TEAMCENTER

Systems Engineering Fundamentals

Teamcenter 2412

SIEMENS

Unpublished work. © 2025 Siemens

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

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

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

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

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

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

About Siemens Digital Industries Software

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

Support Center: **support.sw.siemens.com**

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

Contents

© 2025 Siemens

© 2025 Siemens

1. Getting started

Overview of systems engineering

The systems engineering discipline is a method of managing the design of complex, multi-domain engineering products throughout the life cycle of a product development project.

It allows you to identify the necessary information to make decisions about:

- When product development begins.
- What requirements are defined.
- When architecture decisions are made.
- What interfaces are defined.
- How globally optimized decisions are made.
- What requirements, assemblies, and components can be reused.

Teamcenter Systems Engineering allows you to:

- Develop requirement structures and create functional and logical architectures for the product design.
- Define interfaces between design elements in functional and logical architectures, and understand the relationships between them.
- Create traceability between requirements and functional, logical, and physical architectures.
- Support formal Systems Engineering processes and standards such as IEEE 1220, ISO 15288, EIA 632, and INCOSE.
- Support ad hoc systems engineering and product development activities that do not conform to a formal standard.

Before you begin

Prerequisites	You need a basic understanding of the systems engineering discipline to use Systems Engineering. You do not need special permissions to use Systems Engineering. Standard Teamcenter user and administrator accounts are adequate. You need valid feature licenses to work with Systems Engineering. Certain menu commands are disabled if these licenses are

© 2025 Siemens

not installed. Contact your Siemens Digital Industries Software representative for more information.

To use Teamcenter Extensions for Microsoft Office, the following must be installed on your computer:

- Teamcenter Extensions for Microsoft Office
- Microsoft .NET Framework

> Note:
>
> For certified versions of Microsoft products, see the hardware and software certifications matrix.

Enable Systems Engineering

Systems Engineering is an optional module installed as part of the standard Teamcenter core installation procedure. It does not need to be enabled before you use it.

During installation in Teamcenter Environment Manager, select the following check boxes under **Extensions→Systems Engineering and Requirements Management** in the **Features** panel:

- **Requirements Management**
- **Systems Engineering**

If you have trouble gaining access to Systems Engineering, consult your Teamcenter system administrator. It may be a licensing issue.

> Note:
>
> You can log on to Teamcenter only once. If you try to log on to more than one workstation at a time, an error message appears.

Configure Systems Engineering

Before using Systems Engineering, you must:

- Configure the necessary business rules.
- Set the appropriate preferences.

You can optionally configure some Systems Engineering features by setting user preferences.

- The **MS Word** view
- Keywords for parsing Word import documents

© 2025 Siemens

- Baselines
- Property columns
- Trace links

> Note:
>
> Trace link features are not displayed until you set the corresponding preferences.

- Display of defining and complying objects as properties
- The **Summary** and **Properties** panes

Start Systems Engineering

Do one of the following:

- Select **Systems Engineering** in the Teamcenter navigation pane.

> Note:
>
> You may need to enter your user name and password to log on.

The Systems Engineering main window is displayed. You can begin using Systems Engineering by doing any of the following:

- Choose **File→Import spec** to display the Import Spec wizard.
- Double-click a requirement specification, requirement, paragraph, function, or logical block to display a new tab containing the item.
- Click the **Most recently used structures** button on the toolbar to select an object from the list.

- In another Teamcenter application, such as My Teamcenter, Structure Manager, or Multi-Structure Manager:

1. Navigate to a requirement specification, a requirement, a paragraph, a function, or a logical block that you want to send to Systems Engineering.
2. Right-click the object and choose **Send To→Systems Engineering**.

© 2025 Siemens

Systems Engineering opens with the object displayed in a new tab.

> Tip:
>
> When you start a Systems Engineering session, the view you had open in the previous session is restored. The object you previously selected is highlighted, and the column configuration and revision rule are also restored. The previous session's sorting sequence is not retained, and the level expansion is retained only for the child object selected when the previous session ended. If multiple structure views were previously open, content is restored only for the last active view. If the client cache is cleared, the previous session settings are not restored.

Basic concepts for Systems Engineering

Designing a product

Designing a product is comprised of three main activities:

- Gathering and communicating the requirements
- Analyzing the requirements and synthesizing solutions
- Creating successively more precise definitions of the product functional, logical, and physical decompositions

These activities overlap, and an iteration of one activity probably affects the other.

Your starting point may depend on whether you are designing a new system or developing an existing one.

Projects that inadequately communicate requirements to developers are prone to run over budget and behind schedule, with late-cycle changes and extensive integration efforts. In turn, these conditions can lead to functional deviations and inferior quality, and ultimately to a product that the customer may reject.

Developing a new product

You typically begin the new product development process by capturing input requirements from customer specifications, standards, lessons learned, telephone conversations, user evaluation sessions, and many other sources. When you analyze these requirements, you may start to derive new requirements; for example, a functional requirement may derive design and manufacturing

© 2025 Siemens

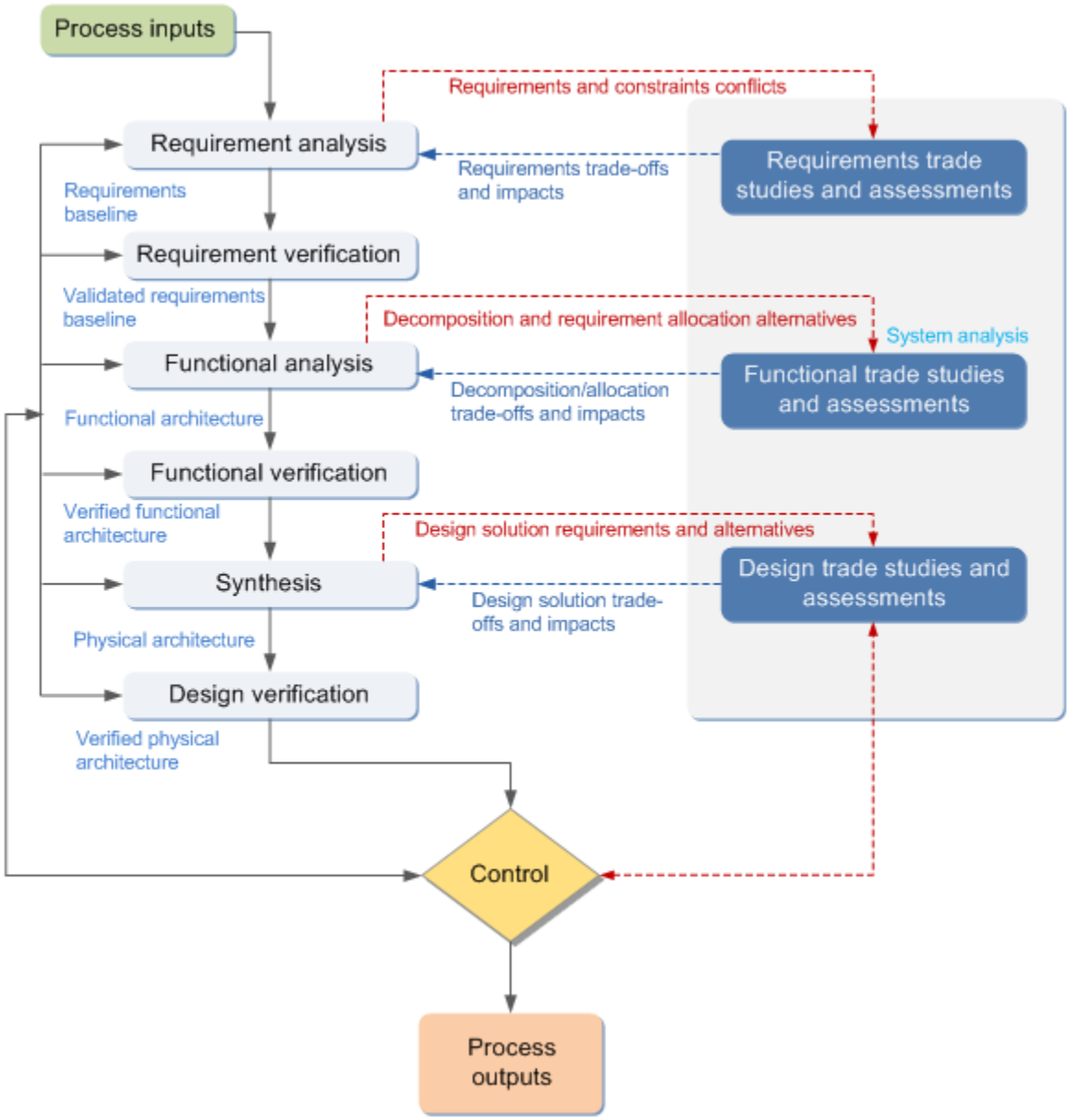
requirements. This can lead to an explosion of requirements in large systems—a single requirement may have thousands of derived requirements.

As you develop these requirements, you draft a list of features or functions that need to be built into the product. You may also discuss alternative ways of satisfying the requirements. As this list matures, you can create an initial functional model (decomposition or breakdown) of the system, followed by a logical model. You can then allocate functions to logical blocks. Later, when these models are sufficiently mature, you can create a physical model.

You should also consider other views of the system to ensure you can test it, manufacture it, support it in the field, and so on. You create relationships between these different views using *trace links*. A trace link creates a directional relationship between two objects. Trace links allow you to follow a path between an original requirement and the way it is satisfied in the product. For example, you can start with the requirement, and then identify how it is implemented in the product structure and how it is assembled in the manufacturing process plan.

The IEEE 1220 representation of the systems engineering process is shown in the following figure. Note that this process ends at the design verification stage, when design documents are released to design engineering. Depending on your working environment, you may consider the EIA 632 standard, which includes product validation and release process. Alternatively, ISO 15288 describes fully inclusive system life cycle processes.

© 2025 Siemens



Improving an existing product

If you are refining or improving an existing product, you may begin by reviewing the existing solutions or reference architectures to identify areas for improvement, for example, to make a car more economical. You may collect additional improvement requirements from customer feedback, problem reports, service data, and so on. You may modify the original requirements or create a different requirements structure for the development effort.

Functional, logical, and physical models

Teamcenter supports three types of structure:

© 2025 Siemens

- Functional

  The functional model comprises multiple building blocks that describe what the product or system does, and breaks down the top-level functions into lower levels. For example, a car transports occupants, transports luggage, protects occupants, protects pedestrians, and performs many other functions. You can break down protecting occupants into avoiding collisions and providing impact protection. The model is domain independent and is therefore also independent of the solution chosen. The building blocks map to parts or assemblies in the physical model.

- Logical

  The logical model describes the solution elements in abstract terms and how they interact with each other. The solution elements are then detailed out in the physical model. The logical model is the point at which you enter domain-specific design. For example, for an electromechanical product, you can create a logical model that describes the product or system in terms of input ports, output ports, interfaces, and connections.

