstMaster

Adc0DevRqstMaster is the storage class of the item master form. The following figure illustrates the schema definition and attributes of the **Adc0DevRqstMaster** class.



Adc0DevRqstRevision

Adc0DevRqstRevision is a subtype of the **Cm0DevRqstRevision** business object and represents revisions corresponding to an **Adc0DevRqst**. The following figure illustrates the **Adc0DevRqstRevision** business object data model.


Cm0DevRqstRevision
Cm0is_deviation_recurring : Boolean = false
Cm0rational_for_recurr : LongString
Cm0DevRqstRevisionMaster
Adc0DevRqstRevision
Adc0change_category : String(32)
Adc0change_class : String(15)
Adc0effect_on_delivery : LongString
Adc0effect_on_ils : LongString
Adc0implemetation_category : String(32)
Adc0is_controlled_forexport : Boolean = false
Adc0is_other_ci_affected : Boolean = false
Adc0product_systems : String[-1](40)
Adc0product_zones : String[-1](40)
Adc0quotation_date : Date
Adc0quotation_type : String(15)
Adc0DevRqstRevisionMaster

Adc0DevRqstRevision attributes

The **Adc0DevRqstRevision** item is provided by the **adschangemanagement** template.

The following table describes the attributes of the **Adc0DevRqstRevision** class.

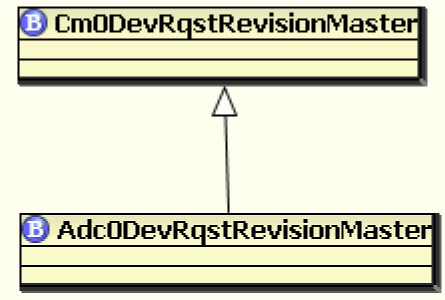
Attribute name	Storage type	Description
Adc0category	**String [32]**	Designates the specific category of change within the change classification.
Adc0change_class	**String [15]**	List of values that designates the government's or customer's classification to a change, for example, **I** or **II**.
Adc0effect_on_delivery	**LongString**	Details the effects of the change on the delivery schedule.
Adc0effect_on_ils	**LongString**	Details the effects of the change on affected

Attribute name	Storage type	Description
		parts, materials, and software.
Adc0implementation_category	**String [32]**	Designates the specific category of the change implementation, for example, **Prior to first flight** or **Post flight test**.
Adc0is_controlled_forexport	**Boolean**	If true, change is under export control. If false, change is not under export control.
Adc0is_other_ci_affected	**Boolean**	If true, other systems or configuration items are affected. If false, no other systems or configuration items are affected by this change.
Adc0product_systems	**String [40]**	List of values that designates the systems or subsystems for an item. A product system denotes a group of parts that perform a common function, for example, **Electrical**, **Environmental Control**, and **Weapons**.
Adc0product_zones	**String [40]**	Designates the zones for an item. A zone denotes an area or section of the product used in managing multiple system interfaces and structural interferences, for example, **Baggage Compartment**, **Electrical**, and **Plumbing**.
Adc0quotation_date	**Date**	Designates the date the quotation type was assigned.
Adc0quotation_type	**String [15]**	Designates the quotation type for the change impact, for example, **Supplier**

Attribute name	Storage type	Description
		ROM, **Buyer Estimate**, or **Supplier Firm**.

Adc0DevRqstRevMaster

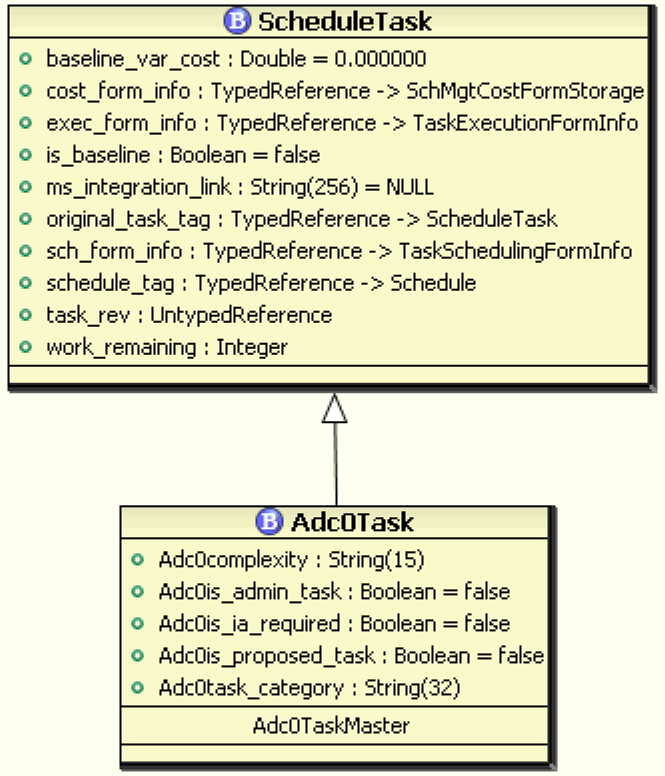
Adc0DevRqstRevMaster is the storage class of the item master form corresponding to the **Adc0DevRqstRev** item. The following figure illustrates the schema definition and attributes of the **Adc0DevRqstRevMaster** class.



Adc0Task

Adc0Task is a subtype of the **ScheduleTask** item type.

The following figure illustrates the **Adc0Task** data model.


ScheduleTask
baseline_var_cost : Double = 0.000000
cost_form_info : TypedReference -> SchMgtCostFormStorage
exec_form_info : TypedReference -> TaskExecutionFormInfo
is_baseline : Boolean = false
ms_integration_link : String(256) = NULL
original_task_tag : TypedReference -> ScheduleTask
sch_form_info : TypedReference -> TaskSchedulingFormInfo
schedule_tag : TypedReference -> Schedule
task_rev : UntypedReference
work_remaining : Integer
Adc0Task
Adc0complexity : String(15)
Adc0is_admin_task : Boolean = false
Adc0is_ia_required : Boolean = false
Adc0is_proposed_task : Boolean = false
Adc0task_category : String(32)
Adc0TaskMaster

Adc0Task attributes

The **Adc0Task** item is provided by the **adschangemanagement** template.

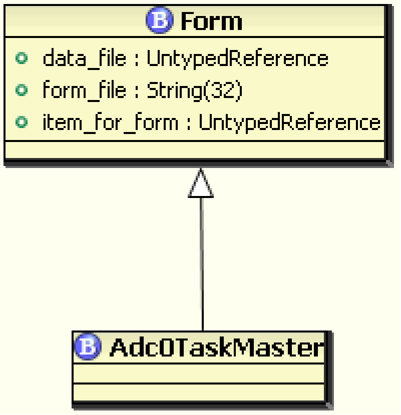
The following table describes the attributes of the **Adc0Task** class.

Attribute name	Storage type	Description
Adc0complexity	**String [15]**	Designates the complexity of the task, for example, **high**, **medium**, or **low**.
Adc0is_admin_task	**Boolean**	If true, this task is assigned to an administrator to complete. If false, it is assigned to any user to complete.
Adc0is_ia_required	**Boolean**	If true, impact analysis is required by impacted supplier or organization. If false, no impact analysis is required.

Attribute name	Storage type	Description
Adc0is_proposed_task	**Boolean**	If true, this task is only proposed and many not be completed. If false, this task will be completed.
Adc0task_category	**String [32]**	List of values that designates the category of the task, for example, **Agenda Event**, **Planning**, or **QA**.

