TEAMCENTER

Integrated Program Planning and Execution

Teamcenter 2412

SIEMENS

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

1. IPP&E Overview

What is Integrated Program Planning and Execution?

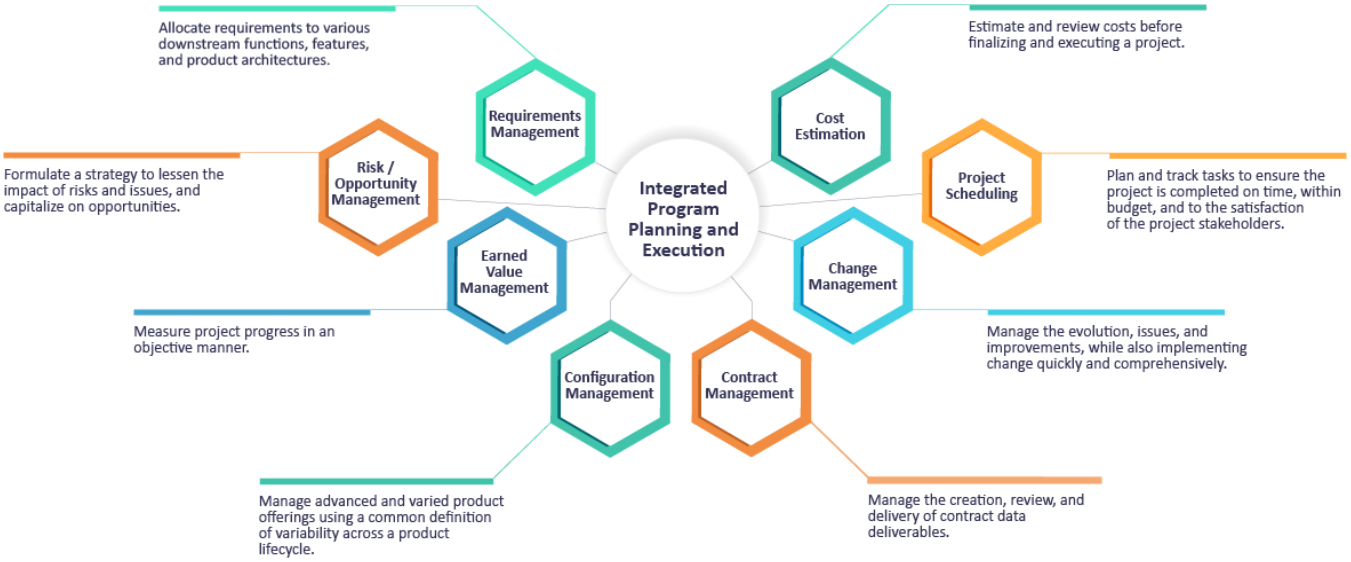
Organizations use varying approaches to plan and manage pursuits and programs, often relying on several versions of Word documents, Excel spreadsheets, and scheduling tools. Gaining a clear view of what is happening within a program is difficult and a main contributor to the high percentage of failed programs. The Integrated Program Planning and Execution (IPP&E) solution allows project planning that integrates cost, schedule, risk and technical requirements in a fully planned, resourced, and budgeted program. It allows configuration control not only of products but also of the project plan. It also communicates the status of requirements to users.

Using IPP&E, you can:

- Create and manage the work breakdown structure (WBS), including work packages.
- Link requirements, statement of work, and contract terms and conditions to the WBS structure.
- Estimate labor and material costs for the work packages.
- Generate an integrated master schedule (IMS) from the program WBS.
- Execute work packages and manage deliverables using the master schedule.
- Manage program risks and opportunity, including risk mitigation and incorporation.
- Update the program WBS and schedules to handle failed, missed scope, and delayed tasks during execution.
- Generate program management reports.

IPP&E integrates a host of domains, including risk, requirements, scheduling, and cost management across the program. Bridging the gap between capture and program execution improves bid accuracy and communication of the status of the pursuit or program. This in turn leads to a better capture win-rate and improved program performance.

The following graphic shows the integrated IPP&E features.



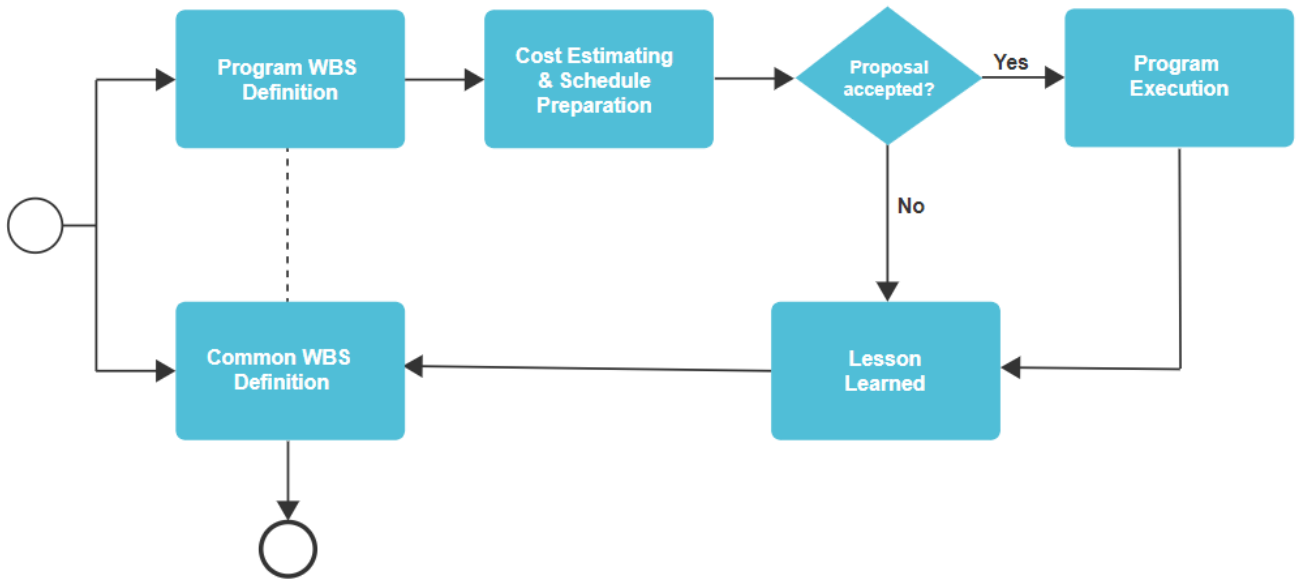
Where do I go from here?

Administrator	
Install Active Workspace and the IPP&E solution.	See the Active Workspace installation on Windows and Linux.
Use IPP&E workflow handlers to extend and customize tasks.	See the IPP&E workflow handlers in the appendix section.
Business User	
Get started with the basics of using the Active Workspace.	Refer to Active Workspace Fundamentals.
Get familiar with the page layout of the Active Workspace.	See the section describing the page layouts.
Learn more details about the IPP&E business process, roles, and business objects.	Start with overview of the business process.
Use the IPP&E solution.	Start with the section on understanding the program WBS.

Understanding the IPP&E business process

Overview of the business process

A typical IPP&E business process is shown below. If the business process differs in your organization, you can configure IPP&E to suit your business needs.



IPP&E provides the following key elements of the program management process:

- **Program WBS definition**, including the creation of an **organizational breakdown structure** and **work packages**
- **Cost estimating and schedule preparation**
- **Program execution**
- **Common WBS definition**
- **Lessons learned**

Defining the program work breakdown structure

In the program work breakdown structure (WBS) definition process, you define the work necessary to accomplish the program goals, based on the specification of the product, the statement of work, and any terms and conditions imposed by the program. The WBS is organized in a hierarchical manner with the lowest level node representing a task for specific deliverables. This lowest level node of the WBS is called a *work package*. For each work package identified in the program WBS, you also define the functional work packages for the functional organizations necessary to accomplish the work. IPP&E provides business objects and capabilities for the program WBS definition phase of the program management process.

Preparing cost estimates and a schedule

In the cost estimation and schedule preparation process, you estimate the labor, material, and other costs such as travel for the work packages in the program WBS. These estimates are then sent for review and approval by the functional organization, the Integrated Product Team (IPT), and the finance department. A schedule is created in the external scheduling software, based on the work packages defined in the program WBS.

Executing the program

In the program execution process, the schedule generated in the external schedule software is executed in Teamcenter. All the deliverables generated during program execution are attached to their corresponding work packages. You can also add new WBS elements in the program WBS during execution to handle failed or missed scope tasks or to mitigate a risk. After updating the WBS, you can update the schedule.

Defining the common work breakdown structure

The common work breakdown structure (WBS) is a template that you can use for generating a WBS for a specific program. When you define the common WBS, you create a library of work breakdown structures representing all categories of products and programs relevant to the organization. In this common WBS, you set filters on WBS elements to mark them for specific categories. You can then configure it for a particular set of filters to generate a WBS for a specific program. You create the common WBS for your organization using Product Configurator.

Implementing the lessons learned from the program

When a program execution is complete, you can use the WBS artifacts created for that program to enrich the common WBS. You compare the program WBS with the common WBS with the **View** → **Compare** command to identify updates and actions. All artifacts in both the program WBS and the common WBS are under configuration control to manage variability, allowing for a valid comparison. This is a closed-loop process to support continuous improvement.

Comparisons can also be made at any time prior to program completion to capture critical improvements.

IPP&E roles

During the various phases of the integrated program planning and execution process, several users in different roles in the organization perform tasks as listed below.

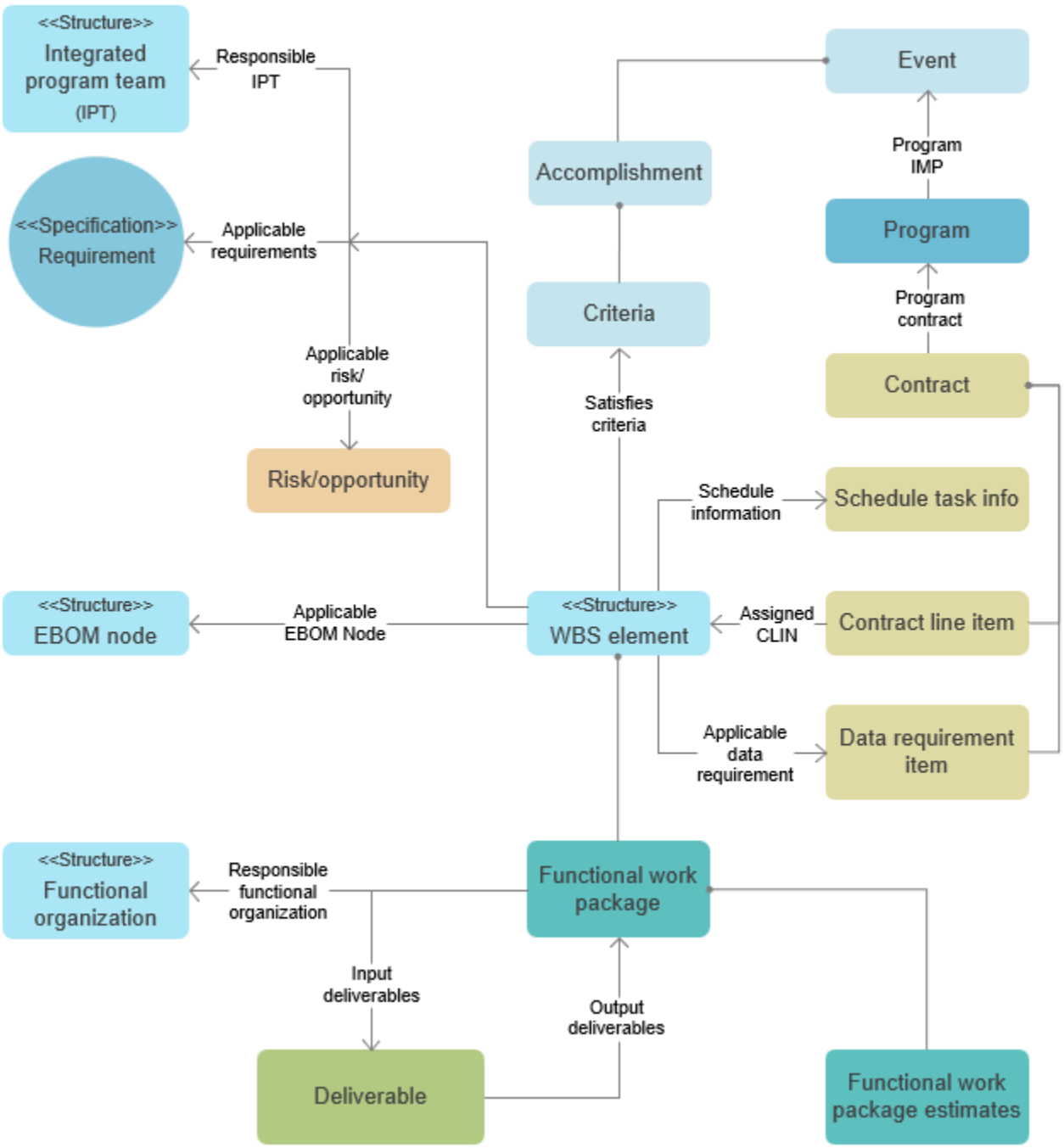
> Note:
>
> Your organization may use different roles or alternative terminology to identify these roles.

Role	Business process tasks performed
Corporate Program Manager	Manages the common WBS library.
Program Manager	Plans and executes a specific program.
Functional Organization Lead	Estimates and executes a functional work package.
Functional Organization User	Creates the deliverables for a functional work package.

Role	Business process tasks performed
Finance Lead	Reviews and approves the estimates for the program WBS.
HR Lead	Creates and manages the organizational breakdown structure (OBS).

IPP&E objects

During the integrated program planning and execution process, Teamcenter objects are created at various stages to store product and process data. These objects are associated with each other through relationships. The following diagram shows a conceptual representation of business objects and relationships added by IPP&E to store and process data.



Object	Description
Project WBS	Work breakdown structure that is associated with the program, project, and subprojects.
Common WBS	A WBS template that you can use to create a project WBS.
WBS Root Element	This is the top node of the WBS hierarchy. If the **Common WBS** property value of WBS root element is **True**, the structure is the common WBS. If the value is **false**, the structure is the project WBS.

Object	Description
WBS Element	This is a node in the Work Breakdown Structure (WBS). A leaf node of the WBS is referred to as a *work package*, which represents a task in the program schedule.
Task Info	It contains schedule information for the WBS work package.
Functional Work Package (FWP)	It contains details of the work for a specific functional organization. Many work packages, each created for a distinct functional organization, can fulfill the work identified for a WBS work package.
FWP Estimates	It contains the estimates, including labor, material and other costs for a functional work package.
Functional Org	This is a specific functional organization node in the organization structure of a company. An Organizational Breakdown Structure (OBS) is a hierarchical organization of functional organization nodes. Individual nodes of the OBS are called OBS elements.
Deliverables	This refers to the deliverable data resulting from the execution of a functional work package.
Risk/Opportunity	This is used to indicate a risk or opportunity that may occur during the execution of the program. A risk may result in increasing the cost or delaying the completion of the program. An opportunity may reduce the cost and can have a positive impact on the program schedule.
Program	It refers to the project that is being planned and executed.
Integrated Product Team (IPT)	This is the hierarchical representation of the team responsible for the planning and execution of a program.
Integrated Master Plan (IMP)	This is an event-based, top-level plan consisting of a hierarchy of program events. Each event is decomposed into specific accomplishments, and each specific accomplishment is decomposed into specific criteria.
Event	It refers to a program assessment point that occurs at the culmination of significant program activities.
Accomplishment	This is the desired result prior to or at the completion of an event indicating the level of the program's progress.
Criteria	It refers to the definitive evidence that a specific accomplishment has been achieved.
Contract	It refers to an agreement to provide a product or service to a customer. It typically includes product and data requirements, statement of work, and terms and conditions.
Contract Line Item Number (CLIN)	This constitutes part of a contract that breaks down the contract based on the commodities being procured. CLINs help identify the supplies or services to be acquired as separately identified line items in a contract to provide for accounting traceability.
Data Requirement Item	This is an individual data requirement for a contract. A collection of these objects is called the Contract Data Requirement List (CDRL).

Object	Description
Requirement	This refers to the requirements for the product.
Engineering Bill of Materials (EBOM)	This is the engineering definition of a product.

2. Planning and executing an integrated program

Defining a program work breakdown structure

Understanding the program WBS

A program WBS is a hierarchical and incremental itemization of a project into phases, deliverables, and work packages. In Teamcenter, it is represented by a tree structure that shows the subdivision of effort required to achieve an objective. In a project or contract, the WBS is developed by starting with the end objective and successively subdividing it into manageable components in terms of size, duration, and responsibility, for example, systems, subsystems, components, tasks, subtasks, and work packages. The WBS includes all the steps necessary to achieve the objective. The following graphic shows part of a WBS for an aircraft system:

Element	WBS Number	ID	Revision Name
Demo Common Aircraft System	1	WBSR-000001	Demo Common Aircraft System
Air Vehicle	1.1	WBS0000001	Air Vehicle
Propulsion	1.1.2	WBS0000002	Propulsion
Engine Inlet	1.1.2.1	WBS0000003	Engine Inlet
Create Engine Inlet Design	1.1.2.1.1	WBS0000004	Create Engine Inlet Design
Conduct Engine Inlet Analysis	1.1.2.1.2	WBS0000005	Conduct Engine Inlet Analysis
Engine Compressor Section	1.1.2.2	WBS0000006	Engine Compressor Section
Engine Compressor First Stage	1.1.2.2.1	WBS0000007	Engine Compressor First Stage
Create Engine Compressor First Stage Design	1.1.2.2.1.1	WBS0000008	Create Engine Compressor First Stage Design
Conduct Engine Compressor First Stage Analysis	1.1.2.2.1.2	WBS0000009	Conduct Engine Compressor First Stage Analysi
Perform Engine Compressor First Stage Verification by Analysis	1.1.2.2.1.3	WBS0000010	Perform Engine Compressor First Stage Verificat
Engine Compressor Second Stage	1.1.2.2.2	WBS0000011	Engine Compressor Second Stage
Create Engine Compressor Second Stage Design	1.1.2.2.2.1	WBS0000012	Create Engine Compressor Second Stage Desig
Conduct Engine Compressor Second Stage Analysis	1.1.2.2.2.2	WBS0000013	Conduct Engine Compressor Second Stage Ana
Perform Engine Compressor Second Stage Verification by Analysis	1.1.2.2.2.3	WBS0000014	Perform Engine Compressor Second Stage Verif
Engine Compressor Third Stage	1.1.2.2.3	WBS0000015	Engine Compressor Third Stage
Create Engine Compressor Third Stage Design	1.1.2.2.3.1	WBS0000016	Create Engine Compressor Third Stage Design

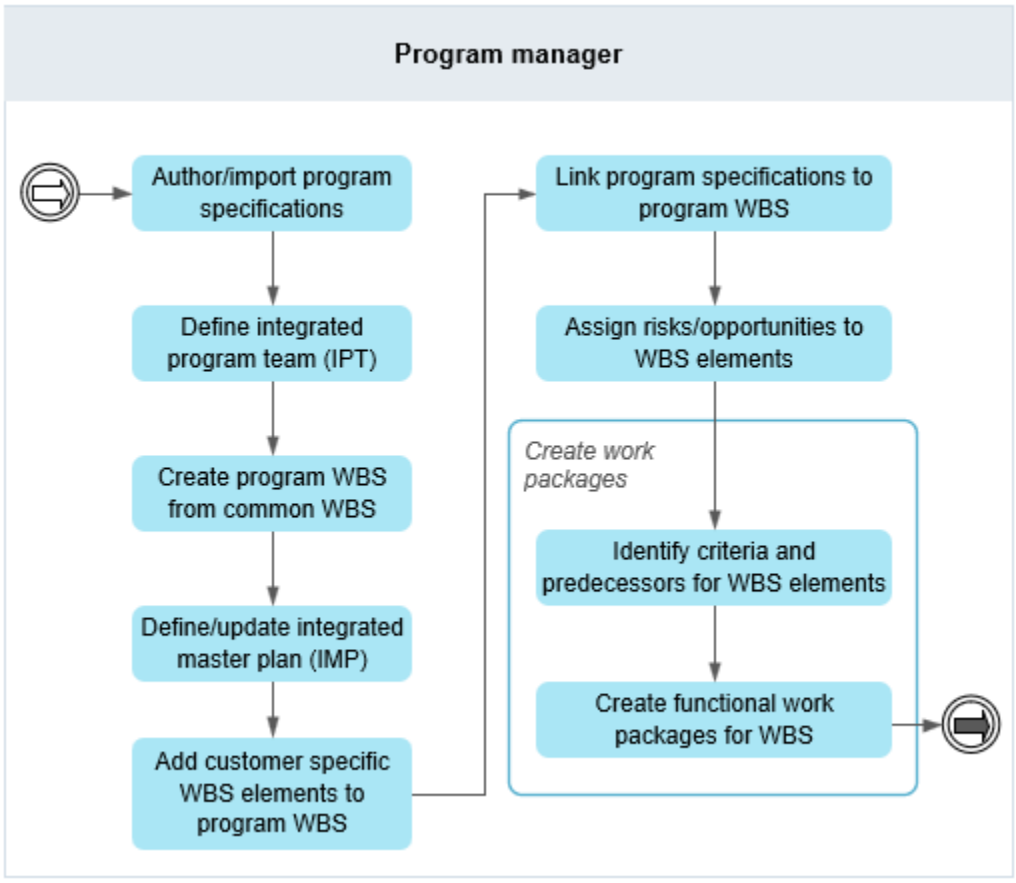
Each element of a program WBS contains:

- The technical requirement for the WBS element.
- The product structure node (logical or physical) for the WBS element.