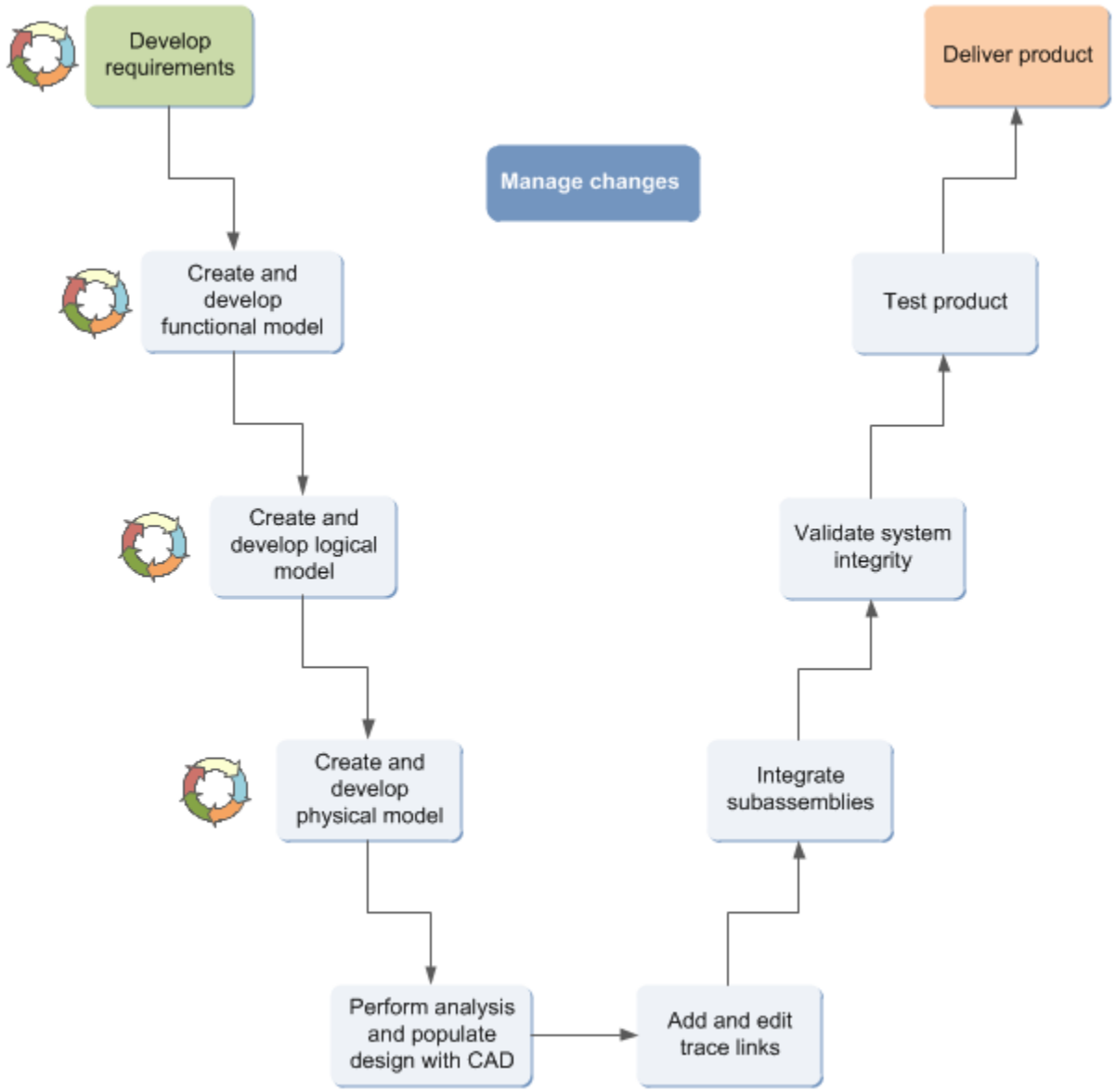
- Physical

  You can map the physical model to the functional and logical models. You also map it to the physical product structure, which you then populate with CAD designs, parts, and design documents. The physical decomposition precedes the creation of CAD files, software codes, and other components; it is the final step before you hand off the emerging design to the engineers responsible for the detailed design.

Basic tasks using Systems Engineering

Use Systems Engineering to perform the following tasks in the systems design process:

© 2025 Siemens



- Develop and manage requirement structures.

  Perform these tasks with the **Requirements** view of the **Systems Engineering** perspective.

- Create and maintain functional models structures.

  Perform these tasks with **Functional** view of the **Systems Engineering** perspective and the Microsoft Office Visio diagramming tool.

- Create and maintain logical model structures.

  Perform these tasks in Structure Manager (Embedded Software Solutions option) and Multi-Structure Manager.

© 2025 Siemens

- Relate requirements and structure components with trace links.

  Perform these tasks in the **Requirements** and other views of the **Systems Engineering** perspective, and in the **Structure Manager** perspective.

- Create and maintain physical model structures.

  Perform these tasks in Structure Manager, Multi-Structure Manager, Change Manager, and domain-specific CAD tools such as NX.

- Manage changes in Change Manager, including submitting change requests, performing impact analysis, and processing and executing changes.

Users of Systems Engineering

Information in this guide is intended for the following categories of users:

- Existing Teamcenter users who want to implement Systems Engineering and are new to the systems engineering discipline.
- Any customer who is new to Teamcenter and requires an overview of this functionality.

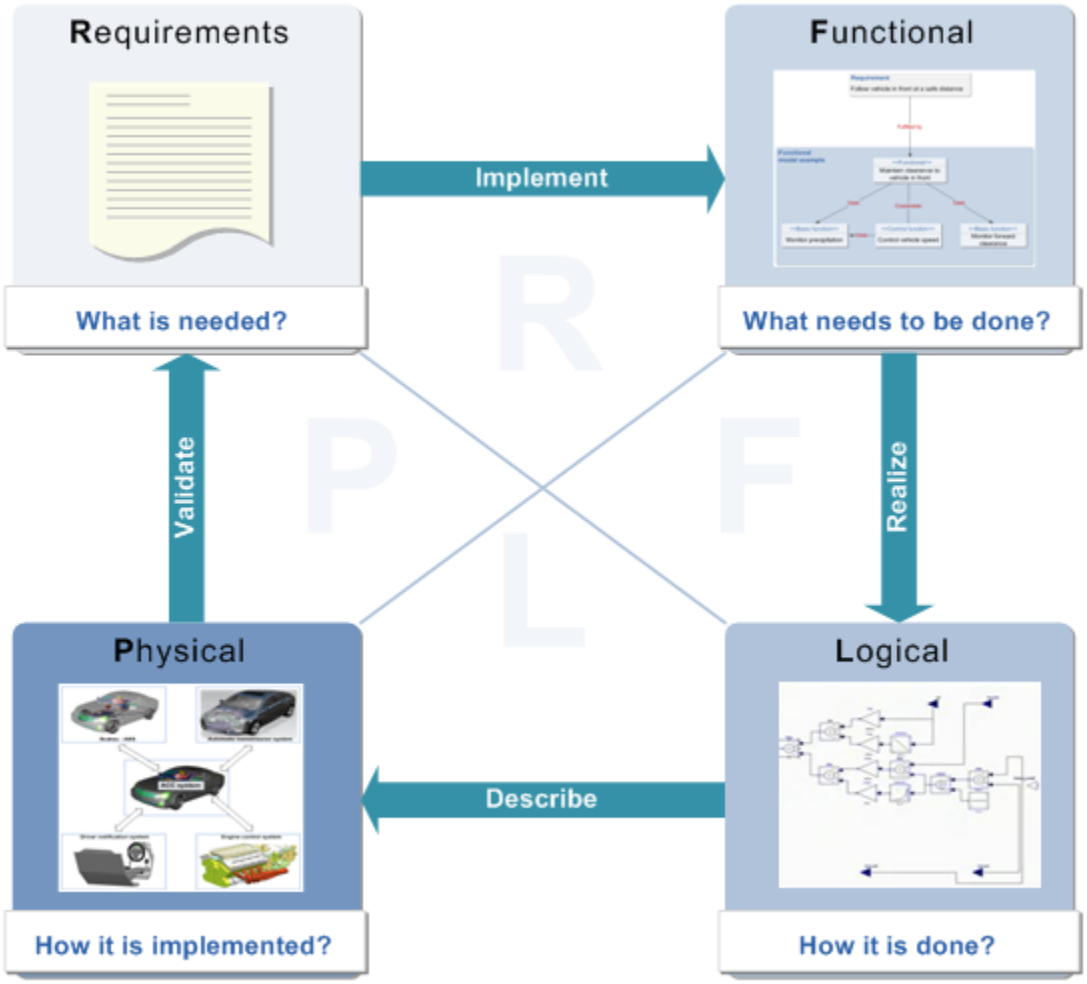
This guide contains information suited to all levels of expertise, including:

- Business analysts (end users) who define and manage models and requirements
- Product managers who create and review requirements
- Application administrators (power users) who manage projects, templates, and import and export data
- System administrators who configure Systems Engineering
- Systems engineers (electrical or software) who develop models

Overall Systems Engineering workflow

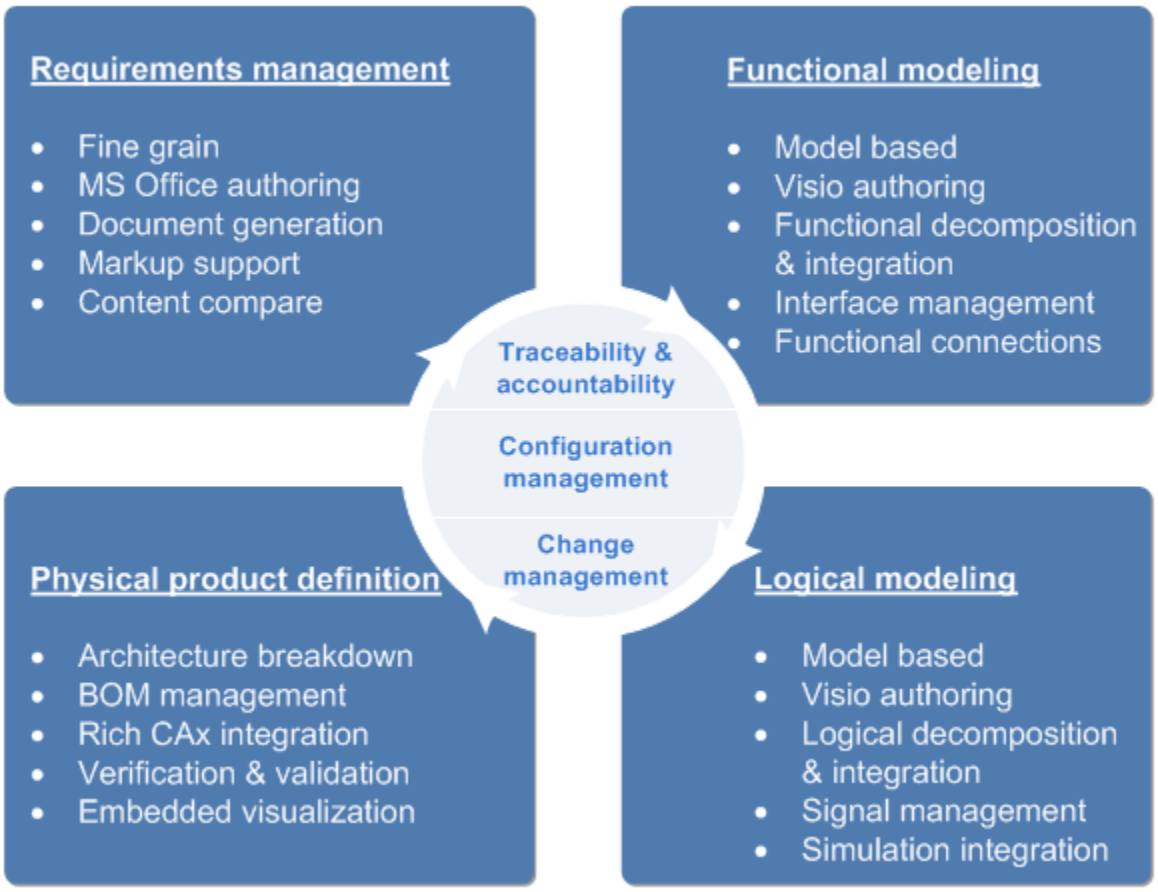
The following diagram shows a typical Systems Engineering workflow, which is referred to as the requirements, functional, logical, and physical design (RFLP) process. Note that the process is iterative and may be repeated multiple times during the design or development of a product.

© 2025 Siemens



The following diagram shows details of the tasks that are performed at each stage of this process.

© 2025 Siemens



Related documentation

For more information about Systems Engineering and associated features, refer to the following documentation:

- *Systems Engineering — Deployment and Rich Client Usage*

  Provides online help for the Systems Engineering perspective, including information about the Teamcenter integrations with Microsoft Office Word, Excel, and Visio.