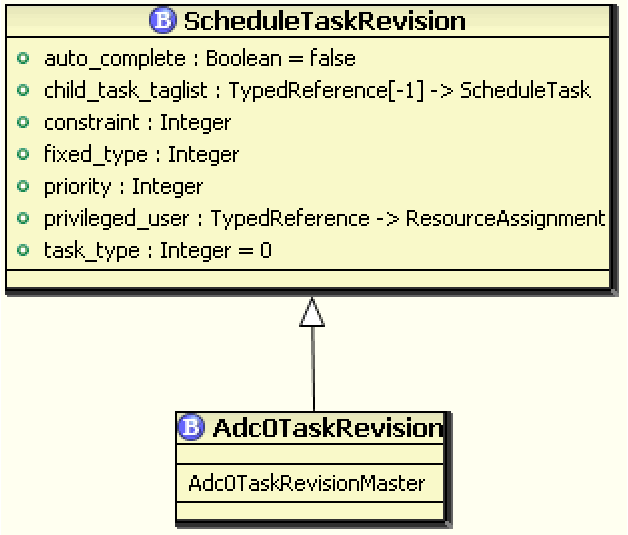
Adc0TaskMaster

Adc0TaskMaster is the storage class of the item master form corresponding to the **Adc0Task** item. The following figure illustrates the schema definition and attributes of the **Adc0TaskMaster** class.


B Form
data_file : UntypedReference
form_file : String(32)
item_for_form : UntypedReference
B Adc0TaskMaster

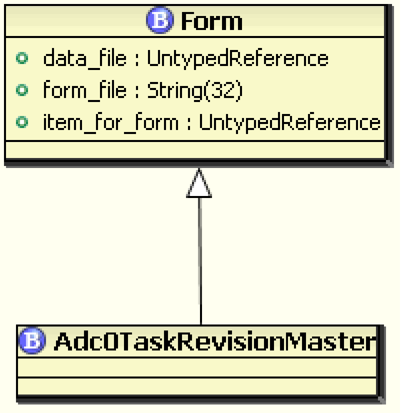
Adc0TaskRevision

The following figure illustrates the **Adc0TaskRevision** business object data model.


ScheduleTaskRevision
auto_complete : Boolean = false
child_task_taglist : TypedReference[-1] -> ScheduleTask
constraint : Integer
fixed_type : Integer
priority : Integer
privileged_user : TypedReference -> ResourceAssignment
task_type : Integer = 0
Adc0TaskRevision
Adc0TaskRevisionMaster

Adc0TaskRevisionMaster

The following figure illustrates the schema definition and attributes of the **Adc0ChangeRqstMaster** class.


Form
data_file : UntypedReference
form_file : String(32)
item_for_form : UntypedReference
Adc0TaskRevisionMaster

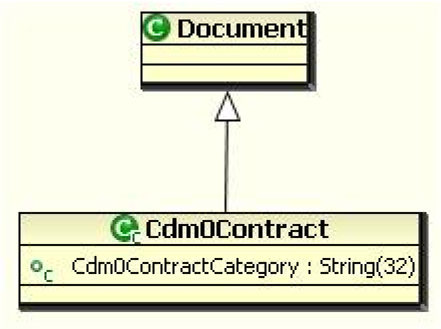
Contract data management objects

Contract data management objects

Cdm0Contract is a subtype of the **Document** item type and represents the contract document used in document-centric programs.

Cdm0Contract is a primary business object that stores and represents contract data. A *contract* is a structured procurement document listing the milestones and the schedule dates of the contract event.

The following figure illustrates the **Cdm0Contract** data model.



Cdm0Contract attributes

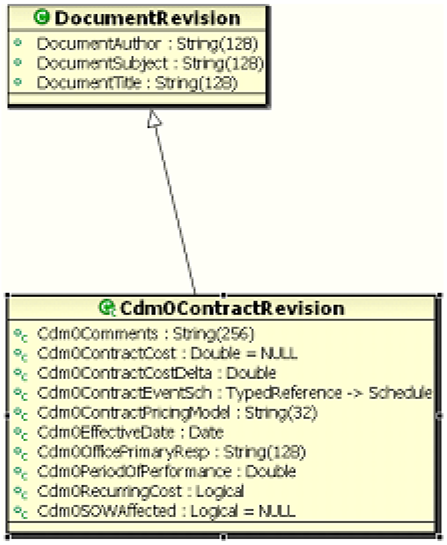
Cdm0Contract items are identified in the system by a **Cdm0** property. The **Cdm0Contract** item is provided by the **contractmanagement** template.

The following table lists the attributes of the **Cdm0** class.

Attribute name	Storage type	Description
Cdm0Contract Category	**String(32)**	Classifies contract item, for example, **CONTRACT**, **TWO**, and **PO**.

Cdm0ContractRevision attribute

The **Cdm0ContractRevision** class is used to store and represent contract revision data. Data requirement items are associated with the contract revision data.

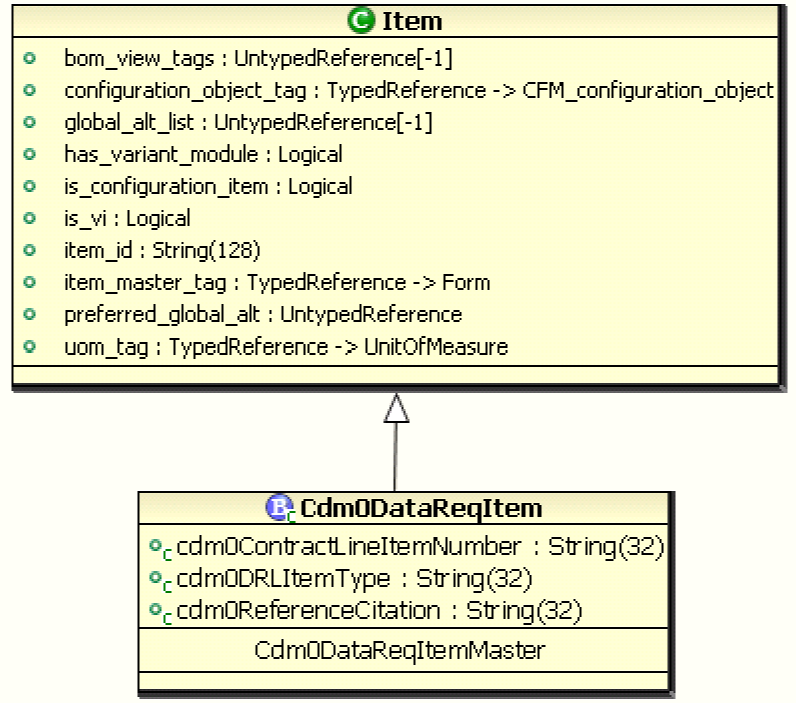

DocumentRevision
DocumentAuthor : String(128)
DocumentSubject : String(128)
DocumentTitle : String(128)
Cdm0ContractRevision
Cdm0Comments : String(256)
Cdm0ContractCost : Double = NULL
Cdm0ContractCostDelta : Double
Cdm0ContractEventSch : TypedReference -> Schedule
Cdm0ContractPricingModel : String(32)
Cdm0EffectiveDate : Date
Cdm0OfficePrimaryResp : String(128)
Cdm0PeriodOfPerformance : Double
Cdm0RecurringCost : Logical
Cdm0SOWAffected : Logical = NULL

The following table lists the attributes of the **Cdm0ContractRevision** class.