- The integrated master plan (IMP) criteria that are satisfied by the completion of the WBS element.
- The work needed for the completion of the WBS element. Each work package for the WBS element should include the following information:
  - The functional organization responsible for the work
  - Base cost estimate for the work
  - Risk and opportunity cost for the work
  - Standard processes or policies and procedures for the work
- The predecessor WBS work packages that must be executed before this WBS work package.

Program WBS creation business process

The program WBS creation business process is as follows:



Author or import program specifications

Author or import requirement specifications

A Request for Proposal (RFP) typically includes the statement of work (SOW), requirement specifications for the product, terms and conditions, and data requirements. You can import and parse the specification documents into individual paragraphs and requirements. Alternatively, you can manually author the requirement specifications with requirements.

Import the requirement specifications from a Microsoft Word document as follows:

1. On the **Home** page, click **Import Specification**.

   The **Import Specification** panel is displayed.

2. Drag the requirement specification Word document to the **Choose File** box.

3. Select the **Type** as **Requirement Specification**, the **Subtype** as **Requirement**, and click **Import**. You can also click **Preview** to verify the requirements before starting the import.

   Select **Run in Background** if you are importing a large amount of data and wish to continue working before the import process is complete.

   IPP&E parses the Word document into a Requirement Specification containing individual requirements.

Create a contract and author a contract data requirement list

To author the contract data requirements list (CDRL) for a program, you must first create a contract. This object acts as a placeholder during the RFP phase. Once the contract is signed, the object represents the actual contract. The CDRL is a collection of individual data requirement items (DRIs).

Create the individual DRIs as follows:

1. Create a contract object with Teamcenter Contract Data Management. The contract provides tabs for referencing the program or project, contract line item numbers (CLINs) and DRIs, and other artifacts related to the management of the contract, such as correspondence.

2. Define an **Integrated Master Plan (IMP)** to ensure that the program events with the event dates exist. These events define the onset and frequency of data delivery for each DRI.

3. Generate a **contract schedule** using the event dates.

4. Create DRIs with Teamcenter Contract Data Management.

Creating an integrated product team

Define an integrated product team

An Integrated Product Team (IPT) consists of representatives of functional areas working together with a team leader to develop the program, identify and resolve issues, and make decisions. They are responsible for the cost, schedule, and technical performance of their program. IPP&E allows you to create an IPT to manage the assignment of individual members to specific IPT functions.

You can build a program IPT by creating a hierarchy of individual IPT elements as follows:

1. Click **Folders**.

   The **Folders Overview** panel is displayed.

2. Click **More commands** ••• > **New** ✳ > **Add** ⊕.

   The **Add** panel is displayed.

3. Select **Integrated Program Team** from the list of types. Use the **Filter** box to find this type if it is not shown by default.

4. Enter a name, optional description, and project for the top-level IPT and then click **Add**. The ID and revision are assigned by the system but you can change them if necessary.

   The top-level IPT is created.

5. Open the top-level IPT and create a child item under it to build the IPT hierarchy.

6. Select the top-level IPT revision from the IPT structure, click the **Overview** tab, and assign a user as the IPT lead in the **IPT Leads** section.

7. Repeat the previous step for the other IPT revision.

Assign an IPT to the WBS root

Assign a top-level IPT to the WBS root as follows:

1. Open the WBS root revision and click the **Overview** tab.

2. Add the top-level IPT in the **Assigned IPT** section.

Assign an IPT to a WBS element and a work package

An IPT can be assigned at any level in the WBS structure. The IPT assigned at a particular level is inherited down to all the lower levels of the structure.

Assign an IPT to a WBS element or a work package as follows:

1. Select the WBS element or work package and click the **IPT** tab.

2. Add the IPT in the **Assigned IPT** section.

Define an integrated master plan

The Integrated Master Plan (IMP) is an event-based plan consisting of a hierarchy of program events. The IPP&E process supports **Events Only** or **Event, Accomplishment, Criteria** schemes defined on the plan object.

An IMP can be defined for a plan (common WBS plan, program, project, subproject). The common WBS plan is created and attached to the common WBS root.

For an IMP with a **milestone breakdown scheme** of **Event, Accomplishment, Criteria**, each event is supported by specific accomplishments. Each accomplishment is associated with specific criteria that must be satisfied for its completion. An event is complete when its supporting accomplishments are complete and when this is evidenced by the satisfaction of the criteria supporting each of those accomplishments.

The IMP is part of the contract and becomes the baseline execution plan for the program or project.

Example:

A portion of an IMP for the Engineering, Manufacturing, and Design (EMD) Phase is as follows:

IMP Activity #	Event Accomplishment Criteria	WBS Work Package that fulfills the criteria
A	Post Award Conference	
A01	Management Planning Reviewed	
A01a	Program Organization Established	1.3
A01b	Manpower Plan Established	1.3

IMP Activity #	Event Accomplishment Criteria	WBS Work Package that fulfills the criteria
A01c	Initial Configuration Management Plan Completed	1.6.2
A01d	Risk Management Process In-Place	1.3
A01e	Program Schedule Reviewed	1.3
A02	Proposed Design Reviewed	
A02a	Proposed Design 3D Models Reviewed	1.1
A02b	Proposed Design Simulations Reviewed	1.1, 1.2
A02c	Proposed Design Manufacturability Assessed	1.1.9
A02d	Proposed Design Supportability Assessed	1.2
B	Integrated Baseline Review	
B01	Integrated Baseline Review Preparation Completed	
B01a	Baseline Schedule Released	1.3
B01b	Baseline Work Packages Updated Based On Contract	1.3
B01c	Baseline Initial Wave of Work Packages Released	1.3
B02	Integrated Baseline Review Completed	
B02a	Integrated Baseline Review Conducted	1.3
B02b	Integrated Baseline Review Minutes Published	1.3
B02c	Integrated Baseline Review Action Items Completed	1.3

You can build a program IMP by creating a hierarchy of individual **Event**, **Accomplishment**, and **Criteria** elements as follows:

1. Open the plan (program, project, or subproject) and click the **IMP** tab. Then, click **Add Event** ⊕ from the work area toolbar of the **Events** section.

The **Add Event** panel is displayed.

2. Select **Event** from the list of types. Use the **Filter** box to find this type if it is not shown by default.

3. Enter details of the event and then click **Add**.

   A new event is created.

4. Repeat steps 1-3 to create additional events.

5. To create an accomplishment for an event, select the event and then click **Add to** ⊕ in the **Accomplishment** section.

   The **Add** panel is displayed.

6. Enter the details of the accomplishment in the **Add** panel and then click **Add**.

   The new accomplishment is created.

7. Repeat steps 5 and 6 to create additional accomplishments.

8. To create criteria for an accomplishment, select the accomplishment and then click **Add to** ⊕ in the **Criteria** section.

   The **Add** panel is displayed.

9. Enter the details of the criteria in the **Add** panel and then click **Add**.

   The new criteria is created.

10. Repeat steps 8 and 9 to create additional criteria.

> Note:
>
> WBS work packages can be related to criteria or events, depending on the IMP scheme (**Event, Accomplishment, Criteria** or **Event only**).

Generating the contract schedule from plan

Generate the contract schedule

The Contract Data Requirements List (CDRL) is a list of data requirement items (DRIs). The delivery schedule of the data described in each DRI depends on specific contract events or milestones. For each DRI, an event table defines the frequency and offset of data delivery, and it is based on the events in the contract schedule. The DRI's event table is used to calculate the dates and frequency of contract

deliverables. IPP&E allows you to generate the contract schedule from the program IMP. Typically, contracts are associated with a program, project, or subproject.

Create the contract schedule as follows:

1. Open the contract associated with the program, project, or subproject, and then click the **Schedule** tab.

2. Click **More commands** ••• > **New** > **Create Contract Schedule**. Alternatively, click **Add** ⊕ from the work area toolbar in the **Contract Event Schedule** section.

   The **Add Schedule** panel is displayed.

3. Select a schedule template, enter an optional description and project, and then click **Add**. The schedule name is generated by the system but you can change it if necessary.

   The system creates a new schedule in the **Contract Event Schedule** table. The planned contract start date and a contract end date are populated from the chosen template, as are the schedule tasks representing each program event.

Update the contract schedule

If a new event is added or event dates are changed, you must update the contract schedule. You should add a new schedule task for each new event or update the dates of the existing events.

1. Open the contract, select the schedule, click **More commands** ••• > **Edit** → **Update Schedule**.

   The **Update Contract Schedule** panel is displayed.

2. Change the schedule dates and then click **Update**.

   The system updates the contract schedule with your changes.

> Note:
>
> You can also click the **Overview** tab, and click **More commands** ••• > **Edit** > **Shift Schedule**. The **Shift Schedule** panel is displayed. This panel allows you to move the schedule backward or forward by a selected number of days.

Deleting an event

IPP&E allows you to delete events even if any of the following conditions apply:

- If the event has an accomplishment associated with it.
- If the schedule tasks are already created for the event in the contract schedule.

- If the program WBS containing the event has already been sent to the external schedule.

IPP&E maintains a record of event deletion so that when the data is exported from Teamcenter to the external schedule, the external schedule deletes the corresponding milestone. If a contract schedule task was created for the event and the task is not referenced in a DRI event table, the schedule task is also deleted.

If accomplishments are associated with the event, cut the relation between the event and the accomplishment to delete the event.

If a contract schedule task was created from the event and the task is not referenced in a DRI event table, delete all the references to the task from the DRI event table to delete the event.

To delete an event:

1. Select the event and click **More commands** ••• > **Edit** > **Delete** .

2. Confirm deletion of the event.

Change event dates in the external schedule

If the program WBS data was already exported to the external schedule from Teamcenter, and you must change event dates or program start date, you should change the dates in the external schedule.

After changing the dates of events in the external schedule, send the updated information back to Teamcenter. This action updates the corresponding events' planned dates in Teamcenter. If the contract schedule was already generated, the system automatically updates the dates of the corresponding tasks.

If the event tasks are referenced in any DRI event table, you must reschedule the impacted DRIs, as follows:

1. Select one or more DRIs in the **DRIs for Reschedule** panel, and choose **Reschedule Data Requirement Items** .

   The **Reschedule Data Requirement Items** panel is displayed.

2. Update the schedules of the selected DRIs and then click **Reschedule**.

Planning the WBS

Manage the WBS

Different programs have different levels of complexity. Complex programs require a WBS to manage the deliverables and schedule, while simpler programs may not require a WBS. During the creation of the plan, you can specify whether the plan is supposed to be managed through a WBS. If you do this, the **Manage WBS** property is set on the plan, program, project, and sub projects, allowing the creation of

the WBS root element (the top element of the WBS structure) as well as the plan object. The system associates the WBS root element with the plan object and the two objects have the same name.

You can set properties on the plan and WBS root such that they behave differently.

Set the following properties on the WBS root:

WBS Number Required

Makes the WBS number on the WBS structure optional or required.

> Caution:
>
> Once set, this property cannot be changed.

WBS Number Separator

Specifies the separator for the WBS number, for example, 1.1.1 or 1-1-1 or 1,1,1.

If not specified, the default is a **.** (period).

Set the following property on the plan:

Milestone Breakdown Scheme

Specifies whether to break down the milestones under the plan by **Events only** or by **Event, Accomplishment, Criteria**.

When set to **Event, Accomplishment, Criteria**, the plan consists of a hierarchy of events supported by accomplishments associated with specific criteria to be satisfied for the event's completion.

Break down the plan

A program can be broken down into multiple projects and a project may have multiple subprojects. By default, IPP&E allows users to associate the WBS structure with any or all levels of the plan breakdown.

To restrict the plan breakdown if the WBS is associated with the plan, the administrator can update the **IPP_disallow_plan_breakdown_if_wbs_related** preference.

To restrict the association of the WBS to the lowest level of plan breakdown, the administrator can update the **IPP_disallow_plan_wbs_relation_if_plan_has_breakdown** preference.

Create the program WBS from the common WBS

The program WBS provides the framework for specifying the work necessary to achieve the objectives of a program. The levels of the program WBS are related to the system requirements and conform to the product-oriented family tree. The detailed technical objectives are defined, and the scope of work is determined for each WBS element. Work packages are assigned to each WBS work package. The resources, materials, and processes required for attaining the objectives of each work package are added

incrementally. This process allows all items to be traced to the same WBS element. The linkage between the requirements specification, the WBS, the integrated master schedule (IMS), and the integrated master plan (IMP) provides insights into the relationship between cost, schedule, and performance.

The program WBS can be authored by cloning a configured version of the common WBS, using a set of characteristics applicable for that program. After the program WBS is created, you can update it to match the needs of the program without affecting the common WBS library.

The terms *Program WBS* and *Project WBS* are used interchangeably and are represented by the same object. The business object for the top node of the program WBS is the **WBS Root Element**.

To create the program WBS from the common WBS, do the following:

1. Open the program, click the **Overview** tab, and then click **Add To** ⊕ from the work area toolbar of the **Work Breakdown Structure** section.

   The **Add WBS Root Element** panel is displayed.

2. Enter the name and optional description of the WBS root element and then click **Add**. The system generates the ID and revision but you can modify these if required.

   Select **Common WBS** if you want to create a common WBS to be used as a template for several WBSs.

   The WBS is based on a program OBS, which you select from the list.

   The system creates the new WBS root element.

3. Use the **Advanced** search to find the common WBS. A pre-configured query called **Find WBS Root** is provided for this.

4. Open the common WBS and choose **Configure** from the results panel toolbar to apply configuration filters, for example, a variant rule.

   When applying the variant rule, clear the **Show Excluded By Variants** option under the **Configure** command.

   The configured WBS is displayed.

5. Choose **Clone WBS Root** from the results panel toolbar.

   The **Clone WBS Root** panel is displayed.

6. In the results panel showing the structure, select any WBS elements you do not want to copy to the project WBS, right-click, and choose **Remove** ⊖.

If you remove an element by mistake, right-click and choose **Undo**.

7. After removing the unnecessary elements, click **Clone**.

8. Make the following selections in the **Clone WBS Root** panel and then click **Clone**.

   - In the **Target WBS Root** field, select the required program WBS root revision from the list. The list consists of all the WBS root revisions containing no structure and for which you have write access.

   - If applicable, select **Check for Duplicate WBS Number**. If selected, IPP&E validates that the configured common WBS that represents the program WBS does not contain more than one WBS element with the same WBS number. This selection is the default for creating the program WBS.

   - Select **Run in Background** to run the program WBS creation process in the background while you continue working. If selected, you must perform the **Do Task** in your worklist to complete the cloning process.

   - If applicable, select **Create Program OBS**. This creates a new program OBS that is then set on the target WBS root. If not selected, the original program OBS value from the source WBS root is retained on the target WBS root.

   - In the **Related Structure** section, you can view all the top nodes of the structures (for example, requirements, physical, and logical) related to the common WBS root. Choose one of these cloning options for the related structures:

     - **Clone**

       All the common WBS related structures relevant to the program WBS are cloned. Links corresponding to the common WBS links are created between the program WBS elements and the corresponding nodes in the cloned related structures.

     - **Ignore**

       Allows you to ignore the related data. This option removes the links between WBS elements and the corresponding nodes in the selected related structures.

     - **Reference**

       Allows you to reference the existing related data in the target WBS structure WBS elements.

     - **Reuse Existing Structure to relate**

       Allows you to select an existing related structure from a list of available structures. Use when the related structures were cloned before creating the program WBS. Links corresponding to the common WBS links are created between the program WBS elements and the corresponding nodes in the selected related structures.

IPP&E creates the program WBS. You can then view the program WBS, the related IMP, and other structures.

Import the WBS structure from Microsoft Excel

You can define the WBS hierarchy in Microsoft Excel and then import the file into Teamcenter. IPP&E includes a sample Excel file that provides the required format for the spreadsheet.

> Note:
>
> The last row of the file must end with **<endtag>** as follows:

2	Wbs0Element	Sustainment/Interim Contractor Support	1.9.5	False		480
1	Wbs0Element	Industrial Facilities	1.10	False		480
2	Wbs0Element	Construction/Conversion/Expansion	1.10.1	False		480
2	Wbs0Element	Equipment Acquisition or Modernization	1.10.2	False		480
2	Wbs0Element	Maintenance (Industrial Facilities)	1.10.3	False		480
1	Wbs0Element	Initial Spares and Repair Parts	1.11	True		9600
<endtag>						

The import process parses all the columns in the Excel file that can be mapped to Teamcenter properties, in numerical order.

1. Create an Excel file containing the WBS hierarchy, based on the sample file and the example above.

2. To use a WBS number separator rather than a period (.), update the **WBS_DefaultSeparatorForWBSNumber** preference. By default, the preference value is **1**, which results in a period separator.

3. From your **Home** page, click the **Import Specification** tile.

   The system displays the **Import Specification** panel.