- *Basic Classification — Deployment and Administration*

  Provides information about setting up a data dictionary.

- *Structure Management on Rich Client — Usage* and *Systems Engineering — Deployment and Rich Client Usage*

  Provide information about searching in and using a data dictionary.

- *Change Management — Deployment and Rich Client Usage* and *Workflow Designer on Rich Client*

© 2025 Siemens

Provide information about how to manage changes, reviews and approvals for model data.

You can also refer to the following standards for additional information:

- IEEE 1220-2005 (IEEE standard for application and management of the Systems Engineering process).
- EIA 632 (processes for engineering a system).
- ISO 15288 (system life cycle processes) and ancillary standards (ISO/IEC 19760).
- MIL-STD-499B for U.S. Department of Defense (DOD) programs. This standard was never formally issued but is used as the basis for many other standards including IEEE-1220.

Note:

Systems Engineering supports all commercial, international, and government systems engineering processes.

It also supports CMMI staged and continuous representations.

© 2025 Siemens

2. Working with requirements

Working with requirements

Requirements describe the product or system that the customer will buy. They communicate the customer's specifications to the various disciplines involved in the development process. To ensure that the finished product or system meets those specifications, developers follow the requirements throughout the development process. When the product conforms to all requirements, it is ready for delivery, and more importantly, it has the functions, attributes, and quality that the customer demands.

Two principles are vital:

- Requirements must be identified and developed at the project's inception. This is necessary so that any problems are revealed and understood before the actual product development begins.
- Requirements must be associated with the product design using trace links. The trace links must be maintained through all successive stages, including development, testing, and changes to drawings, parts, and assemblies.

Incorporating both principles, the Systems Engineering features of Teamcenter provide for:

- Development of requirements in the initial stage of product development.
- Early association of requirements with the functional, logical, and physical models and their individual components.
- Ongoing maintenance of requirements, each as a separate item with specific properties.

Systems Engineering extends customer involvement to all phases of the product development process. The customer can actively influence the design from requirements development and association to product testing and acceptance.

Objects you work with

- Building block

  Building blocks graphically illustrate the hierarchical relationships of elements in a product or system. For example, such hierarchies may reflect design elements in a product, tasks in a work breakdown structure, or job functions in an organizational chart. You can create diagrams with Microsoft Office Visio and attach them to building blocks. You create and organize building blocks in folders. In the same way you can structure requirements within a folder, you can organize building blocks in multiple levels of parents, children, and siblings; alternatively, they may all reside at the top level.

- Complying object

© 2025 Siemens

An object that partially or completely fulfills a condition specified by a defining object. You can follow trace links from complying objects through the hierarchy to one or more originating sources.

- Connection

  A connection defines the association between two or more terminals or nodes in a physical model.

- Defining object

  An object that specifies a condition that a product or component must fulfill. You can follow trace links from defining objects down through the hierarchy to the lowest level requirement that meets the original condition.

- Derived requirement

  A requirement that is translated into parameters suitable for lower level analysis and design. The inputs to the derivation process are higher level requirements and the outputs are new requirements.

- Diagram

  A diagram represents a functional model, logical model, or a subset of one of those models. You can create and edit diagrams with Visio. Each Visio diagram is interactively synchronized with the database and is associated with a custom stencil that contains shapes representing the requirements objects. This allows you to create, modify, and delete objects in the database by adding, modifying, and deleting the corresponding shapes in the diagram. Diagrams are typically attached to building blocks, but can be attached also to folders, requirements, and groups.

- Folder

  In a project hierarchy, the primary level is represented by folders. Folders contain requirements, building blocks, groups, and other folders in the requirements hierarchy.

- Note

  Use notes to record information that relates only to certain objects, rather than to all objects of a type. For example, a note may convey the rationale for a decision, the minutes of a meeting, or an informal discussion about a particular requirement. You can attach notes to folders, requirements, building blocks, and groups. You create and edit the content of notes with Microsoft Office Word.

- Project

  A project represents the highest level of organization. Each project prescribes a boundary for user access and for customization of object types and properties. It also defines a hierarchy in which the other types of objects reside.

- Requirement

© 2025 Siemens

Requirements describe the product that the customer will buy. They communicate the customer's specifications to the various disciplines involved in the product's development. You create and organize requirements in folders. Within each folder, you can define multiple levels of organization to allow flexibility in structuring requirements. For example, all requirements in a folder can reside at the top level. Alternatively, you can organize them in a hierarchy of parent, child, and sibling requirements. Each requirement is a separately managed object, with specific properties and access control. You create and edit the content of requirements in Word.

- Spreadsheet

  Spreadsheets allow you to perform complex analysis and manipulation of requirements data, for example, budgets. You can attach spreadsheets to folders, requirements, building blocks, and groups. Equations and other content can be edited in Microsoft Office Excel.

- Trace link

  Trace links help ensure the components of the physical and logical models have supporting requirements. Likewise, they help ensure that each requirement is satisfied in the physical and logical models. A trace link establishes a directional relationship between two objects. Each trace link indicates which object precedes, or defines, the other in the relationship. The defining and complying objects can reside within the same project, or can reside in different projects. You can create trace links between folders, requirements, building blocks, and groups.

Example: workflow when a requirement changes

Systems Engineering can be used to support many working practices and workflows when creating, reviewing, and managing requirements.

In the following example, changes to an existing design are identified, based on a new requirement. In this scenario, competitive analysis has identified a possible problem with the braking system of an existing car.

1. The vehicle development engineer confirms the issue and identifies that the dry pavement braking distance should be reduced from 90 feet to 70 feet. This change of requirement affects the brake behavior building block of the functional model.

2. The vehicle development engineer locates trace links from the brake behavior building block of the functional model to the braking system, collision avoidance system and the traction control system of the physical model. All these systems are potentially affected by the proposed change.

3. The brake system engineer simulates the effect of a higher brake pressure, plotting speed versus time, and then updates the behavior analysis data in the functional model.

4. The brake system engineer tries alternate part templates, checks the clearance database results, and identifies a potential clearance issue. Likewise, thermal analysis results indicate potential brake disk overheating issues.

© 2025 Siemens

5. The brake system engineer logs both potential problems as issues in Teamcenter, attaching a temperature/time chart as supporting data.

6. The brake system engineer performs a search in Teamcenter and identifies several alternative part templates that may resolve the issues.

7. Teamcenter performs a *what if* analysis of the possible alternatives and notifies each affected group using the project dashboard.

8. Each affected group reviews the available data and responds appropriately. For example, the cost management group must review and approve any cost change.

9. Affected groups process the change notice and update data for their area. For example:

    - The vehicle dynamics group reruns validations.
    - The traction control software engineer updates the calibration tables.
    - The wiring harness group reroutes the wiring harness to the braking system.

10. The program manager reviews the dashboard and the actions taken, and then approves the change.

The detailed process may be represented by the steps in the following flow chart.

© 2025 Siemens

Creating requirements

The first step in requirements identification is to gather information for source requirements. Such information may be gathered from telephone conversations, meetings, regulatory agencies, standards organizations, and external documents.

The next step is to analyze that information, looking for issues, ideas, and keywords expressed by the customer. By those guidelines, irrelevant material can be separated and discarded. The remaining information becomes the content for the source requirements.

You can create and edit requirement content directly in the **Systems Engineering** perspective or author it in Microsoft Office Word. (If you use the stand-alone version of Word, rather than integrated Word, you must import the requirement content into Teamcenter.) The following example shows a very simplified requirements hierarchy.

© 2025 Siemens

000075/A;1 - Speed spec (ReqSpec)
Req001144/A;1 - Acceleration (Requirement)
Req001145/A;1 - Speed range (Requirement)

Requirements can be structured in a hierarchy in which the levels convey relationships among the requirements. Each individual requirement is identified by a unique number defined in the **bl_hierarchical_number** property for the requirement object.

Within a hierarchy, a requirement may occupy the highest level, as a sibling of all other requirements at level 1. Alternatively, a requirement may occupy a lower level, as a child of the requirement at the next higher level. In turn, a child requirement may also be a parent of children at even lower levels.

Although a traditional requirements specification can include hundreds or thousands of requirements, it is typically managed as a single document. In Teamcenter, however, each requirement is managed as a separate object, with specific properties and access control.

Also in the identification process, requirements may be organized by project according to their intended implementation. This organization partitions the requirements around designs, so that records can be generated for reference in comparing changes with previous structures.

Requirement content consists of familiar elements such as text paragraphs and lists, hyperlinks, tables, graphics, equations, and special characters.

Working with hierarchical requirements is similar to working with numbered paragraphs in Word's outline view. Each requirement has a paragraph number that corresponds to the requirement's level in the hierarchy. The current level and paragraph number can be changed by promoting or demoting the requirement or by manually changing the requirement's number property.

Creating derived requirements

You can derive additional (derived) requirements from the original source requirements. This process involves creating more detailed requirements that can be traced back to the originating sources. Trace links, from source requirements to derived requirements to other derived requirements, establish the order of precedence among requirements.

A derived requirement is essentially a child requirement of an existing requirement. You can create a derived requirement in the context of an existing requirement.

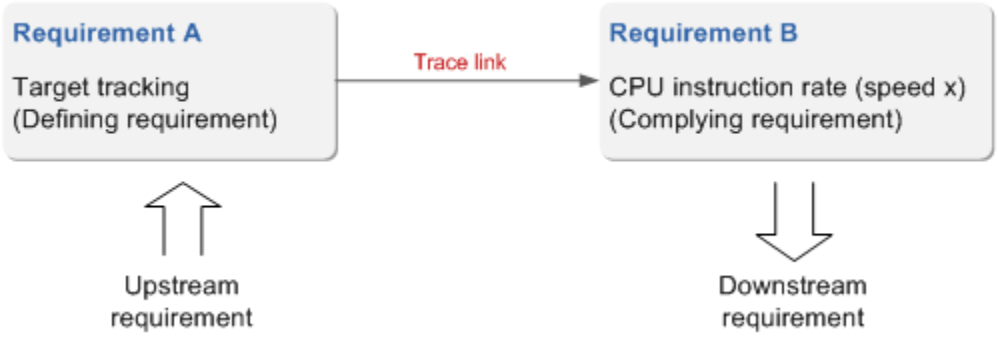
Relating requirements with trace links

A trace link creates a directional relationship between two objects, in this case, requirements. The relationship is conveyed by the terms *defining* and *complying*. A defining object specifies a condition that a product or a component must fulfill. A complying object partially or completely fulfills a condition specified by a defining object. Such a relationship establishes a path in which one object precedes the other.

© 2025 Siemens

For example, functional requirement A, for target tracking, defines hardware requirement B, for a CPU with a certain instruction rate. In this example:

- Requirement A precedes requirement B, with requirement B directly *downstream* in the complying path.
- Requirement B succeeds requirement A, with requirement A directly *upstream* in the defining path.



A requirement can assume both defining and complying relationships, continuing the path upstream and downstream. For example, a weight requirement may define a requirement to use aluminum. That complying requirement may in turn define temperature or environmental requirements consistent with the properties of aluminum. A requirement can have any number of defining and complying trace links.

Trace links can be created between requirements within a single project and also between requirements in two different projects. This two-way relationship allows for change analysis in both directions.

If you are considering a change to a product requirement, you can follow its trace links downstream to all complying requirements that were derived from the product requirement and eventually to all design elements that must comply. Conversely, an engineer working on a design element that must meet detailed, low-level requirements can follow those trace links upstream to the original customer requirements that define the design constraints.

Managing requirement structures

A requirement specification is a container for individual requirements. These items can be structured in a hierarchy of parent, child, and sibling relationships.

You can build a requirement specification structure in several ways:

- Build the structure manually by creating the structure items one at a time. First, you create a requirement specification as the peak item, and then you add the structure elements by creating requirements at lower levels.
- Import the structure from Microsoft Office Word to create all structure items simultaneously. Systems Engineering automatically generates a new requirement specification, complete with a hierarchy of

© 2025 Siemens

parent, child, and sibling items. The items can be either requirements or paragraphs, depending on the subtype you select.