Attribute name	Storage type	Description
Cdm0ContractCost	**Double**	Specifies the total cost of the contract. Contract cost also includes the addition to the cost delta incurred due to modification.
Cdm0Contract PricingModel	**String(32)**	Specifies the payment forms for the contract, for example, cost plus fixed fee.
Cdm0EffectiveDate	**Date**	Specifies the date from which the contract is effective.
Cdm0Contract EventSch	**Typed Reference** (Contract event schedule)	Specifies the milestones for the contract.
Cdm0Comments	**String(256)**	Specifies any special remarks related to the contract.
Cdm0Contract CostDelta	**Double**	Specifies the relationship between the current contract cost and the inherent future contract cost that may arise due to modification of the contract.
Cdm0Office PrimaryResp	**String(128)**	Specifies the name of the person or a role responsible for the review and acceptance of the contract.
Cdm0Period OfPerformance	**Double**	Specifies the duration of the contract. The duration the time period as specified in the contract. It is expressed in months.
Cdm0Recurring Cost	**Logical**	Specifies whether the contract cost is recurring cost.
Cdm0SOWAffected	**Logical**	Specifies whether the statement of work for the contract is affected due to contract modification.

Cdm0DataReqItem attribute

The **Cdm0DataReqItem** class is used to store and represent data requirement item data. Data requirement items data are associated with the contract revision.

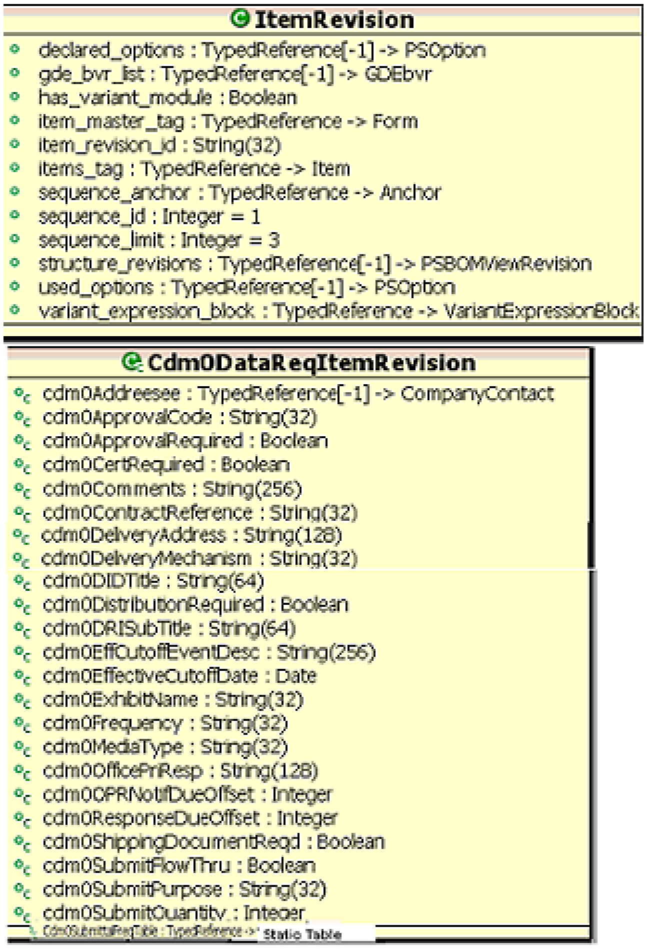

Item
bom_view_tags : UntypedReference[-1]
configuration_object_tag : TypedReference -> CFM_configuration_object
global_alt_list : UntypedReference[-1]
has_variant_module : Logical
is_configuration_item : Logical
is_vi : Logical
item_id : String(128)
item_master_tag : TypedReference -> Form
preferred_global_alt : UntypedReference
uom_tag : TypedReference -> UnitOfMeasure
Cdm0DataReqItem
cdm0ContractLineItemNumber : String(32)
cdm0DRLItemType : String(32)
cdm0ReferenceCitation : String(32)
Cdm0DataReqItemMaster

The following table lists the attributes of the **Cdm0DataReqItem** class.

Attribute name	Storage type	Description
Cdm0ContractLine ItemNumber	**String(32)**	Specifies the section ID within the contract. This section describes how a specific task must be accomplished.
Cdm0Contract Reference	**String(128)**	The unique identifier for the contract on which the part is created.
Cdm0Contract TypedRef	**Typed Reference**	Specifies the contract.
cdm0ProvideRec ContractDel	**Boolean**	Specifies the type of contract data deliverable that is submitted by the customer or the supplier. Select**True** to provide contract deliverable for the supplier to review. Select **False** to receive contract deliverable from the supplier.
Cdm0Reference Citation	**String(32)**	Specifies the ID of a specific reference content in the document.

Cdm0DataReqItemRevision attribute

The **Cdm0DataReqItemRevision** class is used to store and represent data requirement item revision data.


ItemRevision
declared_options : TypedReference[-1] -> PSOption
gde_bvr_list : TypedReference[-1] -> GDEbvr
has_variant_module : Boolean
item_master_tag : TypedReference -> Form
item_revision_id : String(32)
items_tag : TypedReference -> Item
sequence_anchor : TypedReference -> Anchor
sequence_id : Integer = 1
sequence_limit : Integer = 3
structure_revisions : TypedReference[-1] -> PSBOMViewRevision
used_options : TypedReference[-1] -> PSOption
variant_expression_block : TypedReference -> VariantExpressionBlock
Cdm0DataReqItemRevision
cdm0Addressee : TypedReference[-1] -> CompanyContact
cdm0ApprovalCode : String(32)
cdm0ApprovalRequired : Boolean
cdm0CertRequired : Boolean
cdm0Comments : String(256)
cdm0ContractReference : String(32)
cdm0DeliveryAddress : String(128)
cdm0DeliveryMechanism : String(32)
cdm0DIDTitle : String(64)
cdm0DistributionRequired : Boolean
cdm0DRISubTitle : String(64)
cdm0EffCutoffEventDesc : String(256)
cdm0EffectiveCutoffDate : Date
cdm0ExhibitName : String(32)
cdm0Frequency : String(32)
cdm0MediaType : String(32)
cdm0OfficePriResp : String(128)
cdm0OPRNotifDueOffset : Integer
cdm0ResponseDueOffset : Integer
cdm0ShippingDocumentReqd : Boolean
cdm0SubmitFlowThru : Boolean
cdm0SubmitPurpose : String(32)
cdm0SubmitQuantity : Integer
Cdm0SubmittalReqTable : TypedReference -> Static Table

The following table lists the attributes of the **Cdm0DataReqItemRevision** class.

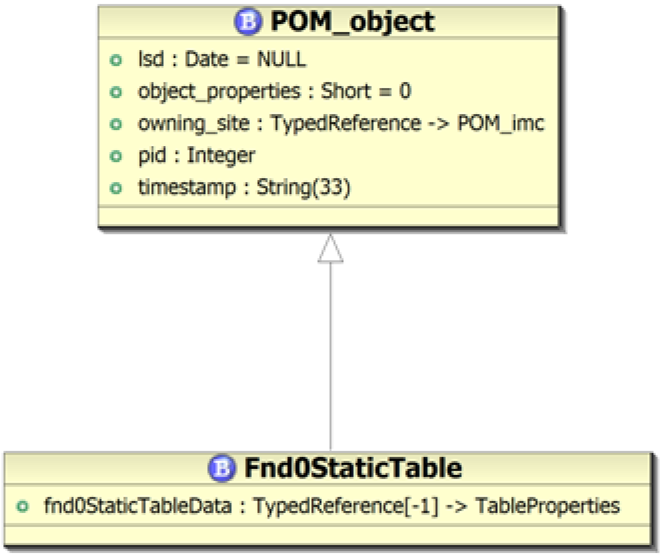
Attribute name	Storage type	Description
cdm0Addressee	**Typed reference** (company contact)	Specifies the list of contacts who receive the contract documents.
cdm0ApprovalCode	**String(32)**	Specifies the approval code for the data requirement item revision.
cdm0Approval Required	**Logical**	Specifies whether the customer must approve the data item.
cdm0CertRequired	**Logical**	Specifies whether the submitted data item requires certification.