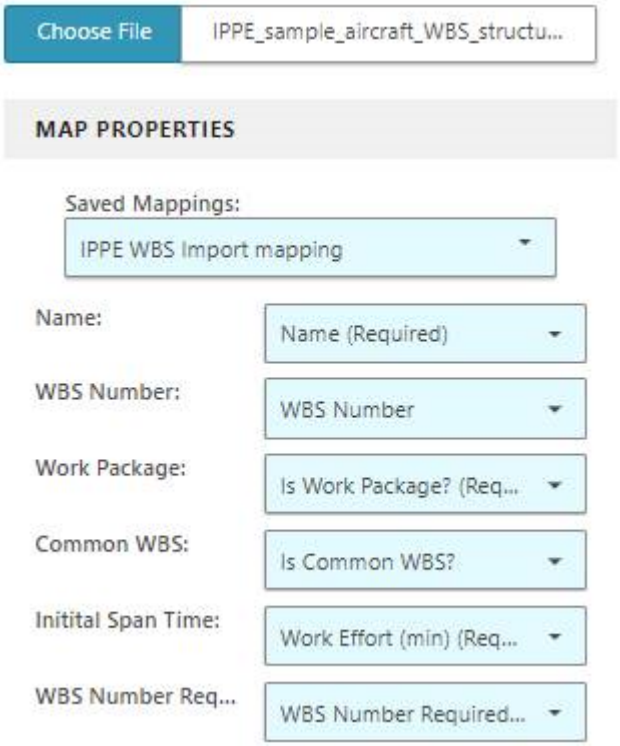
4. Click **Choose File** and then browse to and select the Excel file containing the WBS hierarchy.

5. Select a saved property mapping or choose the properties to map.

   For example, your Excel spreadsheet may contain the following columns:

Tc_Level	Tc_ObjectType	Name	WBS Number	Work Package	Common WBS	Initital Span Time	WBS Number Required
0	Wbs0Root	Demo Common Aircraft System	1		True		True
1	Wbs0Element	Air Vehicle	1.1	False		480	
2	Wbs0Element	Propulsion	1.1.2	False		480	
3	Wbs0Element	Engine Inlet	1.1.2.1	False		480	
4	Wbs0Element	Create Engine Inlet Design	1.1.2.1.1	False		480	
4	Wbs0Element	Conduct Engine Inlet Analysis	1.1.2.1.2	False		480	
3	Wbs0Element	Engine Compressor Section	1.1.2.2	False		480	
4	Wbs0Element	Engine Compressor First Stage	1.1.2.2.1	False		480	
5	Wbs0Element	Create Engine Compressor First Stage Design	1.1.2.2.1.1	False		480	
5	Wbs0Element	Conduct Engine Compressor First Stage Analysis	1.1.2.2.1.2	False		480	

You can map the properties in the spreadsheet to the properties in Teamcenter by either choosing a **Saved Mapping** from the list or by mapping each property individually as shown.



6. Click **Import**.

   The system imports the WBS hierarchy and displays it in Teamcenter.

Add new WBS elements to the WBS

As you develop the WBS for your program, you can add new WBS elements to handle new specifications, SOW paragraphs or terms and conditions that did not exist in the common WBS. You can add a WBS element at any level of the WBS. Each WBS element is identified by a property, the WBS number, that follows a standard format. A typical practice in the aerospace and defense industry is to use a period-separated version number such as 1.1.1.1, where each number represents a specific level.

The lowest level WBS element in the WBS hierarchy contains the details of the actual task and is called the WBS work package. The upper level WBS elements define the hierarchical organization of the tasks for the program.

A WBS work package can be a recurring work package for tasks that need to be repeated at a certain interval, such as the creation of a program status report.

Add a new WBS element to the program WBS as follows:

1. Open the WBS hierarchy, select the parent element under which the new child WBS element must be added, and choose **Add** ⊕ → **Child** from the results panel toolbar.

   The **Add** panel is displayed for adding a WBS element.

2. Enter the required properties for the new WBS element and click **Add**.

   The system adds the new element to the WBS.

Clone WBS elements

To clone the WBS element hierarchy within a WBS or across two different WBSs, do the following:

1. Copy the WBS root revision.

2. Select the WBS element to clone and choose **Clone WBS Element** from the results panel toolbar above the WBS structure tree.

3. In the results panel with the structure, select any WBS elements you do not want to copy to the project WBS, right-click, and choose **Remove** ⊖.

   If you remove an element by mistake, right-click, and choose **Undo**.

4. After removing the unnecessary elements, click **Clone**.

   The **Clone WBS Element** panel is displayed.

5. Click **Add** ⊕, select the copied WBS root from the **Clipboard** section, and click **Add**.

6. Enter a new WBS number for the cloned hierarchy if required.

7. Select **Run in Background** to run the program WBS creation process in the background while you continue working. If selected, you must perform the **Do Task** in your worklist to complete the cloning process.

8. Click **Clone**.

   The system creates the cloned WBS element hierarchy in your **Newstuff** folder.

Link data to the program WBS

Once you have added the WBS elements to represent a requirement, SOW paragraph, or terms and conditions paragraph, record the traceability between the specifications and the corresponding WBS elements. To do this, you create a link between the two objects. Once all the specifications are mapped to the WBS, you can generate a report to ensure that all the work needed to fulfill the requirements,

SOW, and terms and conditions are added in the WBS. Creating links in this way also makes it easier to identify WBS elements that are potentially affected by (for example) changes to requirements.

1. Select a WBS element.

2. Click the appropriate tab in the work area to relate various objects.

   For example, click the **Related Data** tab to relate requirements and program deliverables. Click the **Attachments** tab to relate files and documents.

3. Click **Add to** ⊕ from the work area toolbar in the appropriate section.

   For example, from the **Related Data** tab, click **Add to** ⊕ from the work area toolbar of the **Program Deliverables** section.

   The **Add** panel is displayed.

4. You can link data from the **Add** panel in several ways:

   - Create a new item from the **New** tab.
   - Use the **Palette** tab to relate an existing item copied to the clipboard, added as a favorite, or recently opened.
   - Use the **Search** tab to find an existing item using a search box and filters.

   Click **Add**.

   The system creates a link between the WBS element and the item selected in the **Add** panel.

Update the WBS number

The WBS number is a property on all the elements of the WBS hierarchy. Numbering in this way provides a consistent approach to identifying and managing the WBS across systems, regardless of vendor or service.

IPP&E allows you to update the WBS number of the WBS elements. You can select any element in the WBS hierarchy and update the WBS number. The system updates the WBS number of the selected element and all the child elements in its hierarchy.

The ability to update the WBS number depends on whether a WBS number is required or not. If you cannot update the WBS number because of number pattern validation, you can cut the WBS element from the structure and then update it. The updated elements can be added back into the structure later.

By default, the WBS number can be updated if there is only one revision for all the elements in the selected WBS Element hierarchy. If there are multiple revisions, the system does

not allow you to change the WBS number. You can change this behavior by updating the **IPP_update_wbs_number_for_multiple_revision** preference.

To update the WBS number of a selected element in the WBS structure, click **More commands** ••• > **Edit** > **Update WBS Number** and then edit the number in the **Update WBS Number** panel.

WBS Number as an optional value

Some projects and WBSs do not mandate WBS numbers or may not follow a fixed WBS numbering pattern. IPP&E allows you to make the WBS number optional for the WBS hierarchy. The user can provide the WBS number when creating the root or when creating the plan (program, project, subproject, and common WBS plan).

Depending on whether a WBS number is required, the system performs or skips certain validations. The following validations are performed if a WBS number is required:

- WBS elements are not created without a WBS number.
- When a child WBS node is added to the WBS hierarchy, the system validates if the WBS number pattern matches the parent's number pattern.
- The WBS number pattern is validated whenever the WBS number is updated.
- A WBS element with a duplicated WBS number is not allowed in the project WBS.

The following validations are skipped if a WBS number is not required:

- WBS nodes can be created without a WBS number.
- The WBS number pattern is not checked when a child node is added to the WBS hierarchy.
- The WBS number pattern is not validated when the WBS number is updated,
- The WBS number is not checked for duplication if the user enters the WBS number manually.

The lessons learned process compares two WBS structures to identify if any new elements are added or deleted, or if cost data is updated. By default, this comparison uses the WBS number property. If the WBS number is optional, the administrator can choose an alternative property for the comparison by updating the value of the **Awb0EnableDynamicEquivalenceForCompare** preference.

Using the WBS Number Separator

In general, the WBS number follows a pattern where the WBS number of a child element is the combination of a parent WBS number and a suffix, with a separator in between. IPP&E allows the user to select one of the following WBS number separators:

1. “.” (Period)
2. “-“ (Hyphen)
3. “,” (Comma)

The user can select the WBS number separator when creating the WBS root or the plan. The selected separator is applied to all the elements in the WBS hierarchy.

Managing work packages

Identify work packages for criteria

You define details of the required work at the lowest level of the WBS work package. These work packages eventually map to a schedule task in the program schedule. The sequencing of these tasks is defined by creating predecessor relationships between the corresponding work packages. The outputs from the predecessor work packages become an input to that work package. The last work package in a predecessor-successor chain is linked to a criteria.

Assign a work package to its criteria as follows:

1. Select the WBS root and click the **IMP** tab.
2. In the **Event** table, select the appropriate event.

   The **Accomplishment** table is populated.
3. In the **Accomplishment** table, select the appropriate accomplishment.

   The **Criteria** table is populated.
4. In the **Criteria** table, select the criteria to which the work package is to be assigned.
5. From the work area toolbar of the **Work Package** section, click **Add to** ⊕.
6. Use the **Add** panel to select the appropriate work package and click **Add**.

   The system assigns the work package to its criteria.

Identify predecessors for work packages

Link the work package as a predecessor of another work package as follows:

1. In the WBS hierarchy, select the predecessor work package, and click **Copy** .

The system copies the work package to the clipboard.

2. Select the successor work package, click the **Network** tab, and click **Paste** from the work area toolbar of the **WBS Predecessors** section.

   The system creates a link between the work packages.

Defining work package recurrence

The **Recurrence** property on the work package defines how many tasks are created for the work package when data is sent to the external schedule to generate a schedule. It may have one of the following values:

Single	Only one task is created for the work package. (Default value.)
Multiple	More than one task may be created for a work package. The number of tasks created depends on the value of the **Number of Task** property.
Recurring	This value is available only if Contract Data Management is installed. You can create multiple tasks for a work package. The number of tasks created or dates of the recurring tasks depends on the event table of associated DRI. Predecessor-successor links are not allowed for work packages of the recurring type.

Convert a work package to a WBS element

Work packages are the lowest level elements in the WBS hierarchy. The hierarchy contains work packages for different functional organizations. Work packages are the actual tasks tracked in the schedule, whereas all the intermediate elements between the WBS root and work packages are called *WBS elements*.

You may want to further break work packages into multiple work elements. To do this, you should first convert each such work package to a WBS element.

Select the work package to convert and then choose **Edit** → **Convert to WBS Element**.

The system performs the following checks and, if successful, it completes the conversion.

- There are no FWPs attached to the work package.
- There are no risks or opportunities attached to the work package.
- There are no pending FWP requests for the work package.
- The work package does not have any predecessor or successor relations.
- The work package is not attached to any criteria.

Convert a WBS element to a work package

Convert a selected WBS element to a work package as necessary by choosing **Edit** → **Convert to Work Package**.

The system checks that the WBS element does not have any other WBS elements under it and if not, completes the conversion.

Authoring the WBS element outside of the WBS structure

IPP&E supports the bottom-up authoring of the WBS structure. Users can create a WBS element or work package independently (for example, in a folder or under the event or criteria) and then add the element back to WBS structure under the WBS root or the WBS element. If creating the object in a folder, the system searches for the WBS root from the WBS root property; in the case of a criteria, it is automatically filled.

Creating an organizational breakdown structure hierarchy

Understanding the organizational breakdown structure hierarchy

The organizational breakdown structure (OBS) describes the organizational structure and how the organizations relate to each other. It is necessary for the successful integration of a project's scope, schedule, and budget. Individuals are assigned to work on program tasks according to a function, for example, engineering design or information services. Individuals within these groups can be assigned to support different programs but they remain part of the functional organization. The OBS elements are assigned to specific WBS work packages to ensure that all personnel responsible for planning, managing, and controlling the project are identified.

Create an OBS

As an HR lead, you can create an OBS and assign a user as a leader of each functional organization node, as follows:

1. Open any folder and click **More commands** ••• > **New** > **Add**.

   The **Add** panel is displayed.

2. Select **OBS Element** as the object type, and the panel refreshes with the fields for defining an OBS element. The ID and revision are pre-populated according to the scheme defined by your administrator.

   Enter a name, optional description, and department number for the functional organization, and then click **Add**.

Optionally, select **Is OBS Root** to create the OBS root element. You can create only one OBS root element.

The system creates the OBS root element.

3. Select the OBS root element and create child OBS elements by choosing **Add** ⊕ from the results panel toolbar above the structure.

   When creating child elements, ensure **Is OBS Root** is *not* checked.

4. Select the first OBS element and click the **Overview** tab. In the **Functional Organization Lead** section, click **Add** ⊕ from the work area toolbar.

   The **Add** panel for the functional organization lead is displayed.

5. In the **Users** section, select the required functional lead and then click **Add**. You can only assign an existing user and not create a new one. You can use the **Group** and **Role** lists to filter the list of users.

   The system assigns the selected user as the functional lead for the first OBS element.

6. Repeat steps 4 and 5 for each OBS element.

7. To designate the final approving functional organization, select the corresponding OBS element in the OBS hierarchy and then click the **Overview** tab. Choose **Edit** ✎ → **Start Edit** and then select **Final Approving Functional Organization**.

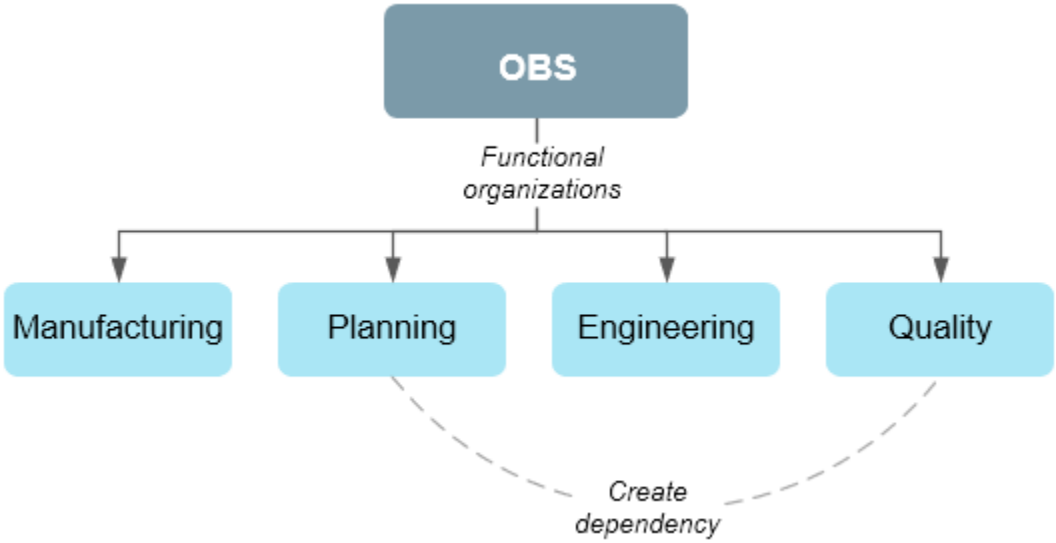
   The system designates this organization as able to approve cost estimates during the planning phase and approval of deliverables of the execution phase.

8. To designate a functional organization that does not play any part in the planning and execution of a program, select the corresponding OBS elements in the OBS hierarchy and then click the **Overview** tab. Choose **Edit** ✎ → **Start Edit** and then clear **Visible in RAM**.

   The system excludes the selected organizations from the Responsibility Assignment Matrix (RAM), thereby reducing clutter.

Create a dependency between OBS nodes

IPP&E allows you to indicate the cost estimate for a functional organization based on the cost estimate of another functional organization for the same WBS work package. Do this by creating a dependency between the two functional organizations in the OBS. The following diagram shows dependencies between the functional organizations in the OBS:



Here, the Quality functional organization depends on the Planning organization. Consequently, the cost estimate of the quality work package is a factor in the cost estimate of the planning work package.

Create an OBS dependency as follows:

1. Open the OBS element or select it in the OBS hierarchy in which the OBS element is dependent (for example, the Planning OBS), and click **Copy**.

2. Open the OBS element or select it in the OBS hierarchy in which it depends on others (for example, the Quality OBS), and click **Paste** from the work area toolbar in the **Depend On** section of the **Related Data** tab.

   The system relates the two OBS elements.

3. To add a dependency factor percentage, select the dependent OBS element in the **Related Data** tab of the depending OBS, choose **Edit** → **Start Edit**, edit the value of the factor in the **Depend On** section, and then click **Save**.

Viewing functional work package dependencies

When you create functional work package (FWPs) under any work package, the system creates a dependency between FWPs according to the OBS dependency and the FWP dependency during the FWP estimation process. When a work package is submitted for estimation, the estimation process begins with only the FWPs that are not dependent on any other FWP. Once these FWPs are estimated and approved, their estimates can be utilized by their dependent FWPs. The estimation process for the dependent FWPs subsequently starts.

View dependencies in the **Depend On** and **Dependent Functional Work Packages** sections of the **Related Data** tab. The **Estimates** tab allows you to view rolled up estimates and calculated cost basis.

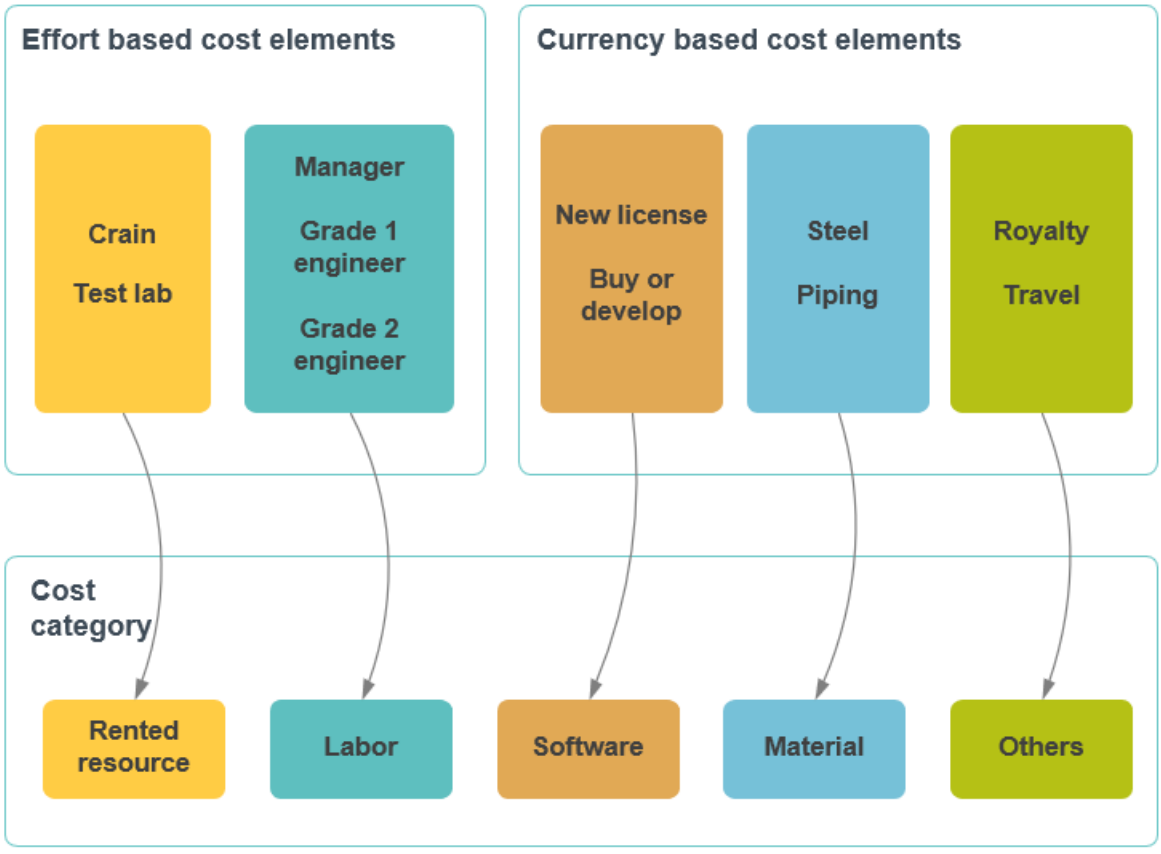
For example, if the Quality FWP depends on Manufacturing FWP and the Quality FWP is estimated, you can view the calculated cost based on a factor of the Manufacturing FWP on the direct cost of the Quality FWP. This assists you in estimating the Quality FWP.