- Copy an existing object as a template for creating requirement specifications, requirements, and paragraphs in a structure.

  You can create a requirement structure by copying an existing requirement structure with the same composition. For example, create the requirements for a new car by copying the requirements for the outgoing model. This process is sometimes called creating a deep copy.

Once created, you can maintain the structure by moving requirements up or down, or by promoting and demoting them. Moving a requirement up or down changes its position under its parent requirement; promoting or demoting a requirement moves it to another, higher or lower, parent.

You can send the requirements structure to the Relation Browser to view and expand the hierarchy of requirements.

Importing requirements

Depending on your company's working practices, you may choose to collect and organize requirements in a separate application, typically Microsoft Office Excel. When the requirements have reached a suitable level of maturity, you can import them into Systems Engineering.

Alternatively, you may choose to maintain requirements in live Excel and dynamically update the requirements in Systems Engineering.

Exporting requirements

You can capture a record of a requirement specification structure in Microsoft Office Excel format at any time. For example, you can:

- Export current requirement content and save the file for comparison with subsequent revisions.
- Export structure elements hierarchically to show ongoing relationships within the structure.

You can send the Excel file to customers, suppliers, and colleagues.

Using variants

You can create variant rules and apply them to the requirements structure to filter out structure elements that are not configured in a particular variant of the system. You can apply variant rules to a structure that is exported to Microsoft Office Word or Excel, as well as structures you are managing directly in Teamcenter.

© 2025 Siemens

3. Designing with Systems Engineering

Designing with Systems Engineering

When the term is used generically, systems engineering is a method of managing the design of complex, multi-domain engineering products or systems over the life cycle of the project. Decisions you make during the design process may also affect the manufacture of the product.

Teamcenter Systems Engineering provides an object model and tools for developing (authoring) and managing the data related to such projects. Teamcenter does not enforce a particular standard, methodology or workflow, and you can use Systems Engineering to support your preferred practices and procedures.

Systems Engineering provides tools to support systems design tasks. You may complete these tasks in an order that suits your business practices.

- Identify and document requirements.

  > Note:
  >
  > Some customers prefer to work with requirements without implementing a systems engineering methodology. For other customers, managing requirements and systems engineering overlap significantly; changes to requirements drive changes to the systems design and vice versa.

- Create several models of the product including functional, logical, and physical model structures.
- Elaborate each model by further decomposition into building blocks.
- Associate requirements with building blocks and components.
- Use modeling and simulation tools and techniques to validate the assumptions and design parameters for the systems, or building blocks, and their interactions.

Objects you work with

> Note:
>
> The Systems Engineering and Mechatronics Process Management data models share many objects.

- Absolute occurrence

© 2025 Siemens

A relationship between a parent and an item one or more levels lower down the structure. The parent is the context in which the absolute occurrence exists.

- Baseline

  A collection of items and the relationships between them that is established to ensure their continued existence. It enables their configuration to be reconstructed and audited. Baselines are often created to preserve the state of a requirements structure or product at a particular checkpoint.

- Budget

  The actual budgetary values for a specific element in the design, for example, weight, power consumption, or cost.

- Budget definition

  Budget definitions are attached to design structures to maintain budget information such as the cost of an assembly, the weight of an assembly, or its power consumption.

- Connection

  A pathway between two interfaces. A connection allows a payload to pass from the output interface that generates it to one or more input interfaces that consume it.

- Decomposition

  The definition of a system or product in terms of functional, logical or physical components. Sometimes called a *breakdown*.

- Diagram

  A graphical view into a functional design represented in the Teamcenter data model.

- Function

  The objects or building blocks of a functional design. Functions in a functional design are equivalent to the parts in a physical design.

- Functional model

  A structured representation of the functions, behaviors, activities, or processes of the system or product.

- Interface

© 2025 Siemens

The points on functions where inputs are consumed or outputs are generated. Sometimes called *ports*.

- Item

  A workspace object generally used to represent a product, part, or component. Items can contain other workspace objects including other items and object folders.

- Item revision

  A workspace object generally used to manage revisions to items.

- Logical model

  A structured representation of electrical signals, logical interfaces, and connections in the system or product.

- Occurrence

  A hierarchical structure relationship between an immediate parent and its child item or item revision in a precise structure. Sometimes called a *relative occurrence*.

- Physical model

  A structured representation of the tangible subassemblies and components of the as-built product.

- Snapshot

  A folder that contains a revision of a configured structure. A snapshot can be used to redisplay the as-saved structure.

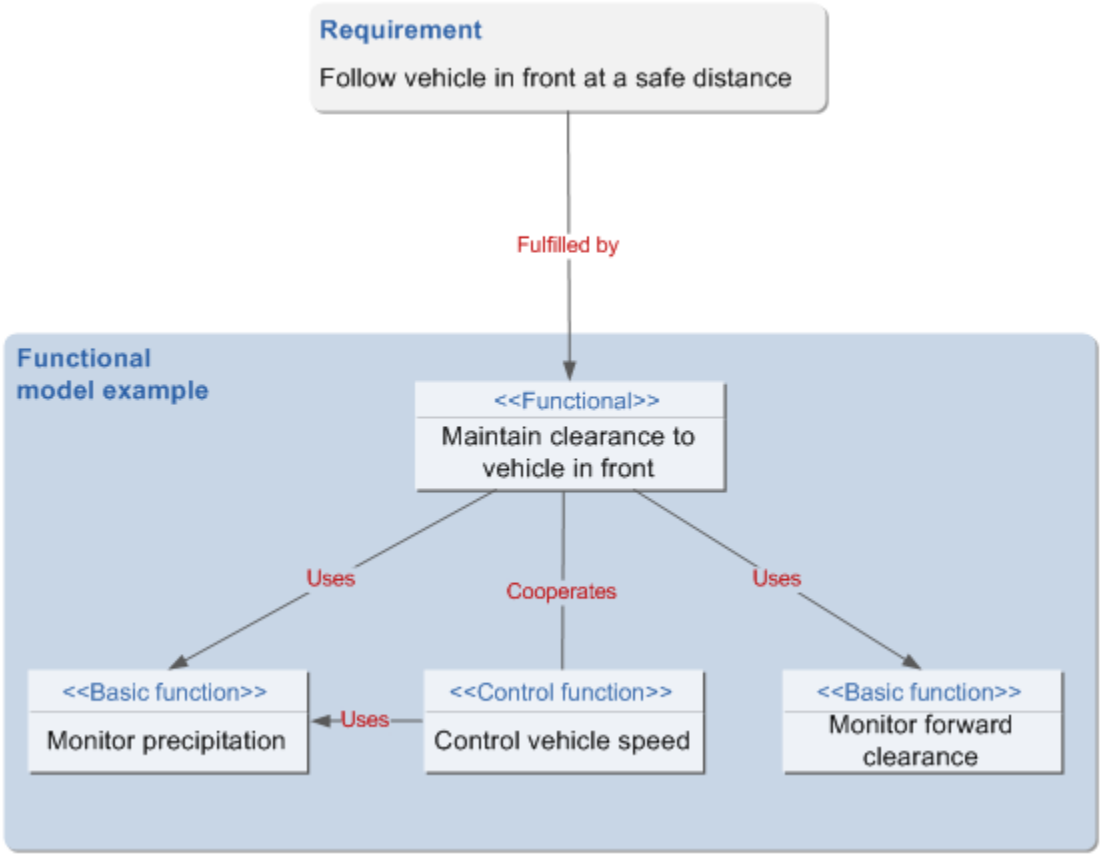
Using functional, logical, and physical models

Understanding the functional model

You can use Teamcenter to represent the functional model of a product or system. For example, the functional model of a car consists of various functional subsystems, such as the drive system and the braking system. In turn, the functional subsystems may contain various individual functions of the subsystems. In a functional model, components can be functions or functional subsystems.

The following example shows how you can represent the adaptive cruise control (ACC) system of a car with a functional model. The initial requirement for the ACC system is that the vehicle must maintain a safe distance from the vehicle in front on the highway.

© 2025 Siemens

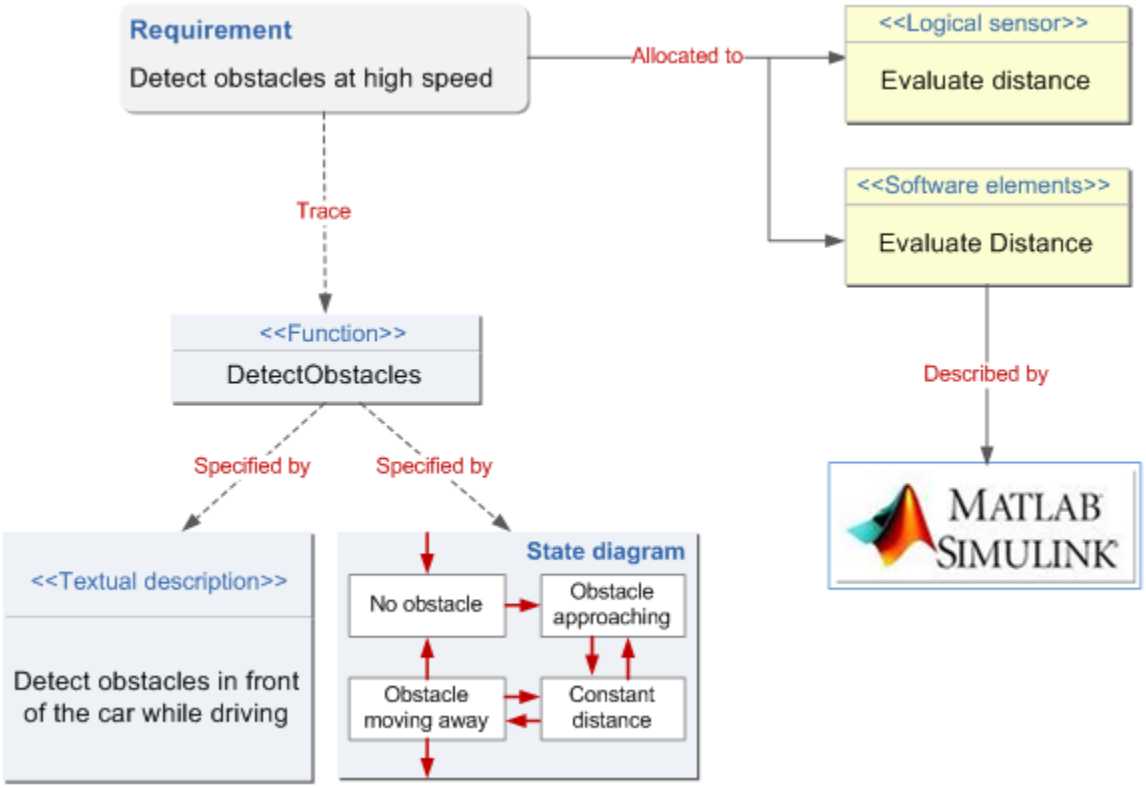


The solution is implemented both in hardware and software, but the actual implementation is not modeled at this point in the development process.

Understanding the logical model

By default, the logical model supports representation of electrical signals, logical interfaces of connectors and devices, and the connectivity between the interfaces (connections). For example, the logical connectivity for the functional subsystems may consist of a set of electrical devices, connectors, and logical connections between their interfaces. The following figure shows the logical model of the ACC system of the car, consisting of the logical sensor and software element.