Attribute name	Storage type	Description
cdm0Comments	**String(256)**	Specifies any comments concerning the item.
cdm0Contact Reference	**String(32)**	Specifies the ID of the contact reference.
cdm0DataMgmt DueOffset	**Integer**	Specifies the offset of the data management due date.
cdm0Delivery Address	**String(128)**	Specifies the delivery address of the customer or the supplier.
cdm0Delivery Mechanism	**String(32)**	Specifies the mode of delivery of the contract.
cdm0DIDTitle	**String(64)**	Specifies the title of the data item description document.
cdm0Distribution Required	**Logical**	Specifies whether any documentation related to the item should be distributed.
cdm0DRISubTitle	**String(64)**	Specifies the subtitle of the data requirement item.
cdm0EffCutoff EventDesc	**String(256)**	Specifies the date that data collection for the document is to stop.
cdm0Effective CutoffDate	**Date**	Specifies the last date for data collection.
cdm0EventList	**Typed Reference** (Fnd0StaticTable)	Specifies the event table for generating event schedule.
Cdm0ExhibitName	**String(32)**	Specifies the ID of a particular data item in a data item description document.
Cdm0MediaType	**String**	Specifies the format of the data, for example, electrostatic, microfilm, and so on.
Cdm0OfficePriResp	**String(128)**	Specifies the name of the contacts, groups, or role responsible for reviewing the data item.
Cdm0OPRNotifDue Offset	**Integer**	Specifies the last date for the creation of the notification to the office of the primary reviewers.
Cdm0Response DueOffset	**Integer**	Specifies the offset to the response due date.
Cdm0Shipping DocumentReqd	**Logical**	Specifies whether **DD Form 250** is required for the delivery of the data item.

Attribute name	Storage type	Description
Cdm0SubmitFlowThru	**Logical**	Specifies whether a supplier's data requirement item should be submitted as a customer's data requirement item.
Cdm0SubmitPurpose	**String(32)**	Specifies the purpose of the submitted data item.
Cdm0SubmitQuantity	**Integer**	Specifies the number of copies required for submission.

Fnd0StaticTable attribute

The **Fnd0StaticTable** class is used to store the rows for tabular data.

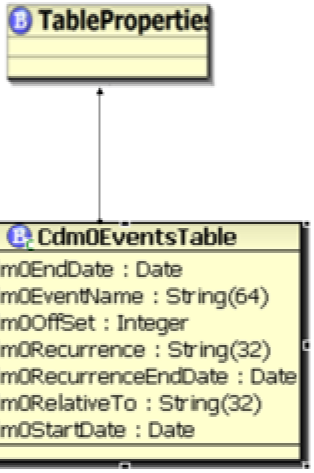

POM_object
lsd : Date = NULL
object_properties : Short = 0
owning_site : TypedReference -> POM_imc
pid : Integer
timestamp : String(33)
Fnd0StaticTable
fnd0StaticTableData : TypedReference[-1] -> TableProperties

The following table lists the attributes of the **Fnd0StaticTable** class.

Attribute name	Storage type	Description
fnd0StaticTableData	**TypedReference** (TableProperties)	Specifies the variable length array attribute of the **TableProperties** class that stores the rows for the data in the event table.

Cdm0EventsTable attribute

The **Cdm0EventsTable** class is used to store event data in the data requirement item (DRI) event table.

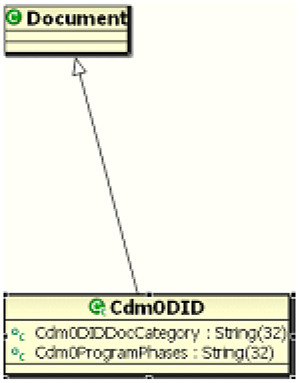

TableProperties
Cdm0EventsTable
cdm0EndDate : Date
cdm0EventName : String(64)
cdm0OffSet : Integer
cdm0Recurrence : String(32)
cdm0RecurrenceEndDate : Date
cdm0RelativeTo : String(32)
cdm0StartDate : Date

The following table lists the attributes of the **Cdm0EventsTable** class.

Attribute name	Storage type	Description
cdm0EndDate	**Date**	Specifies the event end date.
cdm0EventName	**String(64)**	Specifies the contract event name.
cdm0OffSet	**Integer**	Specifies the offset used for calculating the submittal due date for calculating the first submittal delivery.
cdm0Recurrence	**String(32)**	Specifies frequency of the submittal delivery event.
cdm0StartDate	**Date**	Specifies the event start date.
cdm0RelativeTo	**String(32)**	Denotes the relation of the offset to either the start date or the end date.
cdm0Recurrence EndDate	**Date**	Specifies the last date for event schedule.

Cdm0DID attribute

The **Cdm0DID** class is used to store and represent data item description data. A data item description specifies the content and the format of the data item requirement item.

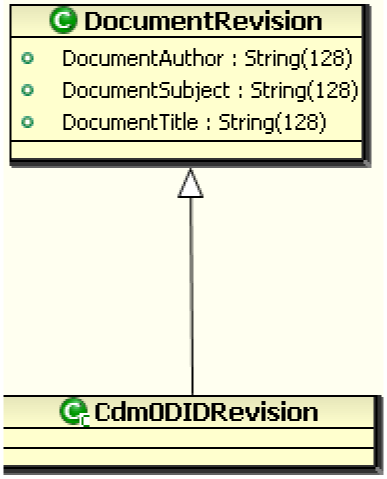

Document
Cdm0DID
Cdm0DIDDocCategory : String(32)
Cdm0ProgramPhases : String(32)

The following table lists the attributes of the **Cdm0EventsTable** class.

Attribute name	Storage type	Description
Cdm0DIDDocCategory	**String(32)**	Specifies user-defined attribute for classifying document items.
Cdm0ProgramPhases	**String(32)**	Specifies the program application for the specified DID type.

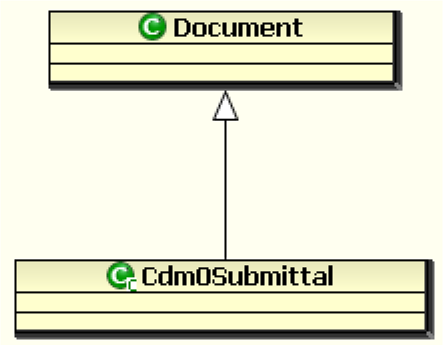
Cdm0DIDRevision attribute

The **Cdm0DIDRevision** class is used to represent relation between the data item description and **Cdm0DataReqItemRevision** class.