Defining valid cost elements for an OBS

Defining valid cost elements for an OBS

IPP&E allows you to estimate the labor, material, and other costs for the work packages in a WBS work package. The different costs are estimated as separate cost elements for each functional work package. These cost elements are created for two broad groupings of cost categories. The first is the effort-based cost categories, which account for labor and other resources, and are estimated based on time. The second is the currency-based cost categories, which are estimated using currency values.

The following diagram shows examples of cost categories and cost elements created in these categories:



Create cost categories

Before creating a *Cost Element*, you must add the necessary *Cost Categories* as follows:

1. Open any folder and click **New** → **Add** ⊕.

The **Add** panel is displayed.

2. Select **Cost Category** as the object type, and the panel refreshes with the fields for defining a cost category.

   Enter a name and optional description, select a **Default UOM** (unit of measure), and then click **Add**.

   If you select **Effort Based**, select a time-based UOM. If you do not select **Effort Based**, select a currency-based UOM, for example, dollars.

   The system creates the cost category.

Create disciplines for effort-based cost elements

A *Discipline* contains sets of users with a common behavior. IPP&E uses disciplines to define effort-based or labor-based cost categories such as Manager, Senior Manager, Engineer, and QA as follows:

1. In the Teamcenter rich client, open the Organization application and select **Disciplines**.

2. Enter the following values in the right-hand panel:

   - **Name**

     The name of the labor cost element.

   - **Default rate**

     The rate for that discipline. This can be set to **zero** if it is not applicable for your business processes.

   - **Default Currency**

     The currency for the user locale.

Create currency cost elements

Once you have defined cost categories, you can create currency cost elements associated with the categories.

1. Open any folder and click **More commands** ••• > **New** ✳ > **Add** ⊕.

   The **Add** panel is displayed.

2. Select **Currency Cost Element** as the object type, and the panel refreshes with the fields for defining a currency cost element.

Enter a name and optional description, select a **Cost Category**, and then click **Add**.

The system creates the currency cost element.

Create effort cost elements

Once you have defined cost categories and cost elements, you can create effort cost elements within the categories.

1. Open any folder and click **New** → **Add** ⊕.

   The **Add** panel is displayed.

2. Select **Effort Cost Element** as the object type, and the panel refreshes with the fields for defining an effort cost element.

   Select a **Cost Category** and **Cost Element**, enter an optional description, and then click **Add**. The **Cost Element ID** is generated by the system, according to a scheme defined by your administrator.

   OBS Controlled Applicability determines whether the effort cost element applies to all OBS elements or only selected elements.

   The system creates the effort cost element.

Assign cost elements to OBS nodes

You can assign specific effort-based cost elements to functional organizations. When you do this, the functional organization can only enter estimates for the assigned cost elements in a work package.

To assign cost elements to a functional organization in the OBS, do the following:

1. Search for the effort-based cost element and copy it to the clipboard.

2. Open the OBS element, click the **Related Data** tab, and then click **Paste** from the work area toolbar in the **Applicable Cost Elements** section.

   The system assigns the effort-based cost element to the OBS element.

Create a program OBS for a program WBS

The OBS hierarchy represents the functional organization hierarchy of the entire enterprise. However, for a specific program, not all function organizations from the OBS hierarchy may be relevant. Use the Product Configurator in the rich client to define a configurator context for applying variability to the OBS hierarchy and assigning it to the program OBS. The program OBS allows you to exclude some functional organizations in the context of a program.

1. Create a program OBS and assign the configurator context to the program OBS item (not item revision).

2. In the Active Workspace, create parent-child relations between the program OBS and the OBS root.

3. Optionally, exclude a selected OBS branch from the context of the program OBS by choosing **Toggle Suppression** from the results panel toolbar.

   This command sets the variant condition on the selected OBS element, in the context of the top program OBS element. It excludes the selected OBS element branch from the RAM. You can subsequently choose **Toggle suppression** again to add the selected OBS element branch back into the RAM.

   The system uses the variant rule defined in the **IPP_variant_rule_for_obs_configuration** preference to configure the program OBS. The OBS elements that are configured are shown in the RAM.

   The RAM dialog box for the program displays only the relevant functional organizations of the program OBS.

Managing changes to the OBS

Managing changes to the OBS

The Organizational Breakdown Structure (OBS) of a company may change over the life of a program. These changes may include modifications to the functional organization hierarchy, personnel changes, or the addition or elimination of levels in the organization. IPP&E ensures that functional work package estimation and execution are performed and approved by the most current functional organization. It is a best practice to keep the OBS structure released and to create a new revision to make any changes.

Changing the functional organization leader

IPP&E ensures that only the current functional organization lead performs department level approval of estimates and the execution of a functional work package. If the leader of a functional organization is changed during the estimation or execution process of a functional work package, IPP&E automatically routes that package to the new leader for the functional organization

Changing the functional organization hierarchy

If levels are added to or removed from the functional organization, IPP&E automatically routes the functional work package approvals to the new hierarchy. If a functional organization is removed from the OBS hierarchy, the functional work packages created prior to those changes still carry the outdated OBS information. When you release the OBS structure, IPP&E identifies the FWPs assigned to removed functional organizations and sends an assignment to the leader of the removed functional organization for further action.

Manually update the functional organization for an FWP

If you have write access to the FWP, you can reassign it to a new functional organization as follows:

1. Select the FWP and then click **More commands** ••• > **Edit** > **Update OBS Information**.

   The **Update OBS Information** panel is displayed.

2. Choose a new functional organization from the **Applicable Functional Organization** list and then click **Update**.

   The system reassigns the FWP to the new functional organization.

Creating functional work packages

Create functional work packages for the WBS using the RAM

Each WBS work package includes work packages for each function responsible for completing the task. A functional work package (FWP) contains the following information:

- Responsible functional organization
- Cost estimates to perform the work
- Deliverables, including scope of work
- Related data, for example, policies and procedures documents

Create functional work packages for a WBS work package as follows:

1. Select one or multiple WBS work packages and then click **RAM**.

   The RAM that is used to identify the intersection of WBS work elements is displayed along with the organization breakdown elements for which a functional work package is to be created. The columns of the matrix show the WBS work packages you selected. Each row corresponds to an organization to which you can assign work.

   If you created the WBS from the **common WBS**, the organizations are copied from those on the common WBS.

2. Click (check) one or more cells in the matrix, and choose **Add** at the top of the matrix to create FWPs.

> Tip:
>
> Easily locate the organizations of interest in the following ways:
>
> - Filter the OBS list by clicking in the table header and using the resulting **Search** box.
> - Use the arrows to expand and collapse the hierarchical OBS structure.

The system assigns each WBS work package to one or more assignees.

Create functional work packages by adding a child

As a work package owner, you can add a functional work package (FWP) as a child object. You can then select a functional organization.

Create a FWP child as follows:

1. Open the WBS root and select the parent WBS work package for which the FWP child is to be created.
2. Choose **Add** ⊕ → **Child** from the results panel toolbar.

   The **Add** panel is displayed.
3. Enter a **Name** for the FWP, choose the **Functional Organization**, and click **Add**.

   The system creates the FWP under the selected WBS work package.

> Note:
>
> If the **IPP_multiple_fwp_for_obs** preference is set to **true**, then you can create multiple functional work packages with the same functional organization.

Create functional work packages using an FWP request

As a functional organization leader, if you determine that a WBS work package requires work from your organization, you can also request the creation of an FWP for that WBS work package. This FWP request can then be submitted to the program manager or IPT lead for approval. Once the FWP request is approved, the system creates the FWP for the WBS work package.

Create a functional work package request as follows:

1. Open the WBS root and select the WBS work package for which the FWP request is to be created.

2. Click the **Resource Package Request** tab.

3. Click **Package Request** from the workarea toolbar of the **Resource Package Requests** section.

   The system displays the **Resource Package Request** panel.

4. Select the OBS element and requested resource package type, enter a justification, and then click **Create Resource Package Request**.

   The system submits the FWP request to the work package owner for approval. On approval, the FWP is created under the WBS work package.

Creating a functional work package without an OBS element

You can use the OBS hierarchy for various purposes, for example, hierarchical approval of estimates and grouping of resource types that belong to the same functional organization. If you do not want to use these capabilities for your program but do want to perform resource estimation, you can create the functional work package (FWP) without OBS elements. To do this, select the work package and then choose **Add** ⊕ to add the necessary functional work package types.

Creating a work package without an FWP

It is only necessary to create FWPs under a work package if you want to capture the resource estimates for the work package. The work package itself contains enough information to create a task. It contains the span time and predecessor-successor link. You can create and execute the tasks for work packages even if they do not have an FWP or resource estimates defined. The system uses the span time as the task duration and the predecessor-successor link as a constraint in the schedule. If there is no FWP under a work package, you can directly attach the deliverables to the work packages.

Managing risks and opportunities

Understanding risk management

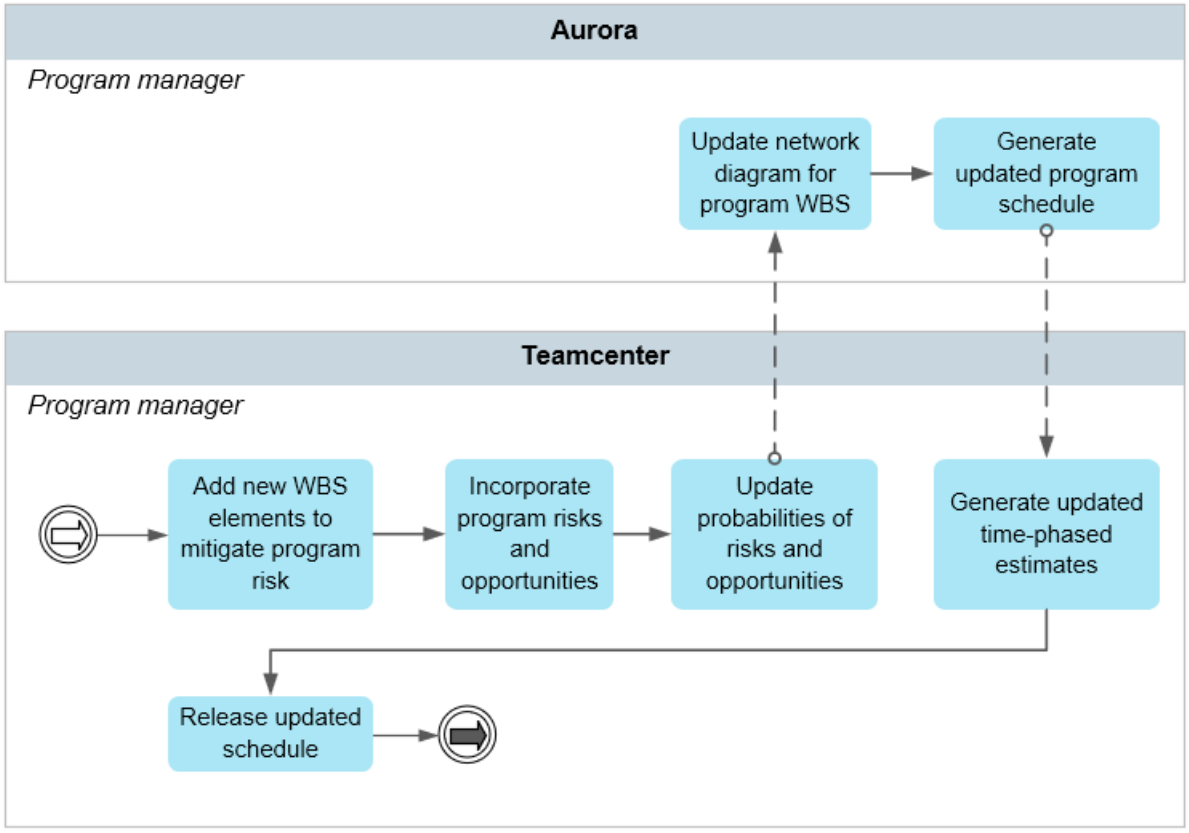
One of the key aspects of project management is the identification, assessment, and development of plans to respond to any uncertain event or condition that, if it occurs, has a positive or a negative effect on the project objective. These events may impact the time, cost, scope, or quality of the project deliverables. Any uncertain event that can cause a negative impact is called a *risk*, while an event that impacts a project in a positive manner is an *opportunity*.

During the program planning phase, the risks and opportunities impacting various tasks in the project plan are identified, assessed for costs and schedule impact and identification, assessment, and development of response plans. During the execution phase, the risk management process consists of monitoring and control of program risks and opportunities. It includes identifying, analyzing, and planning for newly arising risks and the tracking, reanalyzing, and mitigating of already identified risks. In addition, the execution of risk responses is also reviewed to evaluate their effectiveness during risk monitoring and control.

As a program manager, during the execution phase of the risk management process, you can add new WBS elements in the program WBS to mitigate risks. These mitigation WBS elements can be estimated and approved. You can identify new risks and opportunities specifically for the program. If a risk or an opportunity (event) arises, you can incorporate its cost and schedule impact in the program schedule. You can then send the updated program WBS to the external schedule to update the network diagram and generate the updated schedule. All these activities can be performed without affecting the in-work tasks.

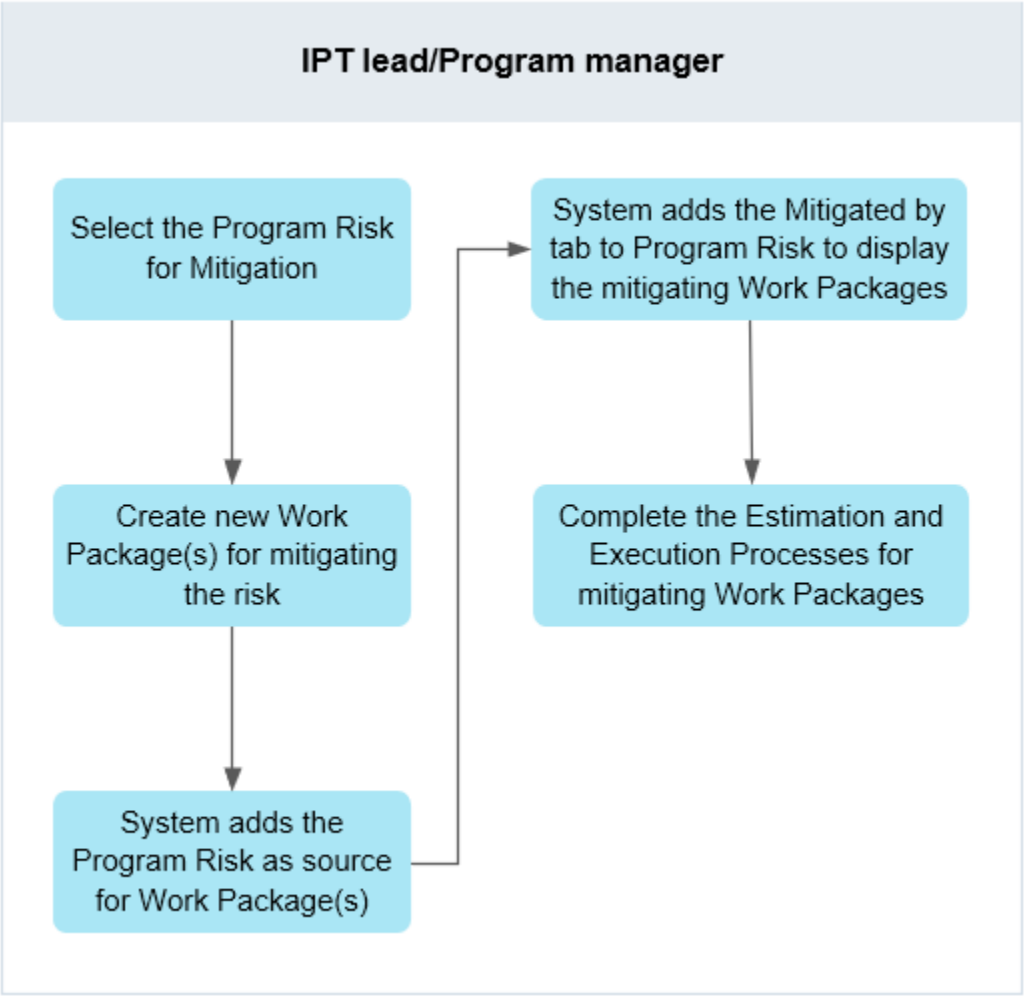
Risk management business process

The risk management business process is as follows:



Mitigate program risks

Risk mitigation is performed to reduce the extent of exposure to a risk and the likelihood of its occurrence. You can create new WBS work packages in the program WBS to mitigate a program risk. You link the newly created WBS elements to the risk and set the disposition of the risk as *mitigated* by submitting the risk to the **IPPE Risk Mitigation Process** workflow. You can then submit the new WBS work packages to the estimation workflow using the **program cost estimation initiation** process.



Once the risk is submitted to the **IPPE Risk Mitigation Process** workflow, as an IPT lead or program manager, mitigate a program risk as follows:

1. Create risk mitigation work packages as **new WBS elements in the program WBS**.

   The risk mitigation process links the newly created risk mitigation work packages with the mitigated program risk.

2. Open the newly created risk mitigation work package, click the **Overview** tab, and verify that the **Source** property is populated with the risk identifier.

3. Open the program risk, click the **Mitigated By** tab, and verify that the newly created risk mitigation work package is displayed in the table.

Incorporating program risks and opportunities

IPP&E incorporates risk and opportunity elements when you accept a program risk or opportunity and then add the cost to the direct cost and schedule impact to your program schedule. You then set the risk or opportunity disposition as *Incorporated* by submitting the risk to the **IPPE Risk Incorporation Process** workflow.

Update the probability of program risks and opportunities

As a program manager, you may decide during the program execution phase that the probability of a risk or an opportunity has changed. As an IPT lead or program manager, you can update the probability of any risk or opportunity for the program without impacting the other programs.

1. Select a program risk or opportunity, and click **More commands** ••• > **Edit** > **Start Edit**.

2. Update the values of the **Probability** and **Span Time** for the program risk or opportunity.

3. Click **More commands** ••• > **Edit** > **Save Edits**.

Updating the program network diagram

After adding the new WBS elements to mitigate risks or incorporate opportunities and receiving an approved estimate, you send the updated program WBS information to the external schedule. All the newly added WBS work packages and their resource requirements appear in the network diagram in the external schedule. As a program manager, you **add the new WBS work packages to the existing network chains**. Also, you can **update the network diagram** to reflect the updated estimate due to risk and opportunity incorporation.

Generating an updated program schedule

Once the network diagram is finalized, you **generate the updated schedule**. The updated schedule information can then be sent back to Teamcenter. You then **update the time-phased estimate** resulting from the schedule change.

Preparing a cost estimate and schedule

Understanding the preparation of a cost estimate and schedule

Whether you are preparing a response to an RFP or developing an internal initiative, estimation of the cost and the timeline for the project are key parts of the planning process. After identifying the tasks to accomplish the goals of the project, the cost and schedule for each task are estimated and reviewed before finalizing and eventually executing the project.