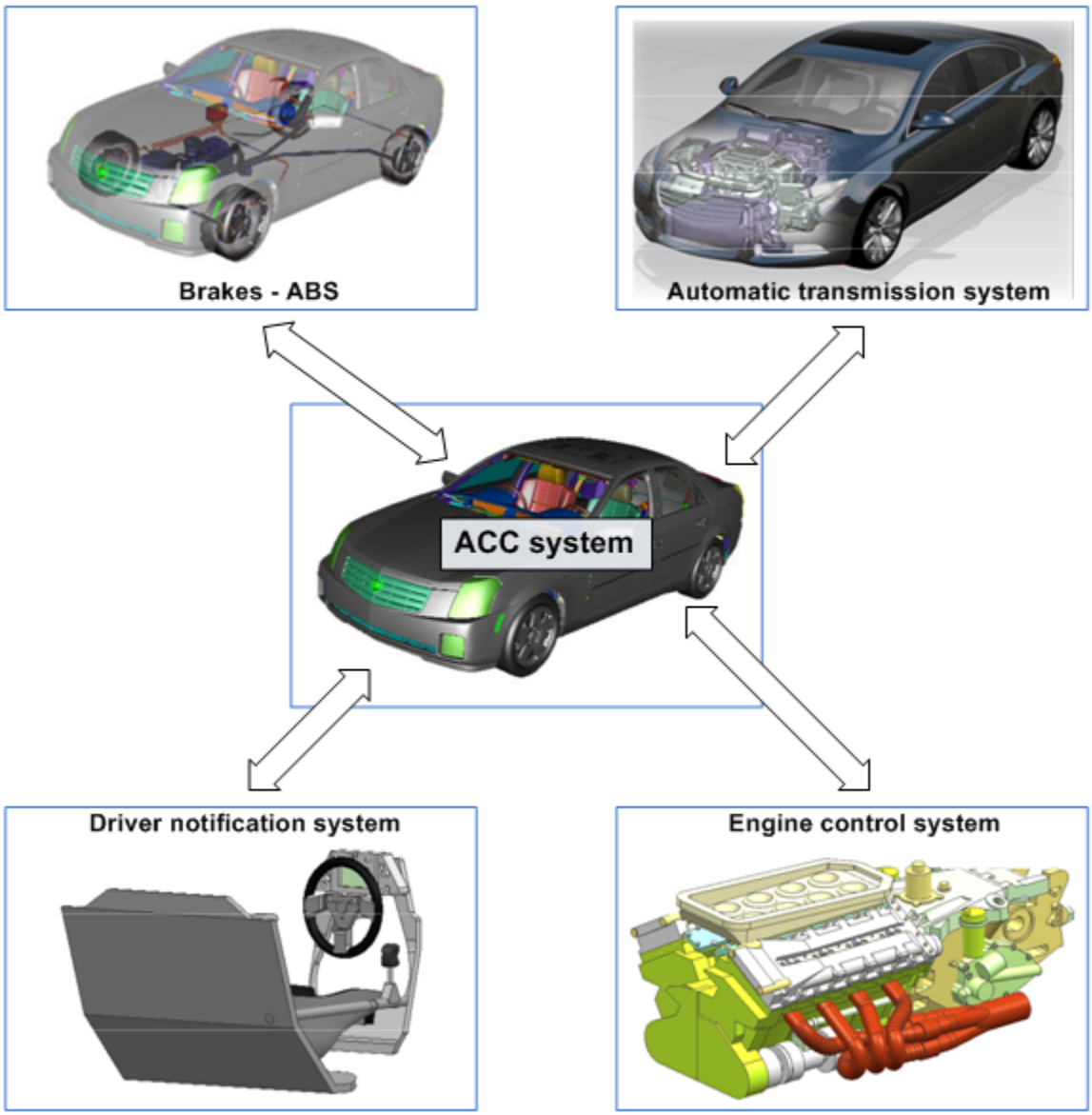
© 2025 Siemens



Understanding the physical model

The physical model represents the tangible subassemblies and components of the as-built product. For example, in an electromechanical product, it may comprise physical parts, assemblies, connectors, and the route associated with wires and cables. The physical model of the ACC system of the car and related assemblies are shown in the next example.

© 2025 Siemens



Creating functional models

What is a functional model?

A functional model is a structured representation of the functions, behaviors, activities or processes of the system or product you want to design or enhance. It describes the functions and processes, assists with the discovery of requirements, and establishes a basis for determining manufacturing and service costs.

This model may comprise functions, transformations, activities, actions, tasks, and other components. The design analysts develop this model and create a data flow diagram. This modeling activity is referred

© 2025 Siemens

to as creating a functional decomposition. If appropriate, you can use mathematical techniques or simulations to describe the relationships between components of the model and how they interact.

You represent components with building blocks in the diagrams and create trace links between the blocks to indicate how they relate to one another. After you identify components and their relationships, you can assign *requirements* to them. Functions satisfy the functional requirements of the product; each function may satisfy more than one functional requirement.

You normally express a function as a verb and an object, for example, **maintain clearance** or **transport material**.

A functional model is independent of a particular discipline such as software development or mechanical design. It supports the collaboration and defines what the product must do. Simultaneously, you can create a set of abstract solution models (logical models) that are organized into a subsystem decomposition and contain discipline-specific abstract solution elements. These two models evolve together and can generate technical constraints or additional or more detailed requirements.

The link between a function and a logical element can be represented as a **realized by** relationship or an **allocated to** relationship, indicating the function is allocated to a specific solution element.

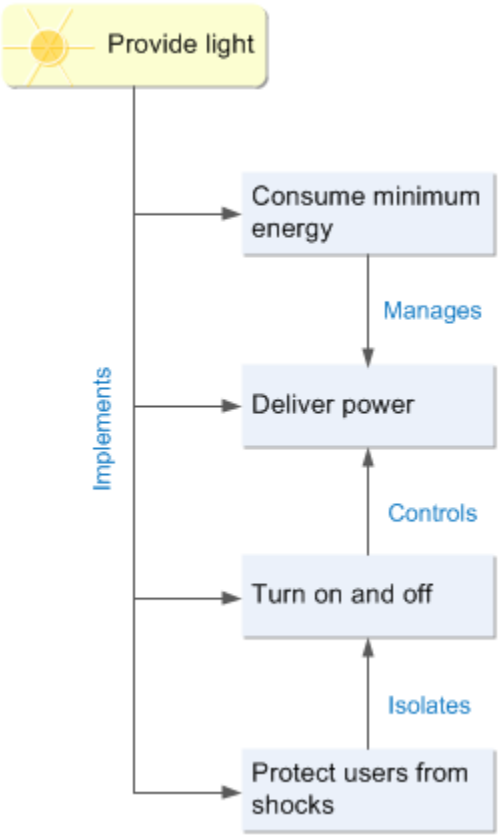
Working with functional models

When you design a system or product, you identify all the functions it should perform and equate each to a functional block. This process creates a functional decomposition of the system or product. Often, your starting point for creating the functional model is performing a high-level analysis of the requirements. A combination of requirements and precedents typically help you define the functions.

When the functional model is complete, you can allocate functional requirements to functions, ensuring that all functional requirements are appropriately allocated.

The following example shows the functional model of a light system.

© 2025 Siemens

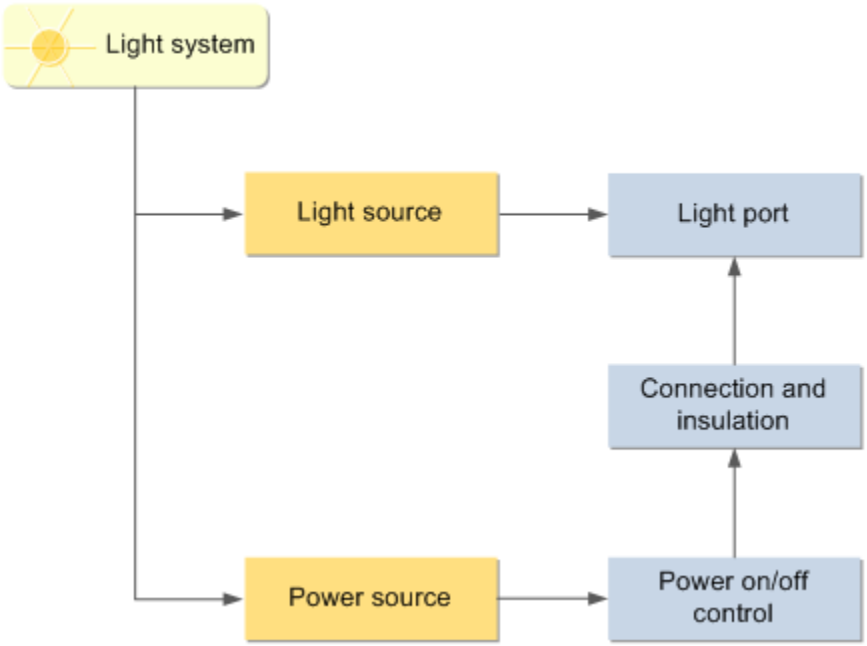


In this example, **Consume minimum energy**, **Deliver power**, and the other blocks represent generic functions. These functions typically cannot or should not be represented by items in the product structure.

After you create the functional blocks, you link the functional model blocks with functional connections. You allocate functional requirements to functions, ensuring all functional requirements are allocated.

You then create the logical model and connect the elements with logical connections, pins, and ports. The following example shows the logical model of the light system. The logical blocks are implemented by equivalent physical parts in the physical structure. Each functional connection has interfaces (or *ports*), each of which may be an input, an output, or bidirectional.

© 2025 Siemens



Creating functional models

You can create functional breakdowns with Microsoft Office Visio or directly with the **Systems Engineering** perspective. You can create an initial model structure in Systems Engineering, and then add Visio diagrams later. Likewise, you can create the initial diagram in Visio, then generate the functional model structure in Systems Engineering. Teamcenter keeps the functional model synchronized with any changes you make in Visio and saves both in the database.

You can nest functions in parent-child relationships in the familiar structure line format using Systems Engineering.

If you are creating a system that is based on a previous design or revising an existing design, you can copy another functional model structure or selected function blocks from it to use in your project.

If you use Visio, your administrator can create diagram templates to represent the various types of functions, connections, interfaces, and trace link relations in your system.

- A diagram template is a blueprint for the diagrams end users create. It defines the content and application domain of the diagrams. It holds information about the diagram members, relations, and the application domain for a diagram. A diagram template has a Visio template file associated with it. The Visio template holds the stencil or the collection of shapes for an application domain.

- A diagram is the graphical representation of a function and its underlying structure, based on the membership and relationship rules specified in the diagram template.

An interface may be input to the selected function, output from it, or bidirectional. A connection joins two interfaces. Dangling connections are not permitted in functional modeling; you must always

© 2025 Siemens

associate both interfaces with a connection or the system deletes the connection when you perform a save operation.

A functional model structure comprises multiple building blocks, each of which represents a discrete function of the system. (Similarly, you can create a logical model that represents the same system in terms of logical blocks.) A building block is represented in the **Systems Engineering** perspective by a structure line of the appropriate type, either functional, logical, or requirements.

You can delete a function with the **Systems Engineering** perspective or Visio unless:

- It is used in another functional model structure.
- There are multiple revisions of it.
- The only revision is in a released state.
- It is in a workflow.

If the function has children, you can optionally delete all of the children, subject to the same restrictions as when you delete a top-level function. Teamcenter also deletes any trace links and connections associated with the function.