DocumentRevision
DocumentAuthor : String(128)
DocumentSubject : String(128)
DocumentTitle : String(128)
Cdm0DIDRevision

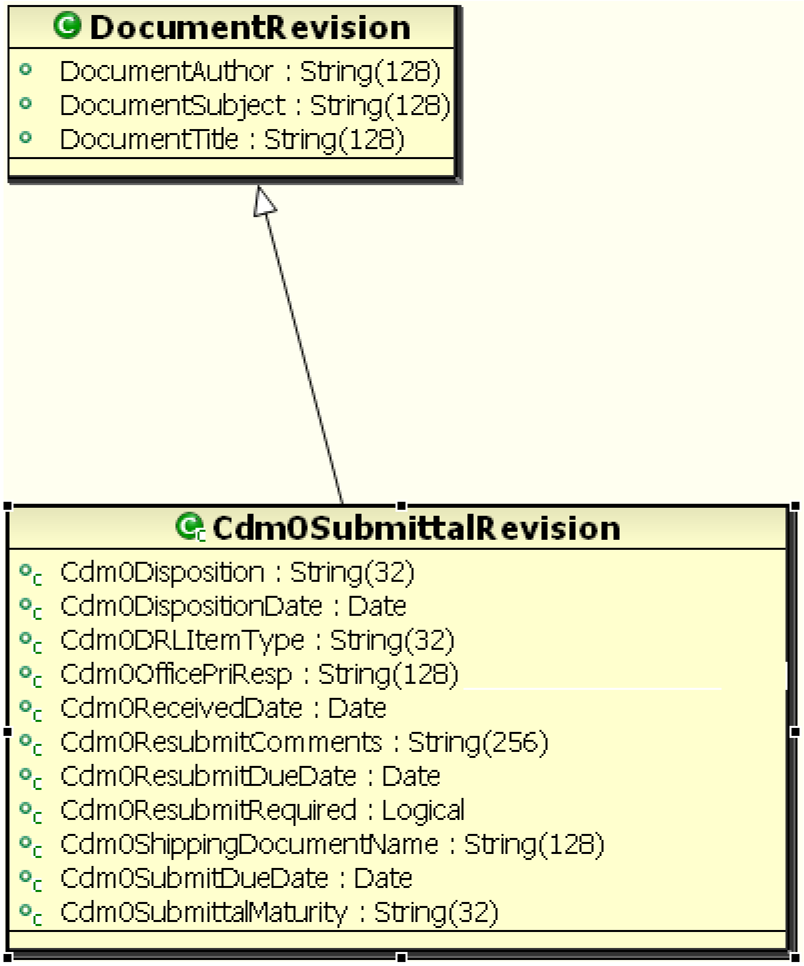
Cdm0Submittal attribute

The **Cdm0Submittal** class is used to store and represent data item submittal package or document to support the requirement of generating submittal delivery schedule for data requirement item. Submittal represents the data item submittal package or documentation that relates to the actual documents submitted.



Cdm0SubmittalRevision attribute

The **Cdm0SubmittalRevision** class is used to store and represent submittal revision data.

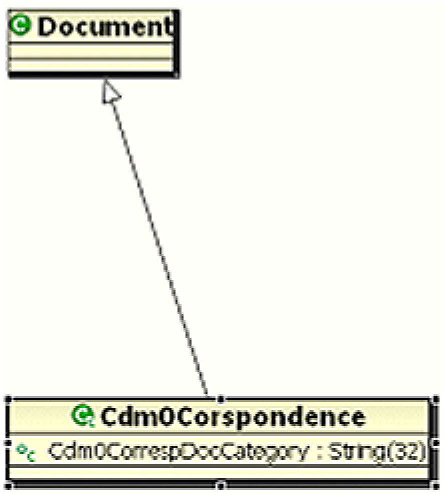

DocumentRevision
DocumentAuthor : String(128)
DocumentSubject : String(128)
DocumentTitle : String(128)
Cdm0SubmittalRevision
Cdm0Disposition : String(32)
Cdm0DispositionDate : Date
Cdm0DRLItemType : String(32)
Cdm0OfficePriResp : String(128)
Cdm0ReceivedDate : Date
Cdm0ResubmitComments : String(256)
Cdm0ResubmitDueDate : Date
Cdm0ResubmitRequired : Logical
Cdm0ShippingDocumentName : String(128)
Cdm0SubmitDueDate : Date
Cdm0SubmittalMaturity : String(32)

The following table lists the attributes of the **Cdm0SubmittalRevision** class.

Attribute name	Storage type	Description
Cdm0Disposition	**String(32)**	Specifies the unique identifier for **DD Form 250** prepared for the contract deliveries.
Cdm0ReceivedDate	**Date**	Specifies the date when an item is received from the supplier or a date when the item is sent to a customer.
Cdm0Resubmit DueDate	**Date**	Specifies the due date of the contract data item to the customer.
Cdm0Submit DueDate	**Date**	Specifies the due date of the contract data item to the customer.
Cdm0Submittal Maturity	**String(32)**	Specifies the data item submittal stage, for example, draft, or final.
Cdm0Disposition Date	**Date**	Specifies the disposition or the incorporation level of a change or submittal.
Cdm0OfficePriResp	**String(128)**	Specifies the person, group, or the role responsible for reviewing acknowledging the data item submittal.
Cdm0Resubmit Comments	**String(256)**	Specifies customer’s comments during resubmittal.
Cdm0Resubmit Required	**Logical**	Specifies whether the data item must be submitted again.
Cdm0DRLItemType	**String(32)**	Specifies whether the data item requirement is from a customer or a supplier.
Cdm0Shipping DocumentName	**String(128)**	Specifies the ID of **DD Form 250** that is prepared to support contract deliveries.

Cdm0Correspondence attribute

The **Cdm0Corspondence** class is used to store and represent correspondence data.



The following table lists the attributes of the **Cdm0Corspondence** class.

Attribute name	Storage type	Description
Cdm0CorrespDoc Category	**String(32)**	Specifies a category of similar technical documents, for example, assembly drawing, wire list, or material specification.

Cdm0CorrespondenceRevision attribute

The **Cdm0CorspondenceRevision** class is used to store and represent correspondence revision data.

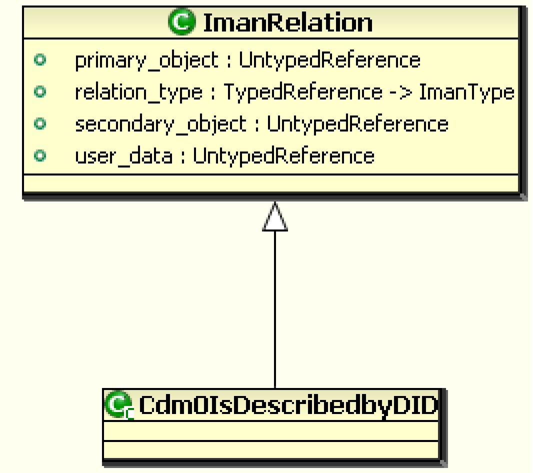
The following table lists the attributes of the **Cdm0CorspondenceRevision** class.

Attribute name	Storage type	Description
Cdm0Category	**String(32)**	Specifies correspondence category, for example, memo.
Cdm0Correspondence Direction	**String(32)**	Specifies whether it is an incoming or an outgoing correspondence.
Cdm0Priority	**String(32)**	Specifies the priority level for accomplishing the task.
Cdm0ReceivedorSent Date	**Date**	Specifies the date of correspondence from the supplier or the customer.
Cdm0ReceivedOrgName	**String(32)**	Specifies the name of the receiving organization.
Cdm0References	**String(32)**	Specifies the list of references.

Attribute name	Storage type	Description
Cdm0ReplyRequested Date	**Date**	Specifies the date of expected reply.
Cdm0ReplyRequired	**Logical**	Specifies whether a reply is required or not.
Cdm0ResponsibleParty	**String(32)**	Specifies the person or roles responsible for replying.

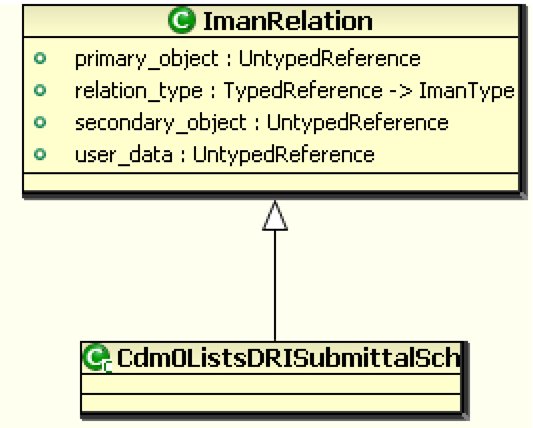
Cdm0IsDescribedbyDID attribute

The **Cdm0IsDescribedbyDID** class is used to represent relation between **Cdm0DIDRevision** and **Cdm0DataReqItem Revision** to support the requirement of creating the data requirement item revision and associating it with the contract revision.