In IPP&E, each work breakdown structure (WBS) work package consists of multiple functional work packages. Each functional work package is assigned to a functional organization. That functional

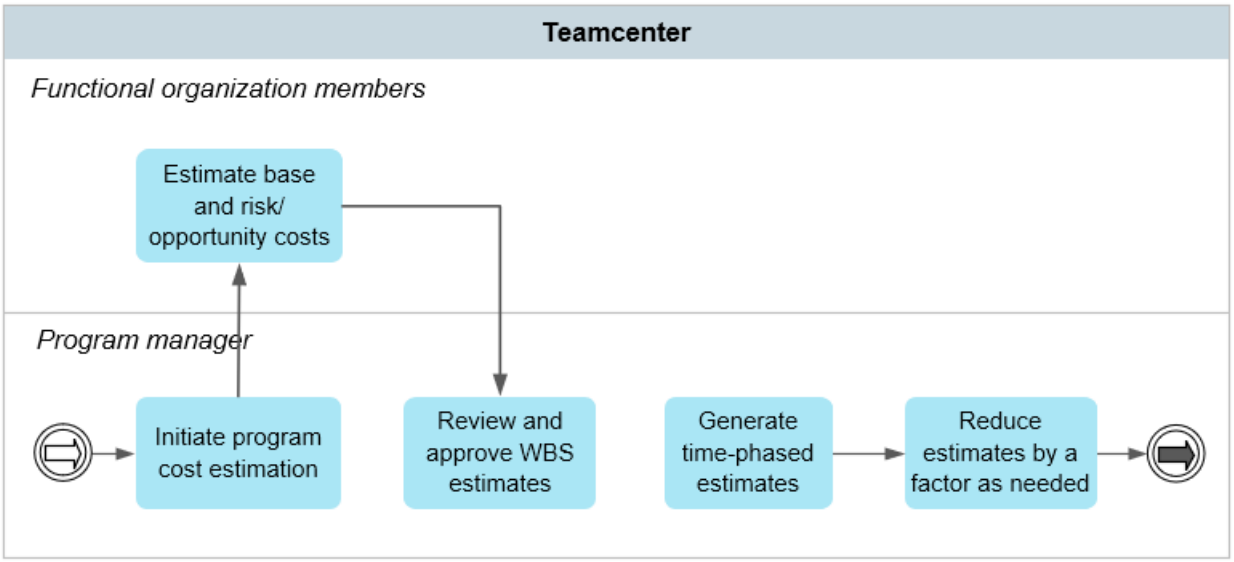
organization is then assigned the responsibility of estimating the level of effort necessary to execute the package and create the deliverables. In addition to the labor hours, the material and travel cost to accomplish the work is also estimated. All these estimates are reviewed and approved by the functional organization, the IPT for the program, and also the finance department.

The WBS work packages are then sequenced in a network of tasks, eventually feeding into the IMP criteria. The network diagram, along with the span time for each WBS work package, is used to generate a program schedule.

Based on the program schedule, the functional work package estimates are distributed to generate a monthly time-phased estimate for the project. This time-phased estimate and the labor rates are used to generate the program cost. This price and the schedule form the proposal for the project.

Cost estimation and schedule preparation business process

The steps in the cost estimation and schedule preparation process are as follows:



Update the FWP cost table

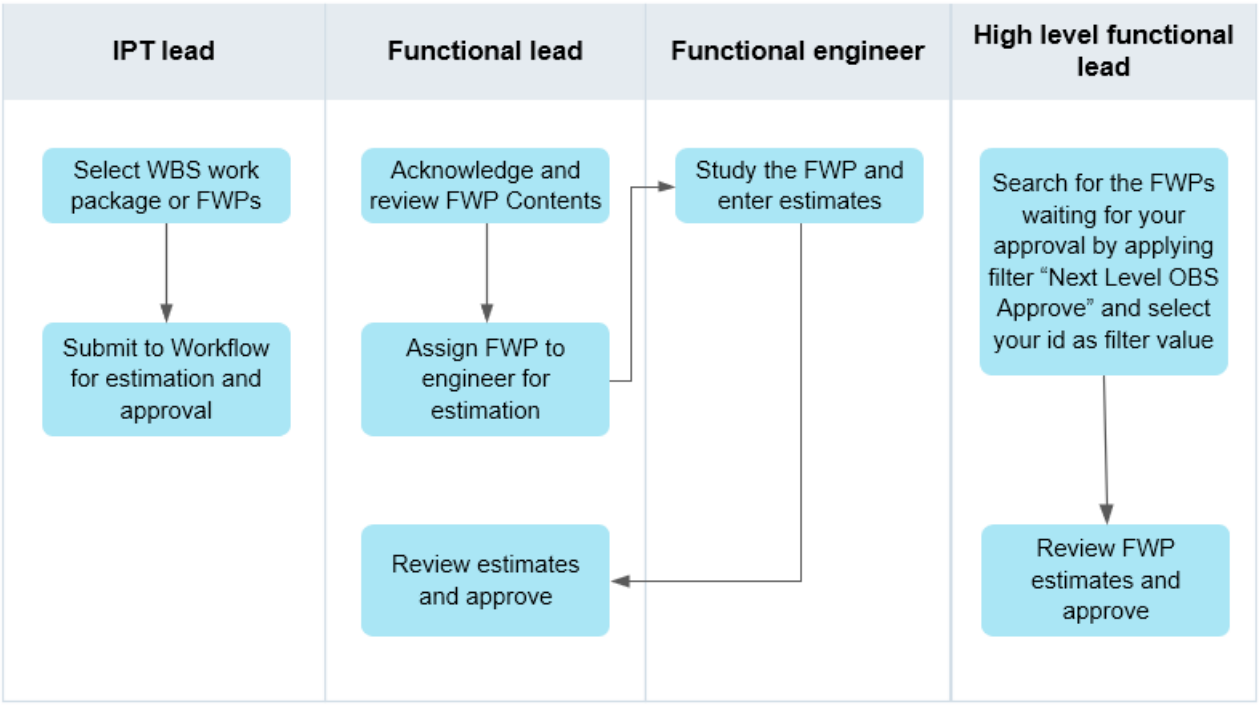
When you create an FWP from the RAM or by FWP request approval, the system creates a target cost form and a direct cost form for each FWP. It creates the rows in the target cost table for each active cost category and also creates rows for each valid cost element in the direct cost table. Valid cost elements comprise all non OBS controlled cost elements. If the cost element is controlled by the OBS, rows are created only for those work packages that are related to the functional organization of the FWP.

If you add a new cost category or cost element, update the cost forms. The system adds a new row in the target cost table for each new cost category. Likewise, it adds a new row in the direct cost table for each new cost element.

To update a cost table, open the target cost form or direct cost form and then choose **Edit** → **Start Edit**.

Initiating the program cost estimation process

Once the WBS contents are finalized, the respective IPTs initiate the estimation process for each WBS work package. The responsible IPTs set a target cost for each functional work package for the WBS work package before triggering the estimation process. The estimation process for the WBS work packages assigns each constituent functional work package to the leader of the responsible functional organization. The leader then starts the estimation process for each WBS work package in the program WBS. The estimation process for each WBS work package in the program WBS is initiated by submitting one or more WBS work packages to the **IPPE Estimation Assignment Process** workflow.



Estimate base and risk or opportunity costs

The leader of the functional organization acknowledges and reviews the work package. If there are risks and opportunities, the leader reviews them and determines their impact on the work package and then creates a placeholder for estimates of the risks and opportunities. The leader also assigns a user from the organization to perform the estimation.

The assigned user can enter the base labor and material estimates and also estimate for any applicable risk and opportunity. The labor estimates are entered in hours, while the material estimates are entered in the actual currency value. Labor estimates in hours can be entered for all the categories of labor for the specific functional organization responsible for this work package.

The leader of the functional organization assigns the task to estimate the cost to a specific user from the **Participants** tab of the functional work package revision.

The responsible user assigns the base estimates for labor and material cost elements to the functional work package as follows:

1. Open the functional work package and click the **Estimates** tab.

2. In the **Direct Cost** section, double-click the appropriate **Estimated Cost** table cells and enter the estimates.

3. Optionally, if a risk for the parent WBS work package impacts the FWP, estimate its cost. Open the FWP, click the **Risks** tab, and then select the risk that applies to the work package.

4. Click **Create Cost** from the work area toolbar above the table.

   The cost is displayed in the **Risk Cost** section.

5. Select the risk in the **Risk Cost** section and in the **Risk Cost Table** below it, double-click the appropriate table cells and enter the estimates.

6. Repeat steps 3 through 5 for any opportunities. Select the opportunity from the **Opportunity Cost** section of the **Opportunities** tab, and enter the estimates in the **Opportunity Cost Table** below.

Review and approve estimates

Reviewing and approving estimates

After the labor and material estimates are entered by the responsible party, the leader of the functional organization reviews the estimate and then approves or rejects it. If the estimate is rejected, the system returns it to the responsible party for re-estimation. If the estimate for the functional work package is approved, it becomes available for review by the upper level management of the functional organization. Upper level management can review and approve or reject multiple functional work packages at the same time.

Once a functional work package is approved by the topmost approval authority in the functional organization, the package is marked as approved. Once all the functional work packages for a WBS work package are approved, the WBS work package becomes available for review by the IPT organization. The responsible IPT can review and approve or reject multiple WBS work packages simultaneously.

Once a WBS work package is approved by the topmost approval authority in the IPT organization, it becomes available for approval by the finance department. The WBS work package is marked as approved after approval by the finance lead and the program manager.

Note:

Hierarchical approval of estimates is denoted as a best practice in a sample implementation. You can implement approval workflows as per your process needs.

The IPP&E solution provides the following sample workflows for the estimation approval process:

- **IPPE Estimation Assignment Process** — This workflow initiates the estimation assignment process on the work package revision and initiates the estimation approval sub-process for each functional work package revision.
- **IPPE FWP Estimation Approval Process** — This workflow provides the estimation assignment and approval for the FWP by the functional organization lead. This process also triggers the estimation process on any dependent FWPs if the current FWP is approved. Additionally, the work package state is updates.
- **IPPE FWP Estimation Management Approval Process** — This workflow allows higher-level functional organizational leads to approve the FWP estimation. Release status is applied if the final approving level in the OBS approves the estimates. The work package estimation state is updated to **Review**.
- **IPPE IPT Estimation Approval Process** — This workflow allows higher-level functional organizational leads to approve the FWP estimation. Release status is applied if the final approving level in the OBS approves the estimates. The work package estimation state is updated to **Review**.
- **IPPE IPT Management Estimation Approval Process** — This workflow allows higher-level functional organizational leads to approve the FWP estimation. Release status is applied if the final approving level in the OBS approves the estimates. The work package estimation state is updated to **Review**.
- **IPPE Finance Estimation Approval Process** — This workflow provides finance-level approval for WBS work package estimates.
- **IPPE Program Manager Estimation Approval Process** — This workflow provides management approval of work packages for estimation.
- **IPPE Work Package Owner Approval Process** — This workflow provides work package owner approval when an IPT is not assigned to the work package.

Review and approve estimates as a higher level functional organization leader

As a senior leader responsible for a functional organization, you can review and approve the estimates for several functional work packages at the same time as follows:

1. Use the standard search capability to search for the keyword **fwp**.

All the results matching the keyword are displayed, potentially including ones that are not related to functional work packages.

2. Choose **Filter** and then select **Functional Work Package Revision** from the list under **Type** in the **Filters** panel.

3. Select your user ID from the list under **Next Level OBS Approver** in the **Filters** panel.

   Tip: Save this index search as a named saved search. You can then use it for later approvals without going through the previous steps again.

4. Approve the FWPs waiting for the next-level OBS approval.

Review and approve estimates as an IPT leader

As an IPT lead for a program, you can review and approve the estimates for several WBS work packages at a time as follows:

1. Use the standard search capability to search for the keyword **wbs**.

   All the results matching the keyword are displayed, potentially including ones that are not related to work breakdown structures.

2. Choose **Filter** and then select **WBS Element Revision** from the list under **Type** in the **Filters** panel.

3. Select **IPT Review** from the list under **Estimation State** in the **Filters** panel.

4. Select your user ID from the list under **Next Level IPT Approver** in the **Filters** panel.

   > Tip:
   >
   > Save this index search as a named saved search. You can then use it for later approvals without going through the previous steps again.

5. Approve the WBS work packages awaiting IPT approval.

Review and approve estimates as a finance lead

As a finance lead for a program, you can review and approve the estimates for several WBS work packages at the same time as follows:

1. Use the standard search capability to search for the keyword **wbs**.

   All the results matching the keyword are displayed, potentially including ones that are not related to work breakdown structures.

2. Choose **Filter** and then select **WBS Element Revision** from the list under **Type** in the **Filters** panel.

3. Select **IPT Approved** from the list under **Estimation State** in the **Filters** panel.

> Tip:
>
> Save this index search as a named saved search. You can then use it for later approvals without going through the previous steps again.

4. Approve the WBS work packages.

Review and approve estimates as a program manager

Final approval of the estimates is granted by the program manager. As the program manager, you can review and approve the estimates for several WBS work packages at the same time as follows:

1. Use the standard search capability to search for the keyword **wbs**.

   All the results matching the keyword are displayed, potentially including ones that are not related to work breakdown structures.

2. Choose **Filter** and then select **WBS Element Revision** from the list under **Type** in the **Filters** panel.

3. Select **Finance Approved** from the list under **Estimation State** in the **Filters** panel.

> Tip:
>
> Save this index search as a named saved search. You can then use it for later approvals without going through the previous steps again.

4. Approve the WBS work packages.

Create a network diagram of the program WBS

The network diagram represents the sequence of WBS work packages in a chain of tasks leading to specific criteria of the integrated master plan (IMP) for the program. You create the network diagram by linking the WBS work packages together and to the criteria to form a network of program tasks.

> Note:
>
> This capability is only available with the Aurora integration, which is available in Support Center.

The network diagram is created in the external schedule software using WBS data from Teamcenter. The data pushed from Teamcenter to the external schedule consists of:

- All WBS work packages in the program WBS.
- The events, accomplishments, and criteria representing the program IMP.
- The initial span time for each work package.
- Any links already created in Teamcenter between the WBS work packages and IMP criteria.
- Resource requirements for each work package. The resource requirement includes labor hours for each labor category for all functional work packages contained in the WBS work package.

The program manager sends the released program WBS to the external schedule.

As a program manager, create links between the WBS work packages and IMP criteria as follows:

1. Select the WBS root and export it to the external scheduling software.

   The network diagram opens in the external schedule. It displays the network of program WBS elements and also events, accomplishments, and criteria as tasks. Any links that already exist in the program WBS are displayed in the network diagram.

2. Create links between the tasks in the external schedule.

Generate a program schedule

After the network diagram is finalized, create the schedule for the program by assigning it a start date. Add or adjust the span time for each WBS work package, as necessary. Before the schedule can be generated, you may have to adjust the available resource for each labor category in the external schedule.

Once the schedule is generated, the system assigns each WBS work package (represented by a schedule task in the schedule) an early start and an early end date along with any slack available to execute that task.

The schedule data for each work package is then sent to Teamcenter and is populated as **Task Info**. This information is visible when executing this task in the Active Workspace.

Generate time-phased estimates

Once the start and end date for each work package is populated, the estimates for the functional work packages are distributed to generate a monthly time-phased estimate. The leader of the functional organization can manually override those estimates. However, the system enforces the calculation such that the manual distribution of the estimates still adds up to the base estimate.

As a program manager, generate the time-phased estimate for each functional work package as follows:

1. Select an FWP and choose **New** → **Time Phased Estimates**.

   The system generates the time-phased cost for the FWP.

2. To view the generated time-phased estimates, select the FWP, click the **Estimates** tab, and select the latest item in the **Time Phased Estimates** section.

   The **Time Phased Estimate Table** is populated with values.

3. As leader of the functional organization, you can optionally override the time-phased estimate. From the **Estimates** tab, choose **Edit** → **Start Edit**, and adjust the distributed cost from one month to another as necessary in the **Time Phased Estimate Table**.

4. Choose **Edit** → **Save Edits** to save the changes.

Generating special schedules

IPP&E can generate three special schedules: the best-case schedule, the worst-case schedule, and the most-likely schedule. You can compare the original schedule with any of these schedules and then do a *what if* analysis to provide an assessment of the original schedule for the bid.

> Note:
>
> This capability is only available with the Aurora integration, which is available in Support Center.

Ideally, the original schedule for the WBS structure must be created before generating these special schedules. Therefore, as a pre-condition for using these schedules:

- Data for the WBS structure must be exported to the external schedule.
- All the constraints and span time must be updated in the external schedule.
- Schedule dates must be generated in the external schedule.
- Data must be exported back to Teamcenter from the external schedule.

The special schedule can be generated by exporting the data to the external schedule using the appropriate workflow. The system collects information based on the type of schedule to generate. Once the export is complete, you can generate the schedule in the external schedule.

- *Best-Case Schedule*

  The best-case schedule assumes that all the opportunities associated with different elements of the WBS structure are realized, allowing you to visualize the schedule as if all the opportunities are incorporated. You can compare the original schedule with the best-case schedule to perform the *what if* analysis.

The best-case schedule can be generated by submitting the WBS root to the **IPPE TC To External Schedule Transfer Best Case** workflow. When data for the best-case schedule is exported to the external schedule, the sum of the time span of all the opportunities applicable to the work package is subtracted from the span time of the work package. The resultant time span becomes the task duration in the external schedule. Effectively, opportunities reduce the duration of the task and schedule.

- *Worst-Case Schedule*

  The worst-case schedule assumes that all the risks associated with different elements of the WBS structure are realized, allowing you to visualize the schedule as if all the risks are incorporated. You can compare the original schedule with the worst case schedule to perform the *what if* analysis.

  You generate the worst-case schedule by submitting the WBS root to the **IPPE TC To External Schedule Transfer Worst Case** workflow. When data for the worst-case schedule is exported to the external schedule, the sum of time span of all the risks applicable to the work package is added to the span time of the work package. The resultant time span becomes the task duration in the external schedule. Effectively, risks increase the duration of the task and schedule.

- *Most-Likely Schedule*

  The most-likely schedule assumes that all the risks and opportunities associated with different elements of the WBS structure are realized, allowing you to visualize the schedule as if all the risks and opportunities are incorporated. You can compare the original schedule with most-likely schedule to perform analysis.

  You generate the most-likely schedule by submitting the WBS root to the **IPPE TC To External Schedule Transfer Most Likely Case** workflow. When data for most-likely schedule is exported to the external schedule, the sum of the time span with its probability of all the risks applicable to the work package is added to the span time of the work package. Also, the sum of the time span with its probability of all the opportunities applicable to the work package is subtracted from the span time of the work package. The resultant time span becomes the task duration in the external schedule.