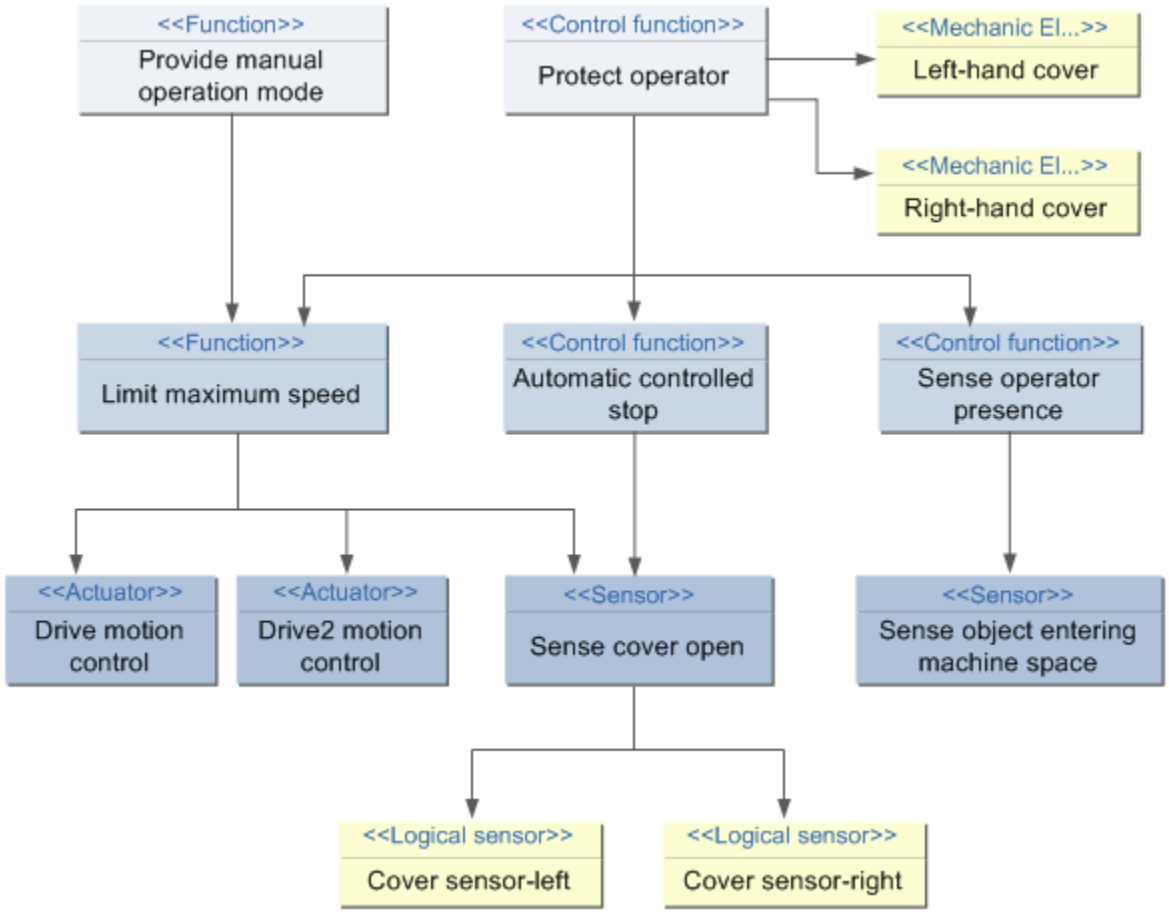
Creating functional models for machinery

Another task where it may be advantageous to create functional model structures is if you design machinery that must comply with the European Machinery Directive. This legislation requires you to prove that operator safety can be guaranteed through a standard development process. Changes to the existing machine design may affect many components, including hardware and software. You can create a functional model to support cross-discipline problem solving.

In the following example, the **Protect Operator** function is defined as a set of subfunctions. The function is controlled by software that uses inputs from sensors on different components. The sensors are not directly related to each other. The controlling software changes certain parameters that control or limit the machine's operating speed and block certain features completely.

You must understand all relevant functions to design in new requirements. You must also understand the functional decomposition of the machine to assess the impact of changes to any of the components.

© 2025 Siemens



Running accountability checks

You can check two structures for consistency to ensure all lines in the source structure are consumed in the target structure. You can compare the entire product or a selected assembly.

For example, you can compare the product structure and requirements structure to ensure all the requirements are accounted for in the final product design. You can also use the accountability check for impact analysis. For example, if a requirement changes, you can identify what physical parts are affected.

Teamcenter reports the following anomalies:

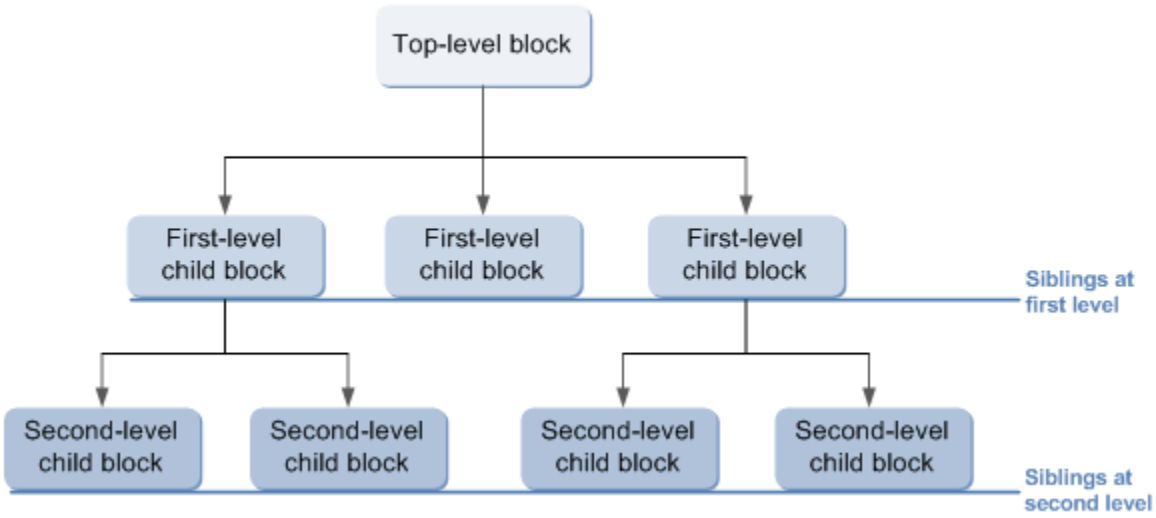
- Lines in the source structure that are not consumed in the target structure.
- Lines in the target structure that are not present in the source structure.
- Any line that is consumed more than once.
- Any line that is not fully consumed, for example, if a quantity of 3 is specified in the source but only two components are consumed in the target.

© 2025 Siemens

Creating building blocks and structures

When you create building blocks, you construct hierarchies of blocks that decompose systems and illustrate relationships among system elements. A building block can represent any element in a hierarchy, such as a function or a component of a product, a task in a work breakdown structure, or a job function in an organizational chart.

In Teamcenter, building blocks reside in folders, and a folder can contain any number of building blocks. Below each top-level building block, subordinate building blocks can be organized in multiple levels of parents, children, and siblings. To construct a view of an entire system, create a single building block for the system at the top level. At the next lower level, create child building blocks for the major subsystems. Below those, continue to decompose the system into finer levels of detail. For example:



In such a hierarchy, you can create building blocks at specific levels, promote them to higher levels, demote them to lower levels, and move them up or down within a level. For each building block, the number shows the level in the hierarchy and the position within the level. As the structure changes, Teamcenter renumbers values automatically for affected building blocks.

You can create trace links to and from building blocks, to show defining and complying relationships with other objects. Such relationships may exist between building blocks that reside within the same hierarchy. Also, building blocks may have trace links to objects that reside elsewhere in the same project or in a different project.

Measuring technical progress for design solutions

Defining budgets

As successive decomposition elicits lower levels of detail, technical measurements can help decision makers to determine whether a proposed design is achieving performance goals. Such assessments

© 2025 Siemens

at strategic intervals provide benchmarks of progress toward a solution that meets the customer's expectations.

In Systems Engineering, quantifiable parameters (such as technical performance measures, or TPMs) can be defined through *budgets*. When budget values are applied to structure elements, the parameters can be tracked at given points in the development cycle.

For example, initial values would be expected only to fall within a tolerance band, because early designs likely would not meet performance goals. As the solution matures, values would move increasingly closer to the goals.

Budget values can be calculated and reported from the lowest level to the system level, showing the status of downstream targets to:

- Verify targets that conform to parameters.
- Identify deficient targets for recovery or correction.

> Note:
>
> Teamcenter budgets can be applied not only in solution design but throughout the lifecycle, for example, on simulation data, on physical prototypes, and on the production system.

You define budgets for parameters such as *height*, *weight*, or *power*. A budget is a pattern (like a template) on which values for individual elements are based.

Systems Engineering implements budgets through Microsoft Office Excel. For each budget, you specify:

- A *unit* of measurement for the budget values, such as *inches*, *pounds*, or *watts*.
- An *expression* for calculating downstream budget values and reporting the results upstream.
- A default *export template* for generating the budget worksheet in live Excel.

Using these definitions, you can apply budget values to entire structures (the peak elements) and to parent elements and their substructures. The **Budgets** view displays the budgets in which a selected element participates.

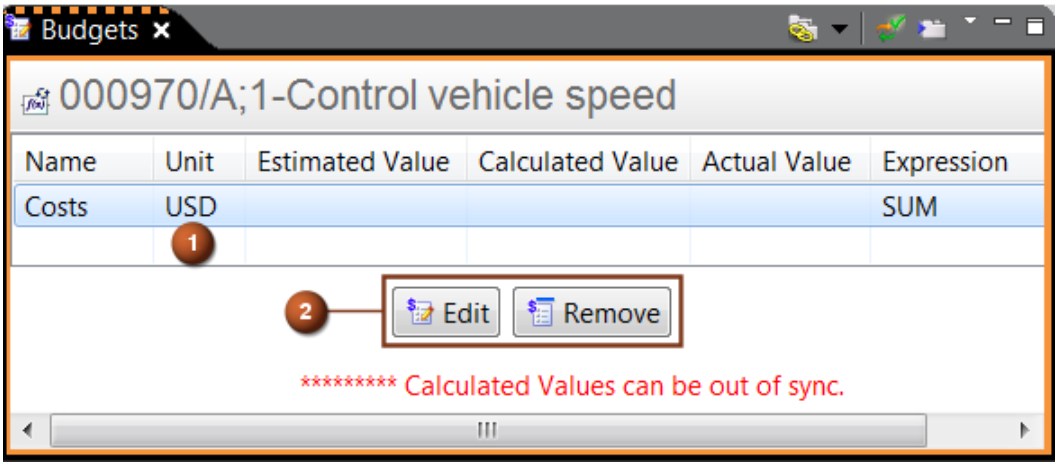
Viewing budget values

In the **Budgets** view, you can monitor the performance of downstream targets by applying technical measurements, such as Technical Performance Measures (TPMs), to structures and their members.

Budgets work through Microsoft Office Excel, in which you define and quantify parameters like *height*, *weight*, or *power*. With parameter values expressed in appropriate units (for example, *inches*, *pounds*, or *watts*), aggregates can be reported upstream to the system level.

© 2025 Siemens

You use Teamcenter live Excel to apply new budgets and initial values, and to edit existing values, for individual structure elements.



1	**Budgets** view data	Displays the actual values of each budget for the selected structure element.
2	**Budgets** view buttons	Allow you to edit a selected budget's values in live Excel and to remove the structure element from participation in the budget.

© 2025 Siemens

4. Creating trace links in models

Understanding trace links in models

A trace link establishes a directional relationship between two objects. In this relationship, one object defines a condition with which the other object must comply, that is, must partially or completely fulfill. A given object may define some objects and may also comply with other objects.

Within Systems Engineering, you can create defining and complying trace links between building blocks in the functional model. You can also create defining and complying trace links between building blocks in the functional model and objects in other models such as design elements in the physical model.