ImanRelation
primary_object : UntypedReference
relation_type : TypedReference -> ImanType
secondary_object : UntypedReference
user_data : UntypedReference
Cdm0IsDescribedbyDID

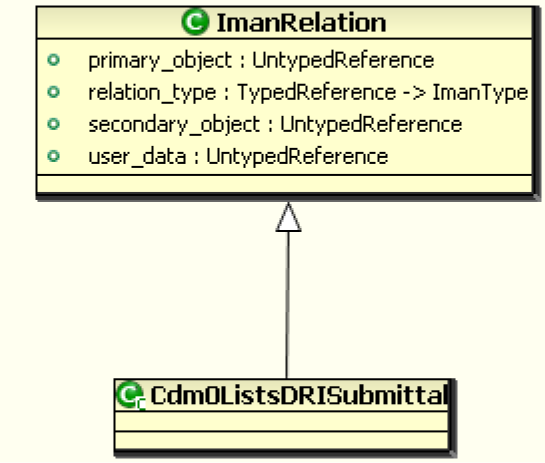
Cdm0ListsDRISubmittalSch attribute

The **Cdm0ListsDRISubmittalSch** class is used to represent relation between the data requirement item revision and the submittal delivery schedule.


ImanRelation
primary_object : UntypedReference
relation_type : TypedReference -> ImanType
secondary_object : UntypedReference
user_data : UntypedReference
Cdm0ListsDRISubmittalSch

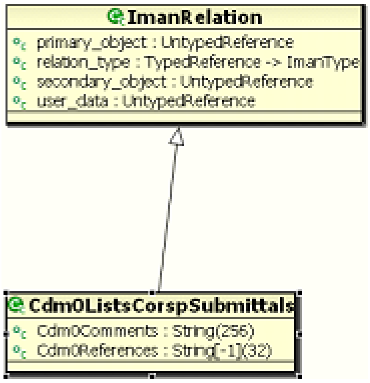
Cdm0ListsDRISubmittal attribute

The **Cdm0ListsDRISubmittalSch** class is used to represent relation between **Cdm0DataReqItemRevision** and submittal of the data item submittal package or documentation that relates to the actual documents submitted.


ImanRelation
primary_object : UntypedReference
relation_type : TypedReference -> ImanType
secondary_object : UntypedReference
user_data : UntypedReference
Cdm0ListsDRISubmittal

Cdm0ListsCorspSubmittals attribute

The **Cdm0ListsCorspSubmittals** class is used to represent relation between **Cdm0SubmittalRevision** and **Cdm0CorspondenceRevision**.

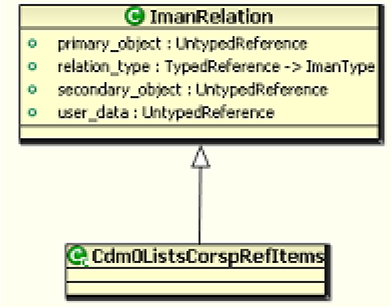

ImanRelation
primary_object : UntypedReference
relation_type : TypedReference -> ImanType
secondary_object : UntypedReference
user_data : UntypedReference
Cdm0ListsCorspSubmittals
Cdm0Comments : String(256)
Cdm0References : String[-1](32)

The following table lists the attributes of the **Cdm0ListsCorspSubmittals** class.

Attribute name	Storage type	Description
Cdm0Comments	**String(256)**	Specifies any comments concerning the item.
Cdm0References	**String [-1] (32)**	Specifies the list of references.

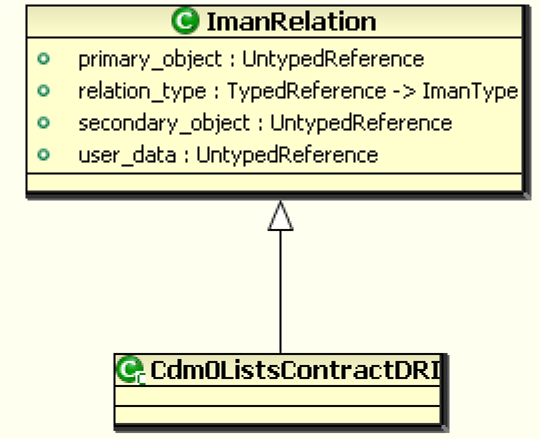
Cdm0ListsCorspRefItems attribute

The **Cdm0ListsCorspRefItems** class is used to represent relation between **Cdm0CorpondenceRevision** and **Cdm0ContractRevision**.


ImanRelation
primary_object : UntypedReference
relation_type : TypedReference -> ImanType
secondary_object : UntypedReference
user_data : UntypedReference
Cdm0ListsCorspRefItems

Cdm0ListsContractDRI attribute

The **Cdm0ListsContractDRI** class is used to represent relation between **Cdm0ContractRevision** and **Cdm0DataReqItemRevision**.


ImanRelation
primary_object : UntypedReference
relation_type : TypedReference -> ImanType
secondary_object : UntypedReference
user_data : UntypedReference
Cdm0ListsContractDRI

Stock material objects

The following table lists the stock material data model objects.

Symbol	Object	Description
	Stock Material (**SM0StockMaterial**)	Represents a stock material object.
	Stock Material Revision (**Stock Material Revision**)	Represents a stock material object revision.

The following table describes the stock material object relation types.

Relationship	Description
Made From (**SM0MadeFrom**)	Defines the relationship between an item revision and a stock material revision or an item revision and a part revision, which specifies that the item is made from a stock material or part.

Work package objects

The following table lists the work package data model objects.

Symbol	Object	Description
📁	Work Package (**Wpm0WorkPackage**)	Represents a work package object.
📁	Work Package Revision (**Wpm0WorkPackage Revision**)	Represents a work package object revision.

The following table describes the work package object relation types.

Relationship	Description
Static Contents (**Wpm0WorkPkg ContainsStatic**)	Defines the relationship between a work package revision and another object. This relationship specifies that if the contents of the work package change when they are outside the work package, the contents inside the work package do not change.
Dynamic Contents (**Wpm0WorkPkg ContainsDynamic**)	Defines the relationship between a work package revision and another object. This relationship specifies that if contents of the work package change when they are outside the work package, the contents inside the work package reflect the change.

Finish objects

The following table lists the finish data model objects.

Symbol	Object	Description
🧩	Finish (**Fsh0Finish**)	Represents a finish object.
🧩	Finish Revision (**Fsh0FinishRevision**)	Represents a finish object revision.
🧩	Finish Group (**Fsh0FinishGroup**)	Represents a finish group.
🧩	Finish Group Revision (**Fsh0FinishGroupRevision**)	Represents a finish group revision.

The following table describes the finish object relation types.

Relationship	Description
Finishes (**Fsh0FinishRel**)	Defines the relationship between a finish or finish group with an item revision or its subclass. This relation is a subclass of the trace link relation.