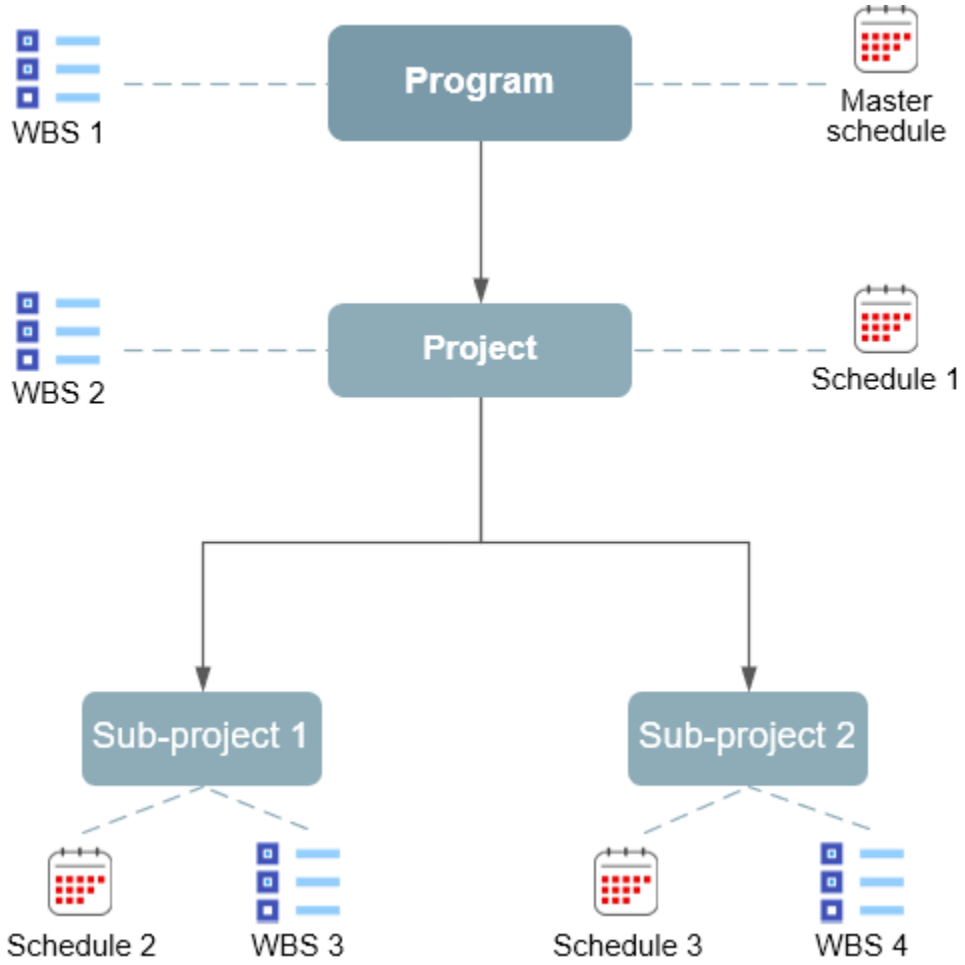
Defining a master schedule and proxy tasks

Defining and updating the master schedule

Depending on the complexity of the program, you can break the program down into multiple projects. Likewise, each project may be broken into multiple subprojects. Each of these projects and subprojects might have its own schedule. If the program manager or project manager wants to monitor all the schedules together, IPP&E allows you to generate the master schedule in the external schedule. The master schedule collects the task information from all schedules attached to each project and subproject and creates a new task in the master schedule. If there are updates to the original schedule, the master schedule is automatically updated. The master schedule is supported only for analysis purposes, for example, *what if* analysis, resource analysis, or risk impact analysis.

Note:

Importing the master schedule from the external schedule into Teamcenter is not supported.



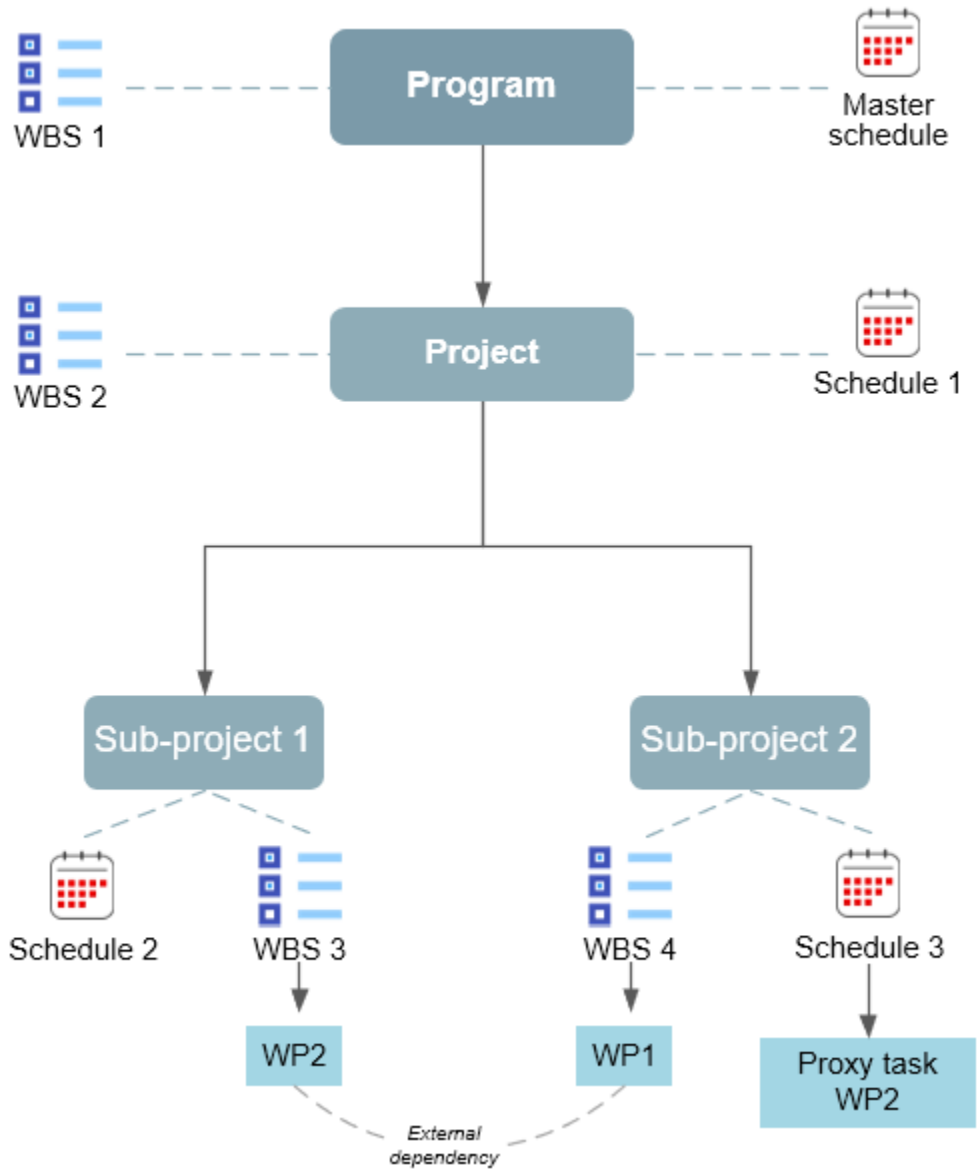
You can generate the master schedule by submitting the plan to the **IPPE Generate Master Schedule** workflow. The workflow communicates internally with the external schedule using Teamcenter Integration Framework and creates the master schedule using the plan ID.

If there are updates to the original schedule in the external schedule, you import the update to Teamcenter and the system updates the master schedule. If the master schedule update fails, you can manually initiate an update by submitting the plan to the **IPPE Update Master Schedule** workflow. This workflow checks the list of updated schedule properties on the plan and collects the data from that schedule to update the master schedule.

Understanding proxy tasks

If there are dependencies between two work packages in different WBS structures, the system creates a proxy task in the schedule for the work package from the other WBS structure. It picks the dates and

duration from that work package. The proxy task is created only if the task information for the work package has dates.



Executing the program

Understanding program execution

A key benefit of the integrated approach to program management is that the same schedule generated during the planning phase is used in the execution phase. The complete traceability between the program schedule and the program WBS and all its artifacts ensures that data used throughout the program management lifecycle is consistent and visible to all stakeholders.

After awarding the contract or receiving approval from management for an internal project, the program team can adjust the program schedule to match the agreed upon start date. After creating an earned value baseline using the external Earned Value Management System (EVMS) software, the program manager can initiate the execution of the program.

During the execution phase, the tasks are automatically triggered based on their start date and the completion of the predecessor tasks in the schedule. All the work packages for the corresponding WBS work packages are assigned to the responsible functional organization for execution. The deliverables generated for each functional work package are sent through a review and approval process by the leaders of the respective functional organizations.

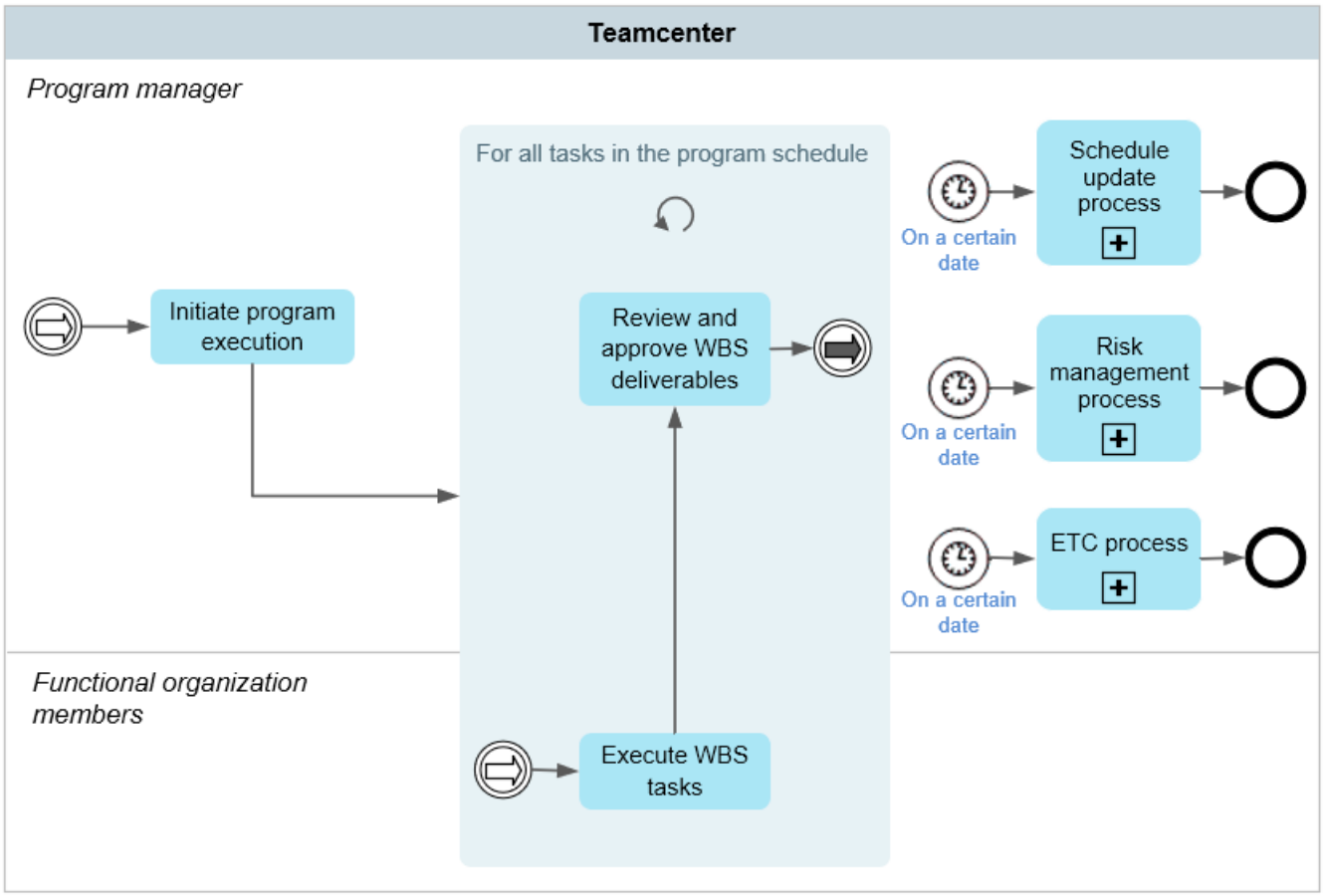
During the execution phase, the program management team can update the WBS and the schedule to handle any delay in the execution of the tasks or any failure or missed scope tasks. The WBS and schedule can also be updated to manage program risk and opportunity.

Regular status updates can be sent to the EVMS software to create the earned value report. Additionally, during the execution phase, the functional organizations can re-estimate the cost for the work packages for future tasks in the schedule.

An EVMS file can be generated from the corresponding Task Info object and WBS properties.

Program execution business process

The program execution business process is as follows:



Initiate program execution

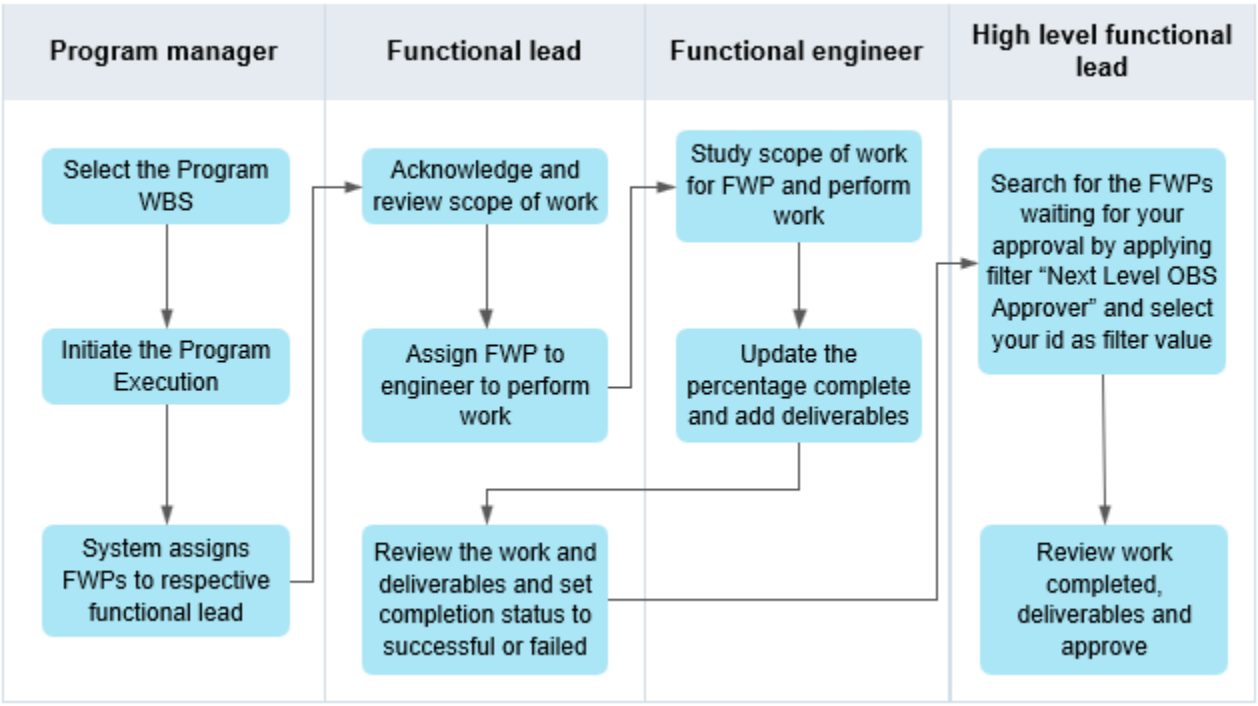
As program manager, execute the program as follows:

1. In the external schedule, adjust the start date of the schedule to the agreed upon program start date.

2. Push the updated schedule to Teamcenter and regenerate the time-phased estimates for the functional work packages.

3. Select the program WBS and initiate the program execution. (Program execution may be initiated by a workflow.)

   The system initiates execution of the program and changes the **Project Phase** property value on the **Overview** panel to **Execution**.



Automatically trigger the Execution Workflow utility

The **ipp0_schedule_execution** utility automatically triggers the workflow for any task. It is packaged with the IPP&E solution and gets installed in the *TC_ROOT/bin* directory when the Teamcenter server package for the IPP&E solution is installed.

Description

During the IPP&E process, when the user generates a schedule in Aurora, all the tasks in the schedule get populated with a start date, an end date, a float time, and other data. The user can export this data back to Teamcenter, and the data gets stored in the **Task Info** objects in Teamcenter. **Task Info** is a business object defined in Teamcenter to store all the properties related to Aurora tasks. For every task in Aurora, there is a corresponding **Task Info** object in Teamcenter.

During the execution phase of any project, when a **Task Info** object is due for execution, it must be submitted to the execution workflow. To automate this process, IPP&E has provided the **ipp0_schedule_execution** utility. This utility automatically submits the **Task Info** object to the workflow.

The **ipp0_schedule_execution** utility normally iterates until the program is terminated. In iterative mode, it queries for the **Task Info** objects by its early start date using the time window determined from the current time and an interval (in minutes). Alternatively, it can be run in a non-interactive, one-pass mode with the time window supplied by the start and end date or time command arguments.

Note:

This utility can be executed by an administrator only when the first program is ready for execution.

Usage

Default mode: Find and execute tasks where the start date of the task is current or past due.

```
ipp0_schedule_execution.exe -u=<user-id> -p=<password> -g=<group>
-interval=<minutes (default=5)>
```

Example:

```
ipp0_schedule_execution.exe –u=Tc-admin-user –p=password –g=group –
interval=2
```

One-pass mode: Find and execute tasks where the start date of the task falls between the specified dates.

```
ipp0_schedule_execution.exe -u=<user-id> -p=<password> -g=<group>
-start=<dd-MMM-yyyy hh:mm> -end=<dd-MMM-yyyy hh:mm>
```

Example:

```
ipp0_schedule_execution.exe –u=Tc-admin-user –p=password –g=group
-start="01-Mar-2021 00:00" -end="15-Mar-2021 23:59" OR -start=01-
Mar-2021 -end=15-Mar-2021
```

Execute WBS tasks

The system assigns each constituent FWP in the WBS work packages to the leader of the responsible functional organization. In this role, you can review and acknowledge the work package. You then assign a user from your organization to execute the package.

As the assigned user, you review the scope of work and any input deliverable from the predecessor tasks. You then perform the work and relate any deliverables you generate to the FWP. Finally, you sign off on the assignment and the package moves to the functional leaders for review and approval.

As the assigned user, perform the necessary work including adding deliverables and updating percentage completion.

To add a deliverable to the FWP:

1. Click the **Deliverables** tab and then click **Add to** ⊕ from the work area toolbar in the **Deliverables** section.

The **Add** panel is displayed.

2. You can add a deliverable from the **Add** panel in several ways:

   - Create a new deliverable item from the **New** tab.
   - Use the **Palette** tab to relate an existing deliverable item copied to the clipboard, added as a favorite, or recently opened.
   - Use the **Search** tab to find an existing deliverable item using a search box and filters.

   The deliverable is attached to the FWP.

Reviewing and approving the deliverables

After the assigned user creates the deliverables, the leader of the functional organization reviews and approves the deliverables. If the deliverables are rejected, the system returns them to the assigned user for rework. While reviewing the deliverables, you can mark the completion state of the work package as successful or failed, depending on the results of the task. When the deliverables for the FWP are approved, the system sends them for review by the upper level management of the functional organization. Management can review and approve or reject several functional work packages at the same time.

Once a functional work package is approved by the topmost approval authority in the functional organization, its package is marked as approved. The execution state of the WBS work package is marked as completed once all the constituent work packages are approved. The completion status of the WBS work package is set to failed or successful if the completion status of any constituent work package is set to one of these statues.

As a functional organization leader, you review the completed work package and review the deliverables, and mark the completion state as **Successful** or **Failed**.

The IPP&E solution provides the following sample workflows for the execution approval process:

- **IPPE Execution Assignment Process** — This process executes the work package based on the start date in the task info. It is triggered by the **ipp0_schedule_execution** utility.
- **IPPE FWP Execution Approval Process** — This process executes the FWP and completes the task.
- **IPPE FWP Execution Management Approval Process** — This process approves the FWP execution by higher-level OBS leads.
- **IPPE Transfer Ownership to IPT** — This process transfers the ownership of the FWP to the assigned IPT.