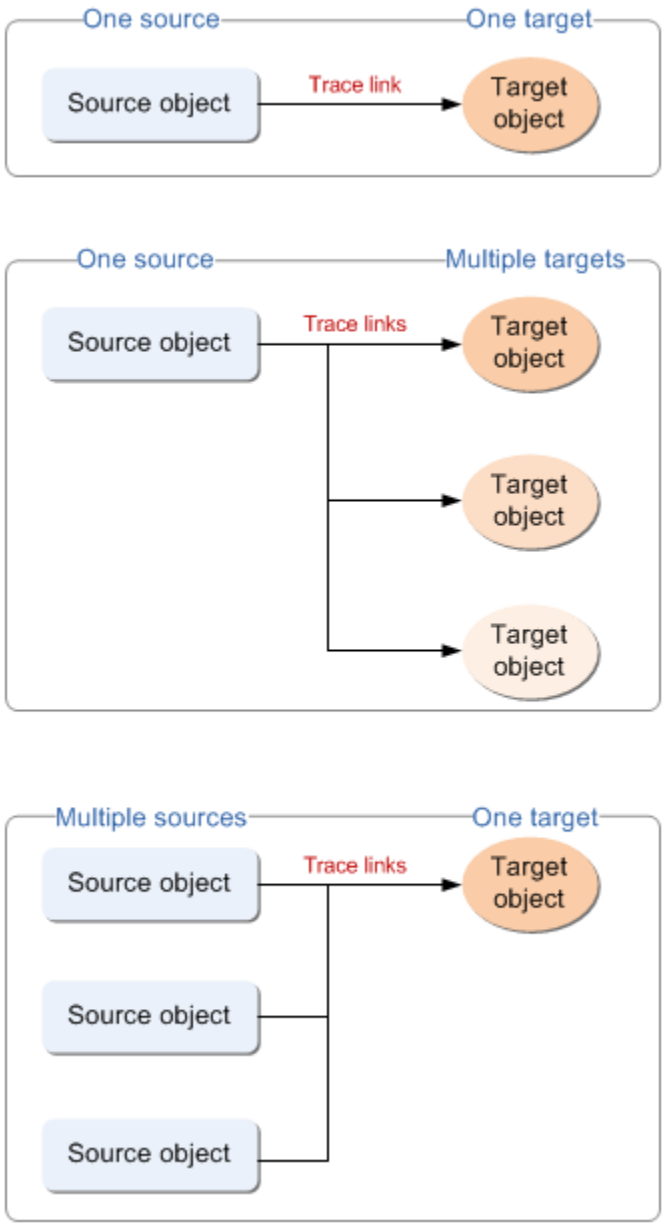
You can create trace links between:

- One source object and one target object, with a single trace link. The source has one defining trace link to the target; the target has one complying trace link from the source.

- One source object and multiple target objects, with multiple trace links. The source has one defining trace link to each target (that is, it has multiple defining trace links, one to each target). Each target has one complying trace link from the source.

- Multiple source objects and one target object, with multiple trace links. Each source has one defining trace link to the target, which has one complying trace link from each source.

© 2025 Siemens



Source and target objects may be workspace objects (for example, a requirement), items, item revisions, and absolute occurrences.

If you revise or otherwise edit a structure with assigned trace links, they are preserved when appropriate. Trace links are also captured if you baseline or snapshot a structure.

Trace links and absolute occurrences

Absolute occurrences are specific usages of an item in the context of a parent assembly. When a user enters in-context values or attachments on an absolute occurrence, Teamcenter overrides certain

© 2025 Siemens

structural data associated with the relative occurrence in the context of the parent assembly. Data you can override includes occurrence notes, occurrence type, quantity, and attachments.

If you want to create a trace link that is valid only in a particular context, you should attach it to the absolute occurrence. For example, if you build a truck with optional off-road suspension and specify higher tire pressures when the option is selected, you can specify the required pressure in a note attached to the in-context absolute occurrence line representing the wheel. You create a trace link between the requirement for a higher tire pressure and the same in-context, absolute occurrence line, rather than the occurrence line, because the requirement is valid only in the context of the off-road option.

© 2025 Siemens

© 2025 Siemens

5. Working with the data dictionary

Data dictionary overview

For systems design, key building blocks or components of designs are reused to save time and effort in typical functional and logical model-structure design activities. To facilitate reuse of these building blocks or components in multiple designs across several projects or programs, Teamcenter provides a central organizational repository for them. Designers can search for and find components based on criteria and reuse them in their designs.

Typically, the organizational repository is tightly controlled and the components or blocks in it have gone through a formal, customer-specific approval process. Once approved, the blocks or components are labeled as ready to use. This central organizational repository is referred to as a *data dictionary*. It allows you to organize data by your own criteria with the goal of reuse.

You can control access to the classes and classification objects making up a data dictionary. You do this in the **Access Control** pane.

Overview

A data dictionary is modeled in the Classification application as a library. The Classification administrator can build a customer-specific hierarchy under one or more libraries for organizing data, for example, signals and interfaces.

Depending on access privileges, users can then:

- Create, modify, and delete data from the dictionary.
- Associate a project or program with one or more dictionaries. Typically, the product design data is also associated with the same project or program.
- Search the dictionary for data meeting specified criteria.
- Add data from the dictionary to a functional, logical, or physical model structure.
- Create signals in structures and associate them with interfaces and connections.

Data objects you work with

When using the data dictionary, you work with the following objects:

- Connection

© 2025 Siemens

A pathway between two interfaces. A connection allows a payload to pass from the output interface that generates it to one or more input interfaces that consume it.

- Function

  The objects or building blocks of a functional design. Functions in a functional design are equivalent to the parts in a physical design.

- Interface

  The points on functions where inputs are consumed or outputs are generated. Sometimes called *ports*.

- Library

  An organizational node for classifying reusable blocks or components by customer-specific criteria. The library type identifies its content, for example, avionics signal data or MATLAB templates.

- Message

  A group of simple signals.

- Project or program

  The administrator can associate each library with one or more projects or programs. Multiple libraries can be associated with a single project or program.

- Signal

  Physical representation of an information flow being generated, processed, or conveyed within an electrotechnical system.

- Source

  The origin or transmitter of a signal.

- Target

  The receiver or destination of a signal.

Data dictionary example

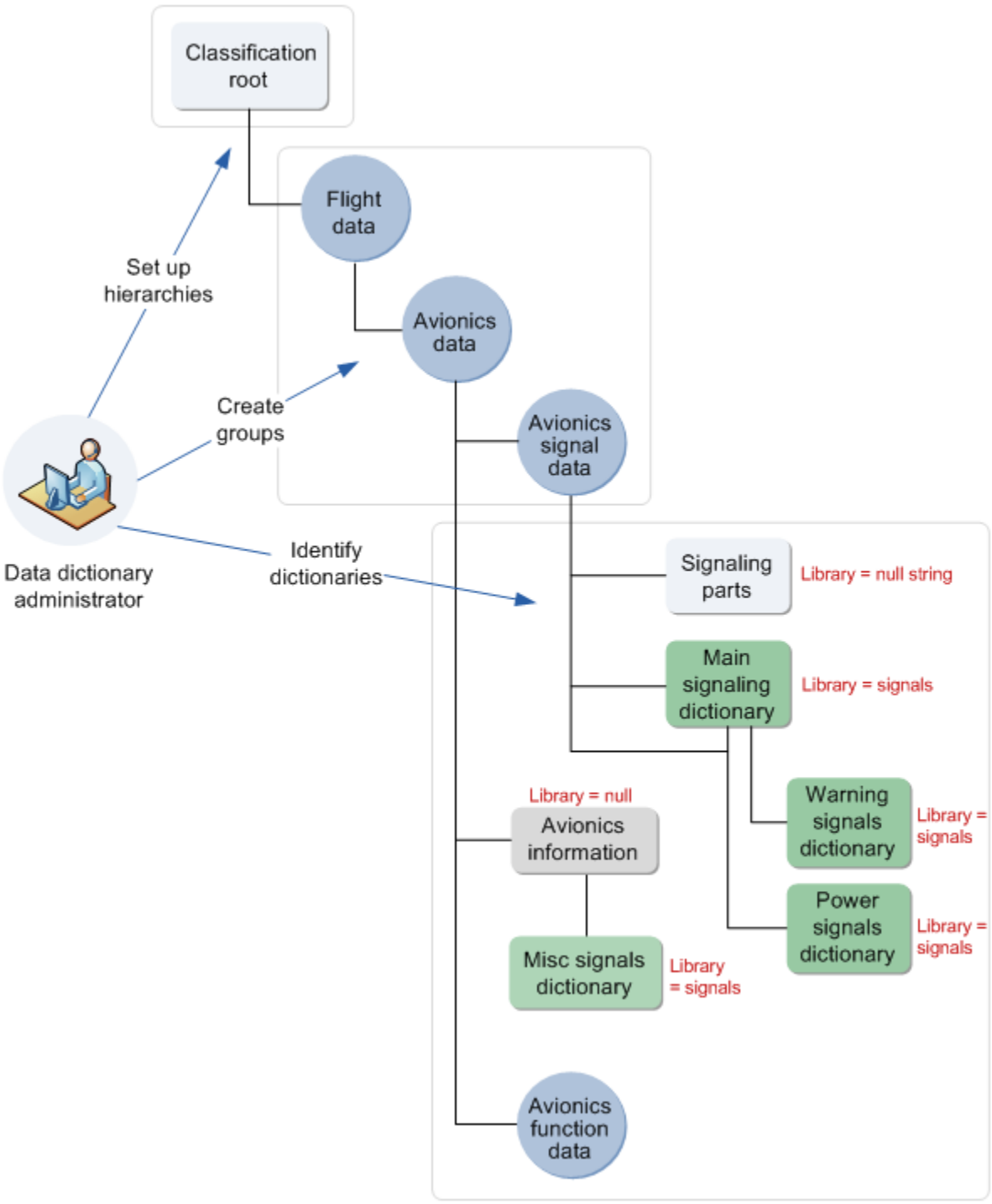
In the following example, flight data, avionics data, avionics signal data, and avionics function data are all Classification groups. They are intended for organizational purposes and cannot have additional qualifying information such as attributes. As you traverse down the hierarchy, the administrator typically qualifies the lower levels with more detail to facilitate quick searches.

© 2025 Siemens

On the main signals dictionary, you may want to capture such information as the subsystem that a signal is used in or the flight programs it is used in. The administrator can set these characteristics as Classification attributes on the main signal dictionary. Anytime functional designers add a signal to the main signal dictionary, they can then specify a value for the subsystem attribute, the flight program information, or both. When searching for the signals in the dictionary, functional design engineers can search based on a specific value for the subsystem, flight program attribute, or both.

In the example, the circles represent Classification groups and the rectangles represent Classification classes. The green rectangles are Classification classes that are marked as libraries, while the gray rectangles are classes that are not libraries.

© 2025 Siemens



Basic roles with the data dictionary

The following roles participate in the creation, administration, and use of the data dictionary. These roles are not provided with the shipped product, but you can create them in the Organization application if appropriate.

© 2025 Siemens

- Data dictionary administrator

  The data dictionary administrator sets up and maintains dictionary hierarchies and attributes on dictionaries that facilitate fast searching for objects in the dictionaries. The data dictionary administrator is not aware of the details of the applications consuming these dictionaries but acts on requests from the owners of applications that consume the dictionary.

  This role typically works only with the Classification Admin application.

- Project or program administrator

  The project administrator creates projects for specific intents, assigns resources to the project, and assigns existing dictionaries to the project. This role may also assign functional designs to the project, although this step could be completed by the design engineer working on the design, depending on your processes for adding designs. Application owners may also be project administrators or a resource in the project.

  The program administrator performs similar tasks with program data.

  The project administrator typically works in the Project application but may also use My Teamcenter. The program administrator typically works in the Program application but may also use My Teamcenter.

- Systems or design engineer

  The systems or design engineer designs and creates functional and logical model structures. Work may begin on these designs while the corresponding projects are still being set up. Once the project configuration is sufficiently mature, the design may become part of the project, depending on your processes. The systems or design engineer searches for objects using Classification to navigate through the dictionaries assigned to that project. The objects found are instantiated into the design and (optionally) associated with other objects in the design. For example, if the systems or design engineers search dictionaries of signals, they may associate the signals found to sources and targets in a logical model structure.

  This role works in CAD, Systems Engineering, and structure editor applications, but may also use My Teamcenter and Classification. Classification Admin is used only if your company policies allow the systems or design engineer to put signals directly into the dictionary.