Data model for Aerospace and Defense relations

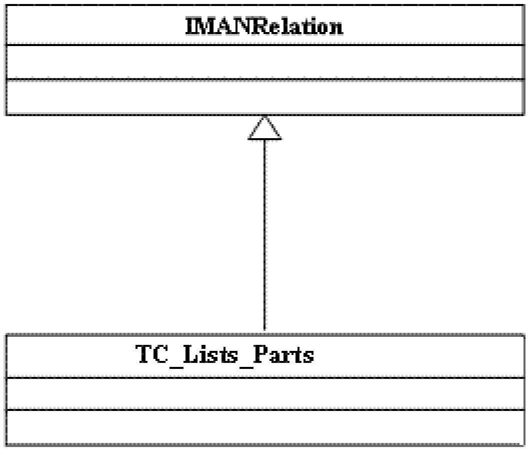
ADS_Lists_Parts

This relation represents the association between a technical document revision and an ADS part or ADSDesign item.

The following table describes the components of the **ADS_Lists_Parts** relation.

Relation name	**ADS_Lists_Parts**
Relation type	**IMAN Relation**
Association	**ADSTechDocument Revision** and **ADSPart** or **ADSDesign**
Primary object	**ADSTechDocument Revision**
Secondary object	**ADSPart** or **ADSDesign**

The following figure illustrates the **ADS_Lists_Parts** relation data model.



- Multiple **ADSPart** and **ADSDesign** items can be associated with a given technical document using this relation.
- When a primary object is revised, the association is carried forward to a new revision of the technical document.

The following table describes the attributes of the **ADS_Lists_Parts** relation.

Attribute name	Storage Type	Description
adsdrawing_find_number	**String [32]**	Specifies the numeric identifier, assigned to a part, that is used to identify the part on the face of the installation drawing instead of the actual part number.
ads0notenumber	**String [128]**	List of flag notes on the parts list that are referenced by the part.
adsdrawing_zone	**String [32]**	Specifies the drawing sheets and/or zones where a part's number or find number is displayed. The letter designates the vertical coordinate and the number designates the horizontal coordinate.
adsremarks	**String [32]**	Textual optional information entered by the user.

ADS_Lists_PartRevisions

This relation represents the association between a technical document revision and ADS part revisions or ADSDesign revisions.

ADS_Lists_PartRevisions associates all **ADSPart** and **ADSDesign** objects created on a given technical document with the technical document revision.

Using this relation:

- Multiple ADS part revisions or ADS design revisions can be associated with a given ADS technical document revision.
- When a primary object is revised, the association is carried forward to a new revision of the technical document.
- When a primary object has an immature revision, the associated secondary object can be revised independently and the revision is associated to the immature primary object revision.
- When a primary object has an immature revision, the primary object must be revised to revise the secondary object.

The following table describes the attributes of the **ADS_Lists_PartRevisions** relation.

Attribute name	Storage Type	Description
adsdrawing_find_number	**String [32]**	Specifies the numeric identifier, assigned to a part, that is used to identify the part on the face of the installation drawing instead of the actual part number.
ads0notenumber	**String [128]**	List of flag notes on the parts list that are referenced by the part.
adsdrawing_zone	**String [32]**	Specifies the drawing sheets and/or zones where a part's number or find number is displayed. The letter designates the vertical coordinate and the number designates the horizontal coordinate.
adsremarks	**String [32]**	Textual optional information entered by the user.

ADS_Lists_DrawingRevisions

This relation represents the association between a technical document revision and an ADS drawing revision.

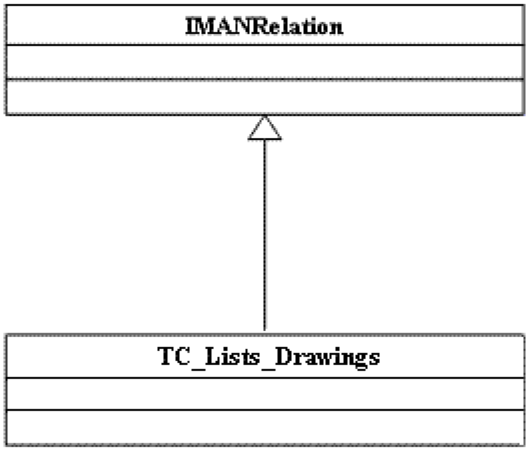
The **TC_Lists_DrawingRevisions** relation helps in associating all the **ADSDrawing** items created for a given technical document with the technical document revision.

The following table describes the components of the **TC_Lists_Drawings** relation.

Relation name	**TC_Lists_Drawings**
Relation type	**IMAN Relation**

Association	**ADSTechDocument Revision** and**ADSDrawing Revision**
Primary object	**ADSTechDocument Revision**
Secondary object	**ADSDrawing Revision**

The following figure displays the **TC_Lists_DrawingRevision** relation data model.



- Multiple **ADSDrawing Revision** objects can be associated to a given technical document revision with this relation.
- When a primary object is revised, the association is carried forward to a new revision of the technical document.
- When the primary object has an immature revision, the associated secondary object can be revised independently and the revision is associated to the immature primary object revision.
- When the primary object does not have any immature revision, the primary object needs to be revised to revise the secondary object.

The following table describes the attributes of the **TC_Lists_DrawingRevisions** object.

Attribute name	Storage Type	Description
adsdrawing_find_number	**String [32]**	Specifies the numeric identifier, assigned to a part, that is used to identify the part on the face of the installation drawing instead of the actual part number.
ads0notenumber	**String [128]**	List of flag notes on the parts list that are referenced by the part.
adsdrawing_zone	**String [32]**	Specifies the drawing sheets and/or zones where a part's number or find number is displayed. The letter

Attribute name	Storage Type	Description
		designates the vertical coordinate and the number designates the horizontal coordinate.
adsremarks	**String [32]**	Textual optional information entered by the user.

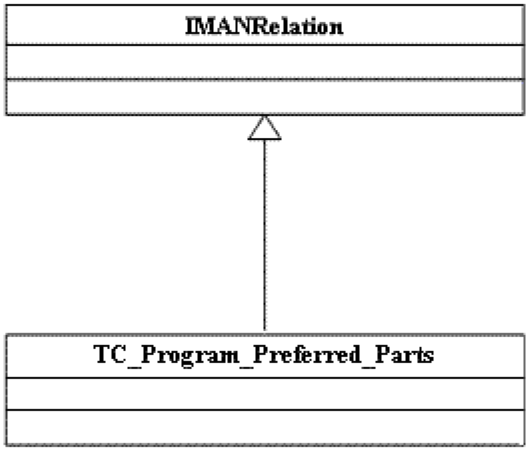
TC_Program_Preferred_Parts

This relation represents the association between a standard part and a program. Multiple standard parts can be associated with a given program using this relation.

The following table describes the components of the **TC_Program_Preferred_Parts** relation.

Relation name	**TC_Program_Preferred_Parts**
Relation type	**IMAN Relation**
Association	**CommercialPart Item** and **TCProject**
Primary object	**TCProject**
Secondary object	**CommercialPart Item**

The following figure displays the data model for the **TC_Program_Preferred_Parts** relation.



The following table describes the attributes of the **TC_Program_Preferred_Parts** relation.

Attribute	Description
Limited Use	Specifies whether the part is in use. The valid values are **Yes** or **No**.
Limitation	Specifies the limitation on the part.

Fnd0LocationForUser

This relation represents the association between a company location and a user.

Using this relation you can specify the nature of association between the company location and the user. The association can be either **true company affiliation** or **design authority affiliation**.

The following table describes the attribute of the **Fnd0LocationForUser** relation.

Attribute	Storage Type	Description
fnd0Location Affiliation	**String [32]**	Specifies the nature of association between the company location and the user. The valid values are **true company affiliation** or **design authority affiliation**.

Fnd0LocationForGroup

This relation represents the association between a company location and a user.

Using this relation you can specify the nature of association between the company location and the group. The association can be either **true company affiliation** or **design authority affiliation**.