Initiating the estimate to complete process

As a program manager, you can allow the functional organization to adjust the estimates for all future tasks in the schedule. The leaders of the functional organization can then update the estimates, following the estimate to complete (ETC) process. If, as a functional organization user, you choose to update the estimate for a work package, you must manually revise it before entering the new estimate. If a new estimate is entered for the work package, you must send it through the review and approval process.

The IPP&E solution provides the **IPPE FWP ETC Initiation Process** sample workflow.

The system notifies the FWP leads of ETC process initiation and also changes the **Is ETC Process** attribute value to **True** for the WBS root.

Updating the schedules

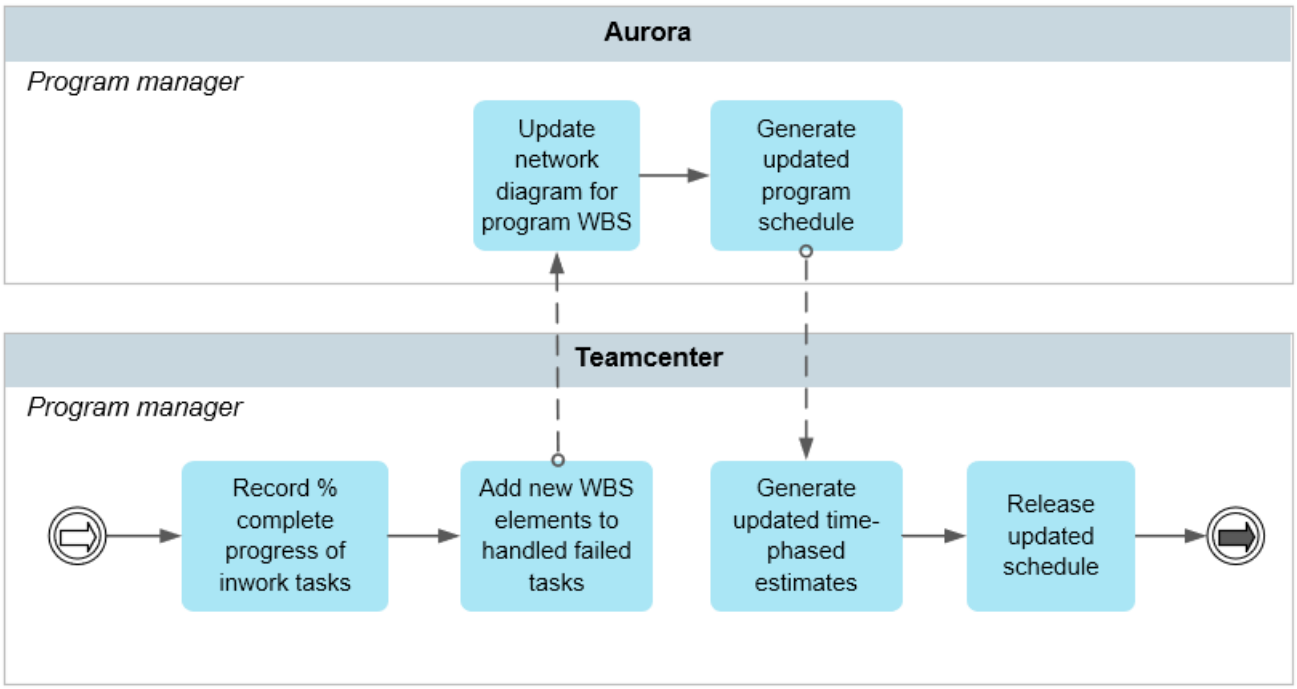
Understanding schedule updates

During the execution of the program schedule, there may be delays to the completion of specific tasks. Some tasks, even if completed on time, may not provide expected results. Sometimes, a task may be marked as complete without fulfilling the complete scope of work. To handle these situations, you may have to update the program schedule or change the sequence of the tasks. You may have to add new recovery tasks.

As a program manager, you can add new WBS elements in the program WBS to handle failed tasks. These WBS elements can then be estimated and approved. You can send the updated program WBS to the external schedule to update the network diagram and generate the updated schedule. The new schedule information is sent back to Teamcenter for pricing and EVMS baseline updates. All these tasks can be performed without affecting the already in-work tasks.

Schedule update business process

The following diagram shows the schedule update business process:



Record program execution progress

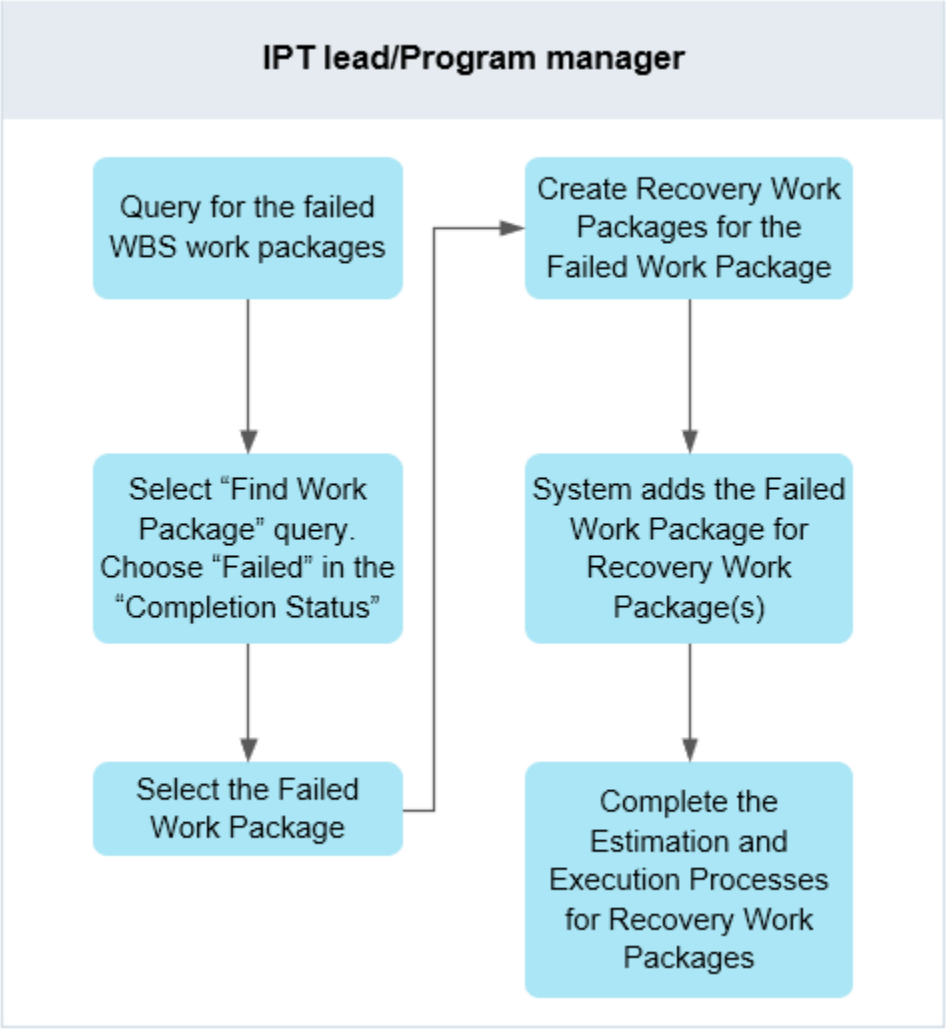
As a functional organization user or leader, you can enter the progress of a functional work package any time during its execution. You enter progress as a percentage complete value using a scale of 1-100. The system calculates progress for a WBS work package by rolling up the percentage completion values of all its constituent work packages. As an IPT lead or program manager, you can review the percentage complete status of the WBS work package. You can then extract the progress information to create an EVMS status report in a format that can be exported to the EVMS software.

1. Open the FWP and click the **Overview** tab.

2. Choose **Edit** → **Start Edit** and then update the **Percentage Complete** field in the **Additional Properties** section.

Manage failed tasks

During the schedule update process, the program manager or IPT lead reviews the current status of all completed tasks. If any of the completed tasks are marked as *failed*, you create new WBS work packages to recover. You then link the new work packages to the failed and missed scope work packages to ensure complete traceability. You then submit the new WBS work packages to the estimation and execution processes.



1. Query for *failed* WBS work packages.
2. Open the program WBS and click **Find** in the results panel toolbar to search within the structure.
3. Select the **Find Work Package** query and then enter **Failed** in the **Completion Status** field.

   A list of all the failed WBS work packages is displayed.
4. Create recovery work packages, following the procedure for **adding WBS elements to the WBS**.
5. Link the recovery work packages to the failed work package.
6. Open the newly created recovery work package and check its **Source** property refers to the failed work package.
7. Repeat steps 3 through 5 for the other failed work packages.

Update the program network diagram

After adding the new WBS elements to handle the failed or missed scope tasks and receiving an approved estimate, send the updated program WBS information to the external schedule. All the newly added WBS work packages and their resource requirements then show in the network diagram in the external schedule. As a program manager, you can add the new WBS work packages to existing network chains. Additionally, you can update the network diagram to handle any delayed tasks as needed.

Send the updated program WBS to the external schedule as follows:

1. Select program WBS and send the updated program WBS to the external schedule.

2. Update the network diagram in the external schedule, following the procedure for **creating a new network diagram**.

Generating an updated program schedule

Once the network diagram is finalized, you can generate the updated schedule. The updated schedule information can then be sent back to Teamcenter and you can update the time-phased estimate resulting from the schedule change.

As a program manager, you can send the updated schedule information to Teamcenter using the same procedure as for **generating the program schedule**.

As a leader of a functional organization, you can manually override the **updated time phased estimates**.

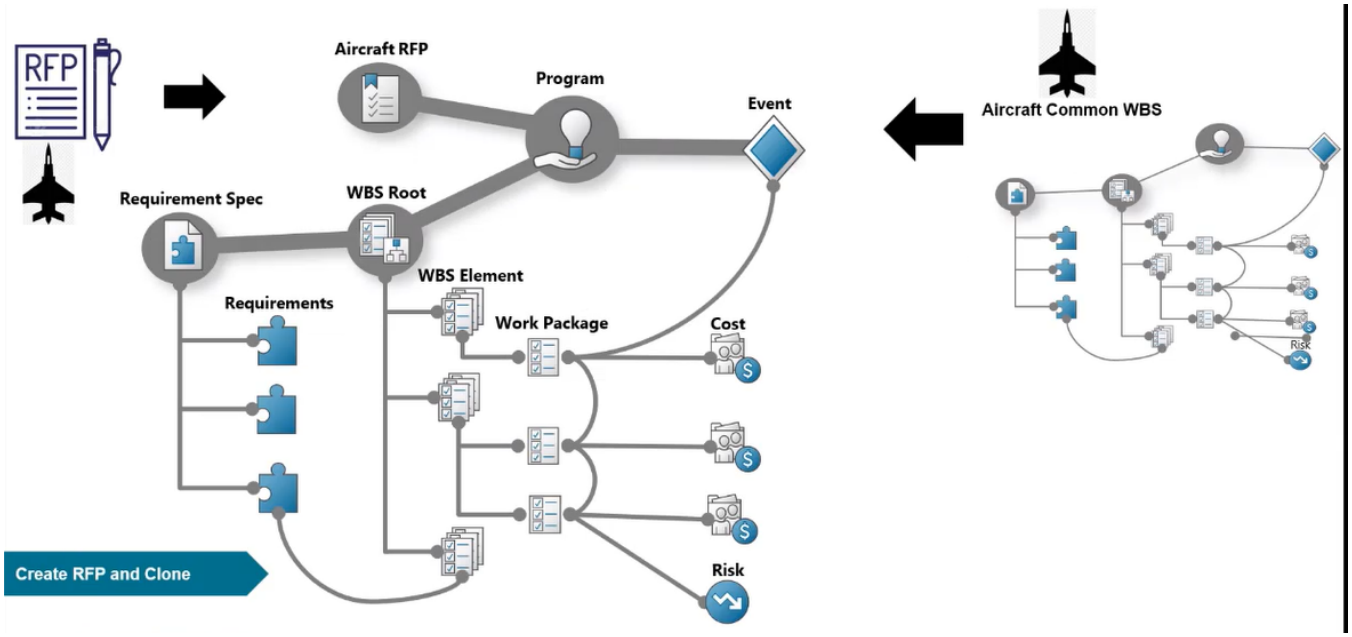
Managing the common WBS

Understanding the common WBS

Maintaining historical WBS data based on best practices and lessons learned from past projects can help companies quickly realize a specific WBS for an RFP. You can then use the same WBS to finalize the cost and schedule after awarding the contract, and to eventually execute the project. You can organize historical WBS data by various criteria, including product category and business unit, to suit specific business needs.

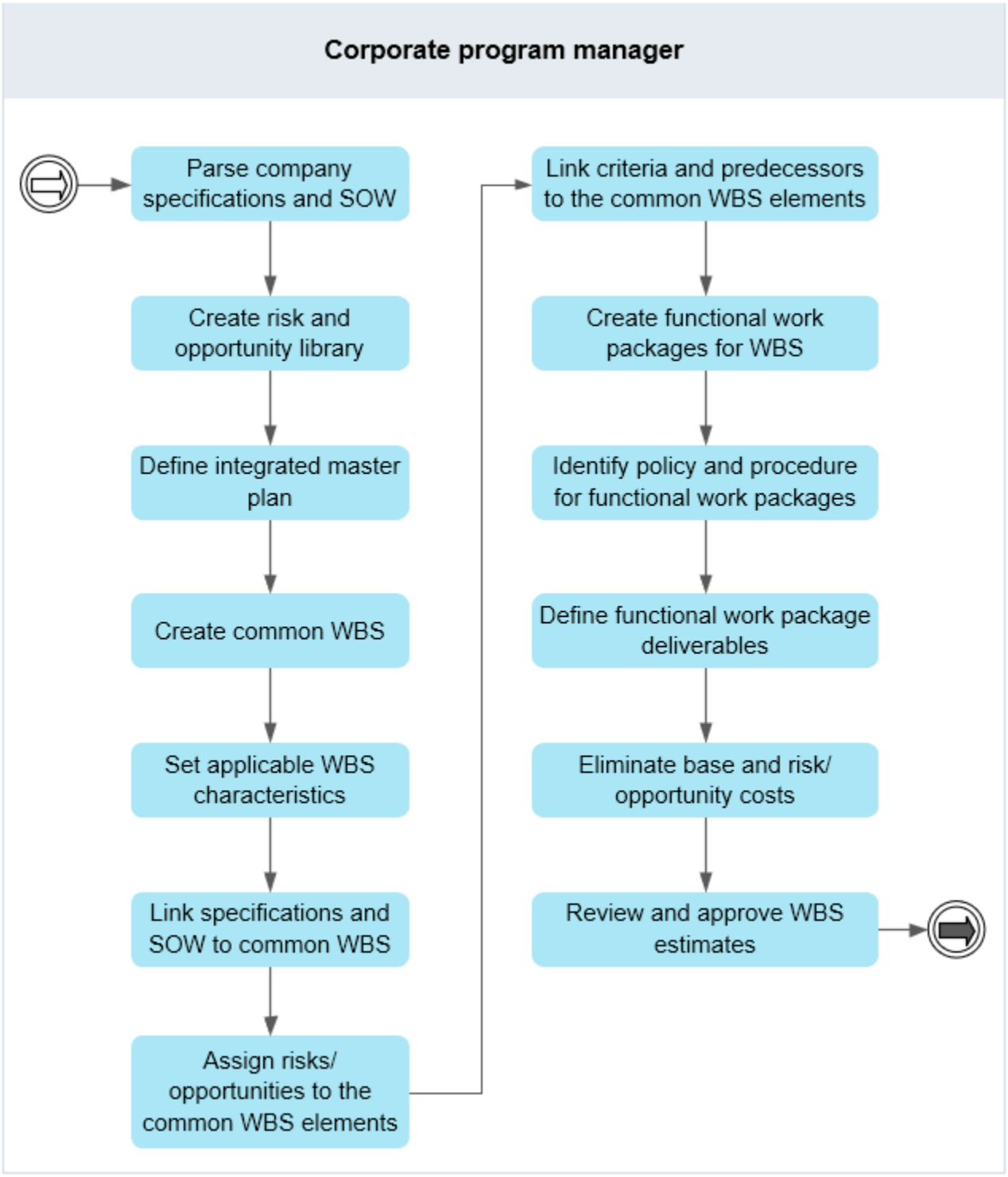
IPP&E allows you to manage a library of common WBSs that represents a collection of WBS elements for a variety of products and processes relevant to the company. It provides a systematic and standardized method of gathering cost and other relevant data across all types of programs. The common WBS provides a template of a particular class of product that you can clone for multiple RFPs.

Having actual historical data to support cost estimates of similar products is a valuable resource. You can organize the library of common WBSs based on the need of the customer, and it can be improved from **Lessons Learned** in the execution of past or present programs.



The common WBS business process

The following diagram shows the common WBS business process:



Parsing company specs and statement of work (SOW) documents

If your company maintains Statement of Work (SOW) documents and company or regulatory requirement specifications, the corporate program manager **authors or imports the following documents** into Teamcenter.

- Company Requirement Specifications.
- Company Statement of Work Document.

Create a library of risks and opportunities

Companies maintain a standard set of risks and opportunities for their business that are available across programs for the risk management process. Each risk and opportunity typically have a unique ID. It can be classified into business specific categories in the library. You can also define their probability of occurrence and also the consequence.

To create a new risk or opportunity, choose **New** → **Add**, and then choose **Risk** or **Opportunity**.

The **Add** panel is displayed, where you enter details of the new risk or opportunity, and then click **Add**.

> Note:
>
> Be sure to clear the **For Program** check box.

Defining an integrated master plan

Programs are planned and executed using standard milestones for the design, manufacturing, and servicing phases. This is done with the Integrated Master Plan (IMP), which is an event-based, top level plan consisting of a hierarchy of *Program Events*. Each event is decomposed into specific accomplishments, and each specific accomplishment is decomposed into specific *Criteria*.

For the common WBS library, you **develop a standard IMP** to use across all programs. When you create a WBS for a program using the common WBS library, all the applicable *Events*, *Accomplishments*, and *Criteria* are copied to create an IMP for the specific program.