Basic tasks with the data dictionary

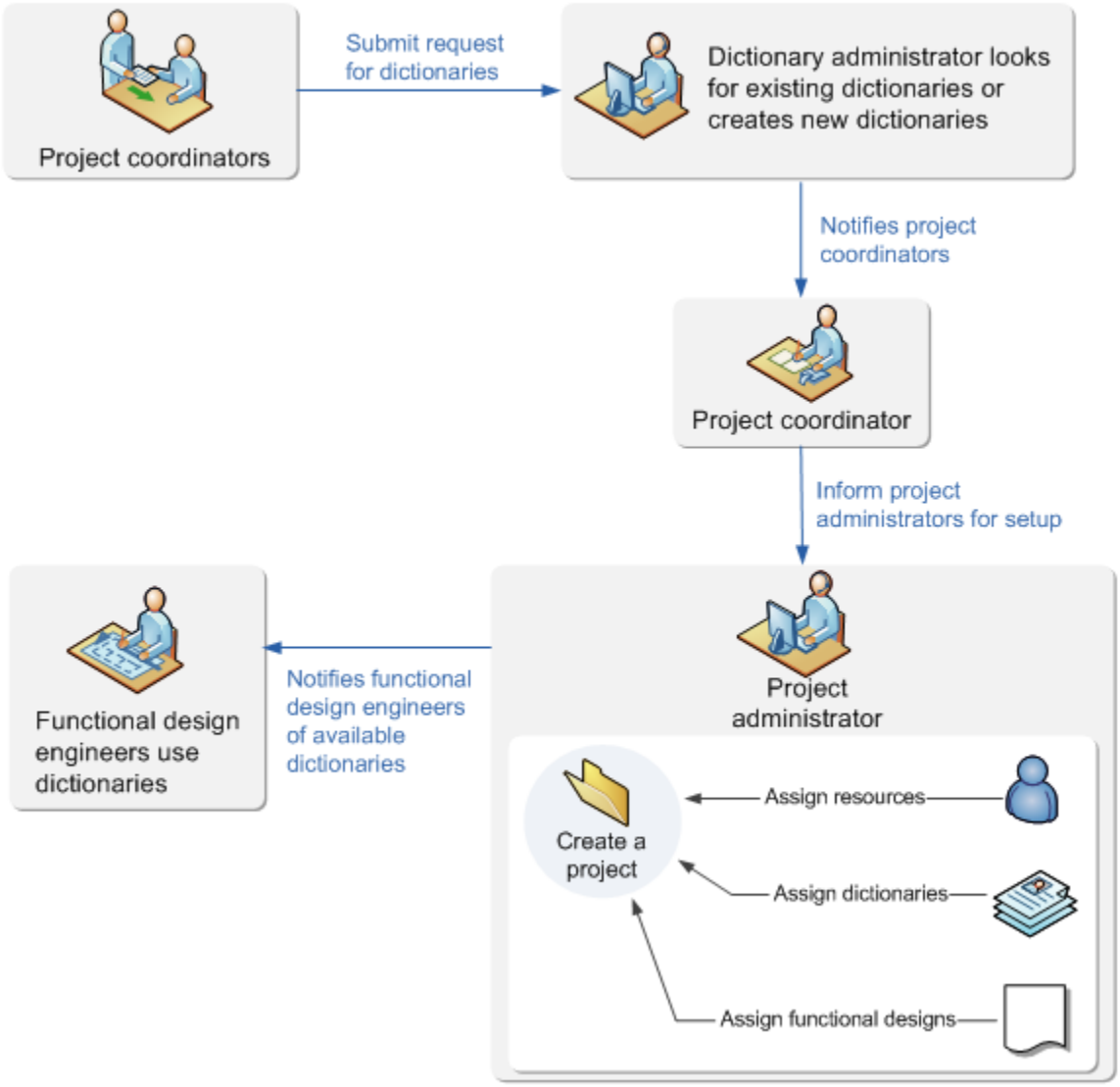
The project administrator:

- Creates, modifies, and deletes projects.

- Assigns resources (users, groups, and roles) to a project.

© 2025 Siemens

- Removes users, groups, and roles from a project.
- Assigns, removes, and views dictionaries for a project.
- Associates designs to a project.
- Assigns access permissions for a project.



Systems and design engineers may use the following typical task flow for populating and accessing elements in a data dictionary:

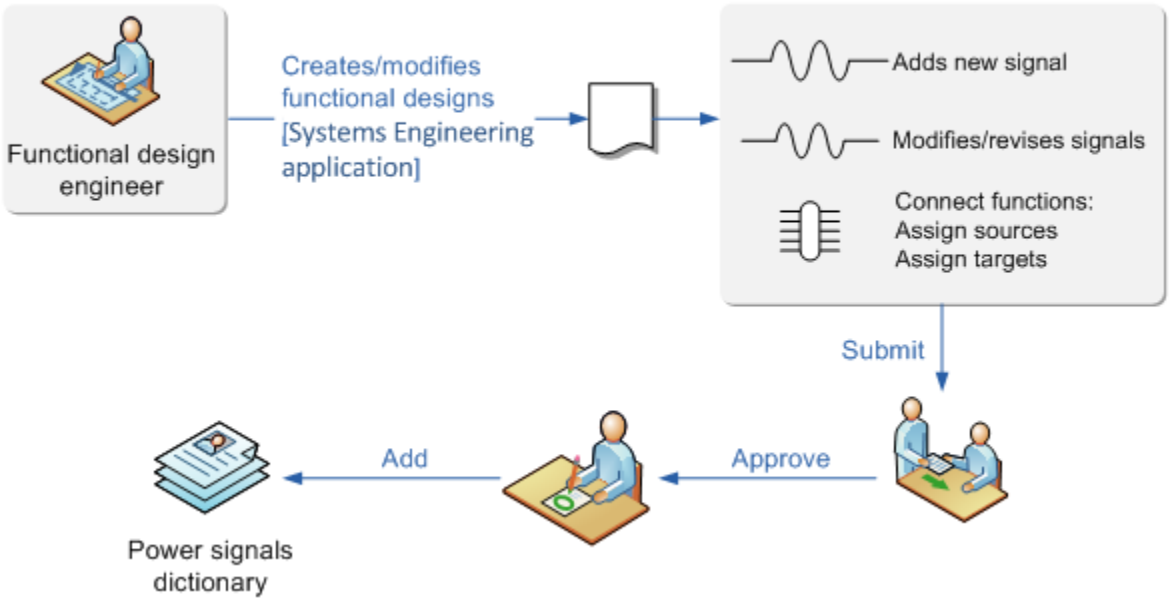
- Works in an assigned project.

© 2025 Siemens

- Views the dictionary organization scheme in the Classification perspective and then navigates, browses, and searches the hierarchy for signals of interest.
- Adds functions, ports, and signals from the dictionary to the functional or logical model structure.
- If the required signal is not available or a signal needs modification:

  1. Creates a new signal or revises an existing signal in My Teamcenter.
  2. Submits a request for the data dictionary administrator to add the signal or signal revision to the dictionary.

     The request is submitted to a company-specific approval process and the signal or signal revision is added to the dictionary.



The data dictionary administrator:

- Creates a dictionary hierarchy.
- Creates or modifies attributes in a dictionary.
- Adds signals to the applicable Signals library.
- Modifies or re-parents the dictionary hierarchy, if needed.

© 2025 Siemens

- Deletes dictionaries or their contents.

> Note:
>
> Deleting a dictionary does not delete its contents, that is, the classified data assigned to it. Classified objects in a dictionary must be deleted by a separate procedure. Also, you must remove any associations with projects before you can delete a dictionary.

© 2025 Siemens

A. Overview of administrative tasks

Overview of administrative tasks

Before you can use Systems Engineering, the administrator or other person with similar privileges must configure the system for your working practices and environment.

Configuring Teamcenter Systems Engineering

Configuring the icon display in structure columns

For viewing requirements in Systems Engineering, you set the following preference:

- **IconShownColumnProperties**

  Specifies the columns where an icon is displayed in the requirements or model structure instead of a string value. The default values are **bl_has_trace_link** and **bl_rev_Fnd0ListsCustomNotes**.

Administering functional models

Before working with functional models, no special configuration steps are necessary. You must install a standard Systems Engineering license to enable this feature.

Administering projects and programs

No special configuration steps are necessary to use Systems Engineering with projects or programs. However, you must set the appropriate general Project and Program application preferences.

Configuring trace links

Before working with trace links, you must:

- If required, set the **Calculate Defining/Complying Objects** option to **On**. If this preference is **ON**, defining and complying objects properties for **SpecElement**, **SpecElement Revision**, **RequirementSpec**, **RequirementSpecRevision**, **Functionality**, and **FunctionalityRevision** objects as well as structure lines are visible by default.

- If required, set the **Fnd0RuntimeTraceProperty** type constant to **True** in the Business Modeler IDE to see the properties for other types of business objects.

© 2025 Siemens

Configuring Microsoft Office for requirements authoring

If you use the Microsoft Office Word for requirements authoring, you must create and manage specification export template documents. Specification export template documents are Word files that control the content and format when you export a structure from Systems Engineering.

Likewise, if you export to Microsoft Office Excel, you must create and manage Excel export templates. Excel export templates are Microsoft Excel files that define the property data and output format used when you export objects from Teamcenter to Excel.

If you use the Excel import/export feature, set the **TC_setProperties** preference to determine if changed objects imported from Excel are automatically overwritten or if the user is prompted for confirmation.

Install the Teamcenter Diagramming Visio Add-in

To create or edit functional models of systems using Microsoft Visio integration with Systems Engineering, you must separately install the Teamcenter Diagramming Visio Add-in.

> Note:
>
> The Teamcenter Diagramming Visio Add-in requires the following software:
>
> - Microsoft Visio Professional version. The Standard Edition is not supported.
> - Microsoft .NET framework.
> - Microsoft Visual Studio Tools for Office (VSTO)
>
> The bit version of VSTO must match the bit version of the operating system.
>
> Make sure these products are installed on the host before you install the Teamcenter Diagramming Visio Add-in.
>
> For compatible versions of these application, refer to the hardware/software compatibility matrix.

1. Browse to the *additional_applications\diagrammingvisioaddin* directory.
2. Double-click the setup.exe file to launch the **Teamcenter Diagramming Visio AddIn InstallShield Wizard**.
3. In the welcome panel, click **Next** to start the installation.
4. In the **License Agreement** panel, select the option to accept the license agreement and click **Next**.
5. In the **Pre-requisites required** panel, verify that the pre-requisites are installed on your system and click **Next**.

© 2025 Siemens

6. In the **Setup Type** panel, specify if you want to install the add-in for all users, or only for the current user, and click **Next**.

7. In the **Choose Destination Location** panel, select the location for the installation and click **Next**.

8. In the **Teamcenter Diagramming Visio Connection** panel, accept the available port numbers, or enter new port numbers, for the **Visio Diagramming Tool Service** and the **Diagramming Service**, and click **Next**.

9. In the **Start Copying Files** panel, review the settings and click **Next** to start the installation.

10. In the **InstallShield Wizard Complete** panel, click **Finish** to exit the wizard and complete the installation.

Configuring the Microsoft Office Visio diagramming integration

The administrator creates the appropriate number of diagram templates for the customer's domains and functional content. End users create each diagram using one of the defined templates; freeform creation of diagrams is not permitted. Once created and initially populated, a diagram may not change its template. For Microsoft Office Visio diagrams, the template holds a Visio template file that provides Teamcenter-enabled shapes to represent Teamcenter objects on diagrams. It also provides the associated mapping XML file that associates them with Teamcenter object types and properties.

Administering a data dictionary

Before using the data dictionary, you must set the following preference and global constant:

- **ICS_classifiable_types** preference

  Specifies the types of objects you can classify, for example, **Message** and **Signal**.

- **Fnd0FilterEntriesForSignal** global constant

  Specifies the libraries that are visible to the user in the Classification tree.

You must also set the necessary general Classification Admin and Classification preferences.

Each user must have a data dictionary author or access license.

© 2025 Siemens