The following table describes the attribute of the **Fnd0LocationForGroup** relation.

Attribute	Storage Type	Description
fnd0Location Affiliation	**String [32]**	Specifies the nature of association between the company location and the group. The valid values are **true company affiliation** or **design authority affiliation**.

Lists of values for Aerospace and Defense

Lists of values for Aerospace and Defense

Lists of values (LOVs) are used to ensure consistent data entry in Teamcenter. The list entries are created either by referencing existing data or by entering customized site data. After it is created, the LOV is implemented throughout the interface by attaching the LOV to one or more properties. Creating LOVs and implementing them throughout Teamcenter can greatly improve productivity at your site and help prevent incorrect user entries.

The Aerospace and Defense solution provides LOVs that are specific to the Aerospace and Defense industry. Use these LOVs to configure and modify the behavior of the template.

Adc0Change Class

Description	Designates the government's or customer's classification to a change. For example, **I** or **II**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeNoticeRevision.Adc0change_class** **Adc0ChangeRqstRevision.Adc0change_class** **Adc0DevRqstRevision.Adc0change_class**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**I** **II**
Valid values	Accepts string as a value. It must be a valid classification.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Change Category

Description	Designates the specific category of change within the change classification.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeRqstRevision.Adc0change_category** **Adc0ChangeNoticeRevision.Adc0category** **Adc0DevRqstRevision.Adc0change_category**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**1** **2** **A**

	B **C** **D** **E** **M**
Valid values	Accepts string as a value.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0CN Types

Description	Specifies acronyms for the types of change documentation used on a program for a change notice.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0ChangeNotice.Adc0change_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**ADCN** **AMR** **DCN** **DCR** **DRN** **ICN** **PCD** **PMD** **REDLINE** **RR** **VAR**
Valid values	Accepts string as a value. It must be a valid acronym that specifies the type of change documentation.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0CR Types

Description	Specifies the change request types. For example, development change or engineering change process.
Type	**ListOfValuesString**

Usage	Exhaustive.
Business object property	**Adc0ChangeRqst.Adc0change_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**CA** **CRBD** **DEV** **ECP** **IPCD** **RFD** **RFW**
Valid values	Accepts string as a value. It must be a valid change request type.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0DR Types

Description	Specifies acronyms for the types of change documentation used on a program for a deviation request, for example, **RFD** or **RFW**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Cm0DevRqst.Cm0change_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**RFD** Request for Deviation **RFW** Request for Waiver

Valid values	Accepts string as a value. It must be a valid acronym that specifies the type of change documentation.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Production Systems

Description	Displays the list of systems or subsystems for an item. A product system denotes a group of parts that perform a common function. For example, **Electrical**, **Environmental Control**, and **Weapons**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0DevRqstRevision.Adc0product_systems** **Adc0ChangeRqstRevision.Adc0product_systems** **Adc0ChangeNoticeRevision.Adc0product_systems**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**Electrical** **Environmental Control** **Hydraulic** **Fuel** **Weapons**
Valid values	Accepts string as a value. It must be a valid system or subsystem.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Quotation Type

Description	Specifies the quotation type for the change impact, for example, **Supplier ROM**, **Buyer Estimate**, or **Supplier Firm**.
Type	**ListOfValuesString**
Usage	Exhaustive.

Business object property	**Adc0ChangeRqstRevision.Adc0quotation_type** **Adc0DevRqstRevision.Adc0quotation_type**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**NTE** **ROM**
Valid values	Accepts string as a value. It must be a valid quotation type.
Notes	This LOV is included in the **adschangemanagement** template.

Adc0Task Category

Description	Specifies the various task categories, for example, **Agenda Event**, **Planning**, and **QA**.
Type	**ListOfValuesString**
Usage	Exhaustive.
Business object property	**Adc0Task.Adc0task_category**
Business object condition	**AnDCMTrainingProgramCondition**
Default values	**Agenda Event** **Engineering** **ILS** **Major Subcontr** **Manufacturing** **Non-Std Event** **Part** **Planning** **QA** **Subcontract**
Valid values	Accepts string as a value. It must be a valid task category.
Notes	This LOV is included in the **adschangemanagement** template.

Ads0CreateStandardNoteAuthority

Description	List of authorized roles for creation of **StandardNote**.
Type	**ListofValuesString**
Usage	Suggestive.

Business object property	None.
Business object condition	None.
Valid Values	Any valid Teamcenter role.
Default Values	*
Notes	Specifies the roles that are authorized to create or revise standard note objects. An asterisk (*****) indicates that any Teamcenter user can create or revise standard note objects. Apart from the user roles that are mentioned as values in the LOV, users with DBA permission are authorized to create, revise, update, and delete standard notes.

ADS Assembly Types

Description	List of ADS assembly types.
Type	**ListofValuesString**
Usage	Exhaustive.
Business object property	**ADSDesign.adsassembly_type**
Business object condition	**isTrue**
Valid Values	Any valid assembly type.
Default Values	**Separable** **Inseparable**
Notes	An inseparable assembly part cannot be disassembled without causing physical harm to at least one of the constituent parts in the assembly.

ADSComponentAuthority

Description	List of **ADSComponent** authorities.
Type	**ListofValuesString**
Usage	Exhaustive.
Business object property	None.

Business object condition	None.
Valid Values	Any valid Teamcenter role.
Default Values	*****
Notes	Specifies the roles that are authorized to create or revise ADS parts or ADS design objects. An asterisk (*****) indicates that any Teamcenter user can create or revise ADS parts or ADS design objects.

ADS Design Levels

Description	List of ADS Design levels.
Type	**ListofValuesString**
Usage	Exhaustive.
Business object property	**ADSTechDocument Revision.adsdesign_level**
Business object condition	**isTrue**
Valid Values	Any valid design level.
Default Values	**Conceptual** **Prototype** **Production**
Notes	None.

ADSDocumentCentricPrograms

Description	List of ADS document-centric programs.
Type	**ListofValuesString**
Usage	Exhaustive.
Business object property	None.
Business object condition	None.
Valid Values	Any valid Teamcenter program.

Default Values	**Training**
Notes	It specifies the programs that are document centric.

ADSDrawingAuthority

Description	List of **ADSDrawing** authorities.
Type	**ListofValuesString**
Usage	Exhaustive.
Business object property	None.
Business object condition	None.
Valid Values	Any valid Teamcenter role.
Default Values	*
Notes	Specifies the roles that are authorized to create or revise ADS drawing objects. An asterisk (*) indicates that any Teamcenter user can create or revise drawing objects.

ADSTechDocAuthority

Description	List of **ADSTechDoc** authorities.
Type	**ListofValuesString**
Usage	Exhaustive.
Business object property	None.
Business object condition	None.
Valid Values	Any valid Teamcenter role.
Default Values	*
Notes	Specifies the roles that are authorized to create or revise ADS technical document objects. An asterisk (*) indicates that any Teamcenter user can create or revise technical document objects.

TCProgramPreferredTypes

Description	Specifies the part types that are treated as preferred part types while creating assemblies.
Valid Values	Accepts string as a value. It must be a valid Teamcenter part type.
Default Values	**CommercialPart**
Template	**foundation** template.

TCProgramsUsingPreferredTypes

Description	Specifies the programs that use preferred parts while creating assemblies of these parts.
Valid Values	Accepts string as a value. It must be a valid Teamcenter program.
Default Values	
Template	**foundation** template.

TcRevisionSkipLetters

Description	Specifies the characters to be excluded from the revision naming rule.
Valid Values	Alphabets.
Default Values	**I, O, Q, S, X, Z**
Template	**foundation** template.