Create a common WBS

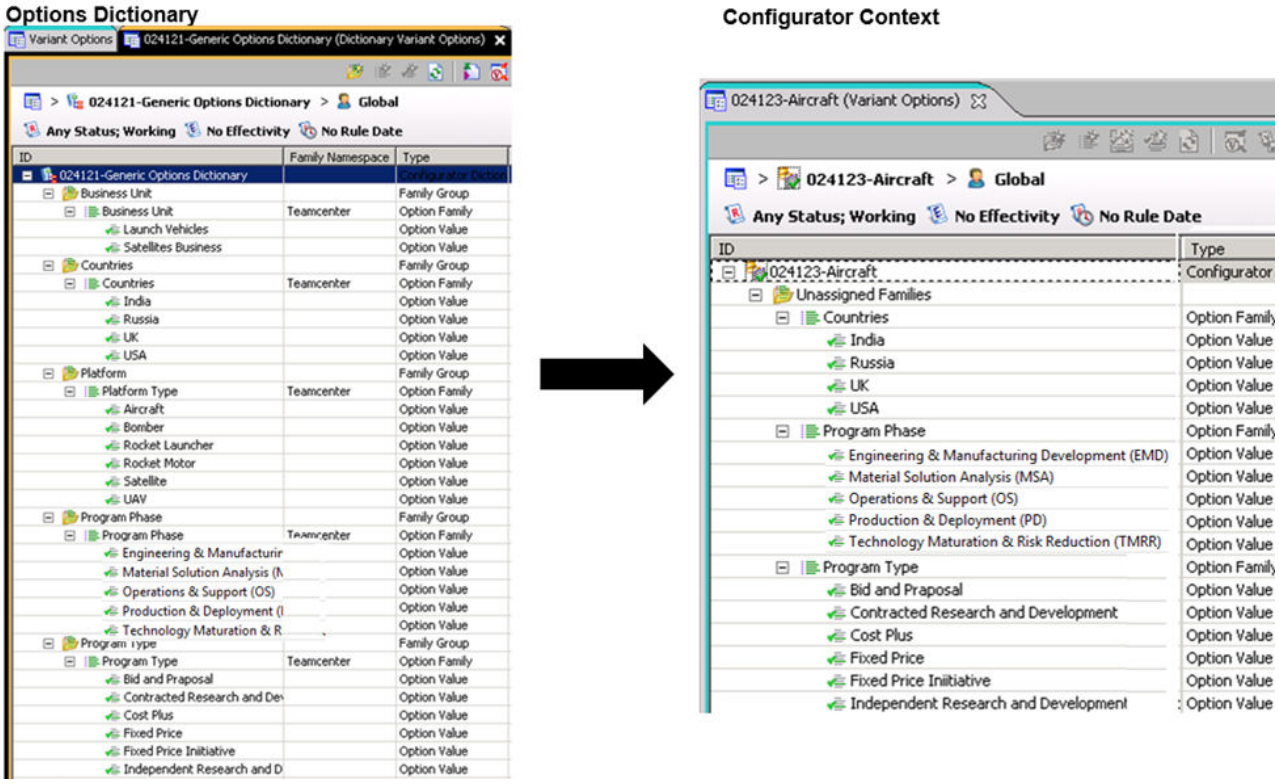
The common WBS consists of multiple WBS root elements, each representing a unique category of products. In the case of the defense industry, Mil-Std (Work Breakdown Structure for Defense Items) defines separate WBSs for different systems such as aircraft systems, electronic systems, missile systems, ordinance systems, sea systems, space systems, surface vehicle systems, and UAV systems. For a commercial product, your company can adopt a similar schemes.

Within the common WBS, a specific category may contain WBS elements with the same number but with different information including additional characteristics.

To create a common WBS library, first define the characteristics for your needs. You then create the WBS root element and the lower level WBS elements, and then apply those characteristics to the WBS elements where needed.

1. Create filters to represent the characteristics of the common WBS. In Product Configurator in the rich client, create an *options dictionary* containing all possible filters (variability).

2. Create a *configurator context* from the *options dictionary*.



3. Open the configurator context and create variant rules to apply the filters.

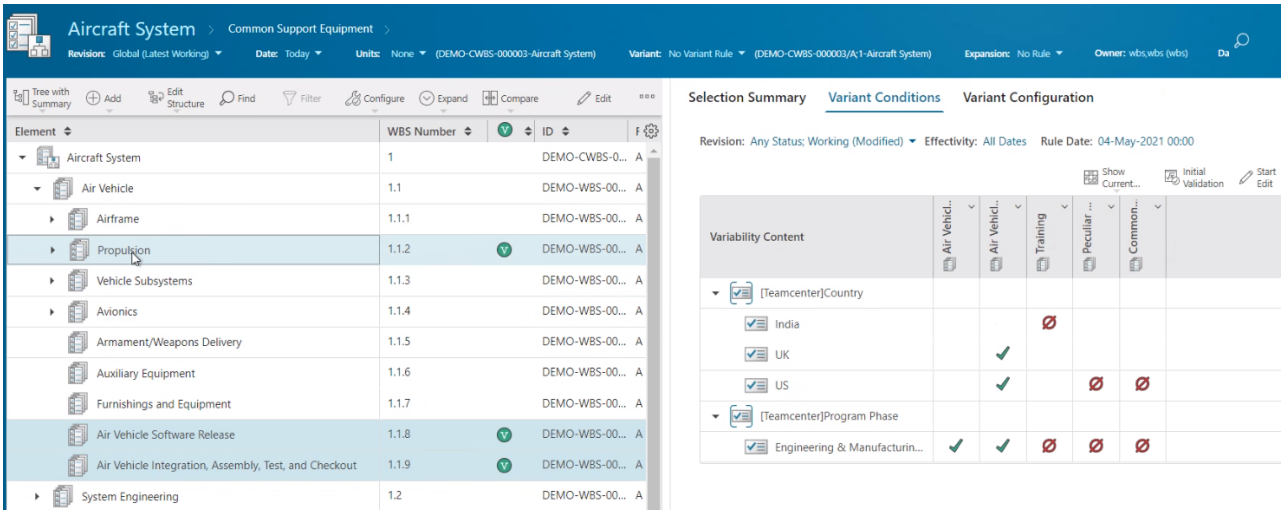
4. Create the common WBS and root WBS elements for the common WBS.

5. Attach the configurator context to the common WBS root element.

> Note:
>
> See *Product Configurator* for more details about creating an options dictionary, a configurator context, and variant rules in the rich client.

6. In the Active Workspace, **add new WBS elements** under the root WBS node.

7. Apply filters to each WBS element, as necessary, from the **Variant Conditions** tab.



Linking artifacts to the common WBS

Once you have added the WBS elements in the common WBS to handle the company or regulatory requirements and SOW paragraphs, record the traceability between the specifications and the corresponding WBS element. To do this, link requirements, SOW paragraphs, criteria, and predecessor work packages to the common WBS using the following procedures:

- **Link the RFP specifications to the program WBS**.
- **Identify criteria and predecessors for WBS work packages.**

Clone WBS elements for the common WBS

The common WBS may contain several WBS elements with the same WBS number if these WBS elements have different filters, for example, WBS elements for different countries. Multiple WBS elements with the same number may be used if the cost estimates vary for the same element. For example, the wing WBS element is part of the WBS hierarchy of an aircraft but the cost differs depending on whether the wing is designed and manufactured in the U.S. or Europe. In this case, the common WBS has two WBS elements with the same number, one for U.S. and another for Europe. Applying a filter for the U.S. or for Europe configures the correct WBS element.

To create a similar branch of a WBS element with different filters, you can **clone an existing branch of WBS elements**. IPP&E clones the WBS elements and all its children as well as the related artifacts.

Additionally, when creating a WBS structure for a program from the common WBS, you can ignore certain WBS elements. In this case, the ignored WBS elements are not cloned from the common WBS to the program WBS. To add the ignored or missing WBS elements from the common WBS to the project WBS after initial creation, use the cloning capability.

Creating functional work packages for the common WBS

You can create and populate functional work packages for the common WBS by performing the following task:

- **Create functional work packages for WBS**.

Estimating functional work packages for the common WBS

You estimate functional work packages in the common WBS by performing the following tasks:

- **Initiate program cost estimation**.
- **Estimate base and risk/opportunity costs**.

Reviewing and approving functional work package estimates for the common WBS

You can review and approve functional work package estimates for the common WBS by performing the following task:

- **Review and approve estimates**.

Incorporating lessons learned

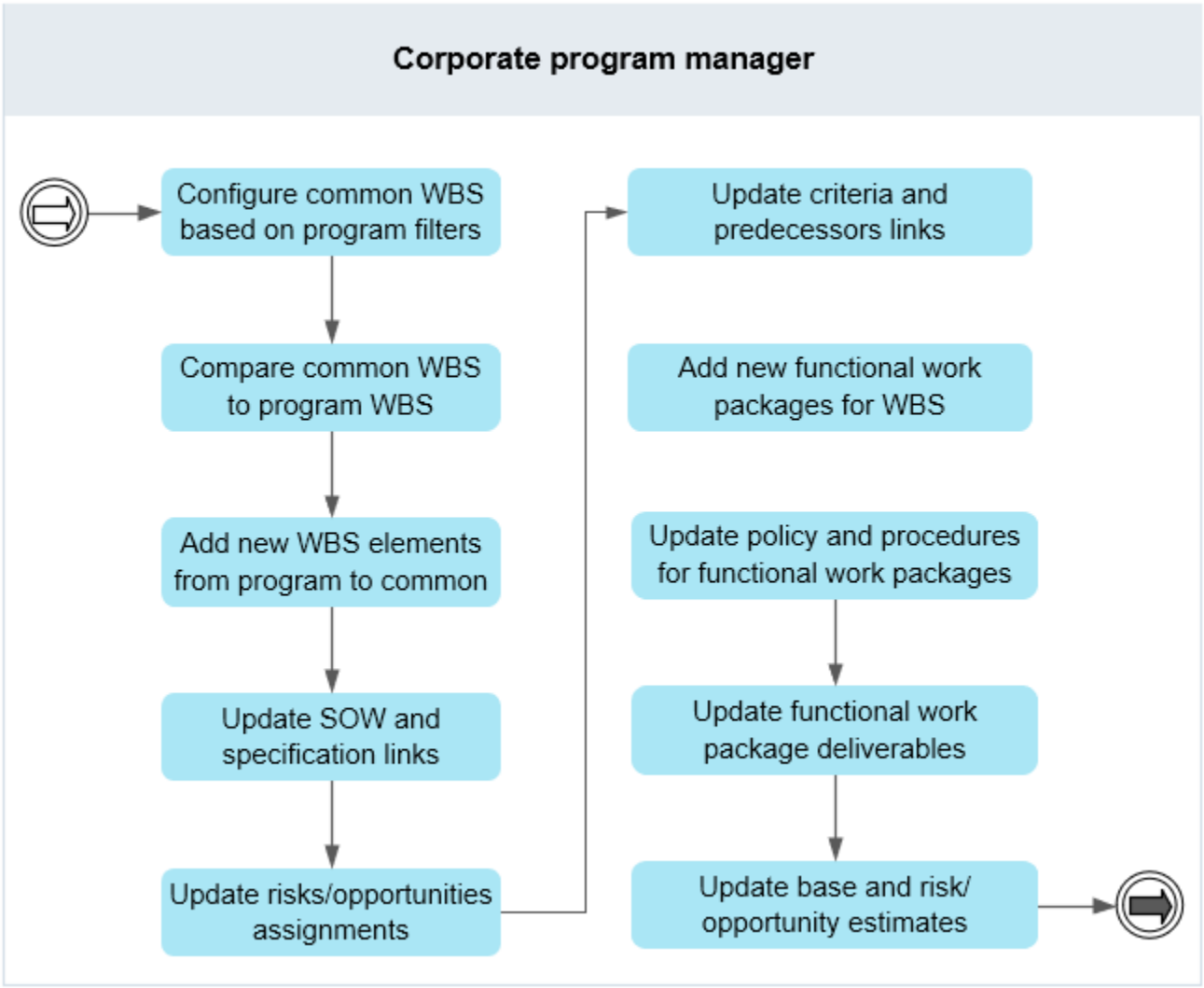
Understanding lessons learned

One of the key values of IPP&E is the availability of the common WBS, containing historical WBS data based on best practices and lessons learned from the past projects. After successful completion of a project, you incorporate the knowledge gained during the planning and execution into the common WBS for use by the new projects.

IPP&E allows you to compare the program WBS and all its related artifacts with similarly configured WBS data in the common WBS. Any difference found between the two is highlighted for subsequent action. You then update the common WBS with any new information from the program WBS, including any new WBS elements or changes to an existing WBS.

The lessons learned incorporation business process

The lessons learned business process is as follows:



Compare the common WBS and a program WBS

Compare the program WBS and a configured common WBS as follows:

1. Configure the common WBS by choosing **Configure** above the common WBS and selecting a variant rule.

2. Select the program WBS root and the configured common WBS root, and click **Compare**.

   The two WBSs are displayed side by side. The **Compare** dialog box is also displayed.

3. Compare the content of two structures by selecting different actions and then click **Compare**.

   The missing WBS elements are highlighted in the common WBS structure compared to the program WBS in red.

4. Add the missing elements to the common WBS by choosing **Add Missing Elements** from the toolbar above the structure tree of the common WBS.

   The system adds the missing elements to the common WBS.

A. IPP&E workflow handlers

Action handlers

IPP-add-successor-wbs-work-element

DESCRIPTION

This handler attaches a successor WBS element revision to a target WBS element revision with a **Psi0PredecessorWorkElement** relation. The predecessor must have the execution state specified in the **predecessor_execution_state** argument.

It ignores successors with the same execution state as the **disallowed_execution_state** argument value.

If these conditions are satisfied, it attaches the successor WBS element as the target.

SYNTAX

IPP-add-successor-wbs-work-element

ARGUMENTS

-predecessor_execution_state

Specifies the allowed execution state of the predecessor WBS work package.

-attachment

The attachment folder under which the successor WBS work packages will be copied. The predecessors of these WBS work packages should be released.

-disallowed_execution_state

Specifies the disallowed execution state of the successor WBS work packages.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-predecessor_execution_state	Created
-attachment	target
-disallowed_execution_state	Completed

In this example, the handler will find the successor work packages which do not have the execution state as **Completed** and all of its predecessor work packages with the execution state as **Created**. It will then add these successor work packages to the target attachment of the workflow.

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-adhoc-signoff

DESCRIPTION

This handler gets the assignee value of the dynamic participant of the type specified by the **-assignee_property** argument and then sets it as the adhoc signoff for a given task.

SYNTAX

IPP-adhoc-signoff

ARGUMENTS

-include_type

Specifies the object types used to get the property value.

-property

Refers to the object with the property used for adhoc signoff.

-assignee_property

This property exists on the object specified by the **–property** argument. It specifies the property read to get the dynamic participant user for the sign-off team.

PLACEMENT

Typically placed on the **Start** action of a **select-signoff-team** subtask.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-include_type	**Ipp0FuncWorkPkgRevision**
-property	**ipp0FuncOrg**
-assignee_property	**ipp0FuncOrgLead**

In this example, the dynamic participant assigned to the **ipp0FuncOrgLead** property is **Smith**. The handler assigns **Smith** as the signoff attachment to the **select-FWP-OBS-lead** task.

IPP-assign-dynamic-participant-from-OBS

DESCRIPTION

This handler iterates through the OBS structure attached to the target WBS root element. Notifies the OBS lead of each OBS element revision that the ETC process is initiated for the target WBS root element.

SYNTAX

IPP-assign-dynamic-participant-from-OBS

ARGUMENTS

This handler has no arguments.

PLACEMENT

Typically placed on the **Start** action.

RESTRICTIONS

None.

IPP-assign-dynamic-participant-from-property

DESCRIPTION

This handler creates dynamic participants based on the specified argument. It is typically used to create a dynamic participant for the next approving level OBS lead to send workflow notification.

SYNTAX

IPP-assign-dynamic-participant-from-property

ARGUMENTS

-for_type

The object type for which the dynamic participant is created and assigned.

-property

The reference property of the specified object type to look for.

-participant_type

The type of dynamic participant attached to the object referenced by the **-property** argument.

-participant_type_to_create

The dynamic participant to create.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

IPP-assign-wbs-work-element-owner

DESCRIPTION

This handler gets the owning user of a WBS work element, and assigns them as the sign-off member for the task to which this handler is attached.

SYNTAX

IPP-assign-wbs-work-element-owner

ARGUMENTS

This handler has no arguments.

PLACEMENT

Typically place on the **Start** action of a **select-signoff-team** subtask.

RESTRICTIONS

None.

EXAMPLES

A functional work package request is created for a WBS work package owned by user **wbs**. This action handler assigns **wbs** as a signoff attachment to the **select-signoff-team** task.

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-attach-dependent-fwps

DESCRIPTION

This handler finds all FWPs attached to a WBS work package and verifies that all independent FWPs have the specified estimation state property value. If so, it attaches dependent FWPs as targets to the workflow, so that it becomes the target for the sub-process.

SYNTAX

IPP-attach-dependent-fwps

ARGUMENTS

-allowed_state

One of the following states - Created, In Work, In Review or Approved.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

The following example finds whether independent FWPs **FWP1** and **FWP3** have the estimation state value specified in the **allowed_state** argument. If so, it attaches the dependent FWP (**FWP2**) as the target to the workflow.

FWP1 -Independent
FWP2 -Dependent on FWP1 and FWP3
FPW3 –Independent
allowed_status -Approved

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-attach-fwp-with-out-dated-OBS

DESCRIPTION

This handler finds and attaches all functional work package revisions *not* in a specified state whose functional organization is not part of any OBS structure. The functional organization of these functional work package revisions should be updated.

SYNTAX

IPP-attach-fwp-with-out-dated-OBS

ARGUMENTS

-attachment

The attachment folder under which the functional work package revisions with outdated OBSs are copied.

-exclude_fwp_states

One or more states to find - Created, In Work, In Review or Approved.

PLACEMENT

Requires no specific placement. Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

You can use the handler to find all the functional work package revisions with an outdated OBS that do not have one of the specified states (Created, In Work, In Review or Approved). The system copies all such functional work package revisions to the specified reference folder.

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-attach-independent-fwps

DESCRIPTION

This handler finds all the independent FWPs associated with a WBS work package and attaches them as targets to the workflow, so that it becomes the target for the sub-process.

SYNTAX

IPP-attach-independent-fwps

ARGUMENTS

This handler has no arguments.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-attach-related-wbs-element

DESCRIPTION

This handler finds the parent WBS work package of the target functional work package revision. It checks if all functional work package revisions have the property value specified in argument. If so, the system copies the parent WBS work package revision to the attachment folder.

SYNTAX

IPP-attach-related-wbs-element

ARGUMENTS

-attachment

The EPM folder type into which the system copies the WBS work package revisions.

-property

Either Estimation State or Execution State.

-value

The value of the specified property to check for each child FWP.

PLACEMENT

Typically placed on the **Start** action of any task.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-attachment	**reference**
-property	**ipp0ExecutionState**
-value	**In Review**

In this example, the handler checks the value of the **ipp0ExecutionState** property of all the functional work package revisions under the parent of the target functional work package revision, and if the value is **In Review**, then the handler will copy the parent work package revision as a reference attachment to the workflow process.

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-attach-related-wbs-of-task

DESCRIPTION

This handler attaches the latest revision of the WBS work package related to the given TaskInfo as target.

SYNTAX

IPP-attach-related-wbs-of-task

ARGUMENTS

This handler has no arguments.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-create-execution-info-from-task-info

DESCRIPTION

This handler creates a Task Execution Info object, typerefs the attached taskinfo to it, and then relates the created executionInfo to the FWP present as an attachment of the workflow.

SYNTAX

IPP-create-execution-info-from-task-info

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement. Typically placed on the **Start** or **Complete** action of a task.

RESTRICTIONS

None.

IPP-create-time-phased-estimate

DESCRIPTION

This handler creates or updates the distributed cost for each FWP depending on whether the distributed cost table is present on the FWP.

If the distributed cost table is not present, the handler generates a time phased estimate on the FWP for each TaskInfo present on the parent WBS work package.

If the distributed cost table is already present, this handler updates the distributed cost table. If the property is outdated, the distributed cost is true on the FWP.

SYNTAX

IPP-create-time-phased-estimate

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-delete-relation

DESCRIPTION

This handler deletes the specified relation between primary and secondary objects of the specified type.

SYNTAX

IPP-delete-relation

ARGUMENTS

-primary_type

The type of primary object.

-secondary_type

The type of secondary object.

-relation

The type of relation to delete.

PLACEMENT

Place on **Start** or **Complete** action.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-primary_type	**Wbs0RootRevision**
-secondary_type	**Psi0PrgDel**
-relation	**Wbs0ProgramDeliverables**

In this example, the handler deletes the **Wbs0ProgramDeliverables** relation between primary objects of type **Wbs0RootRevision** and secondary objects of type **Psi0PrgDel**.

IPP-export-taskinfo-spantime-to-external-schedule

DESCRIPTION

This handler updates the duration of external schedule task during the risk incorporation process.

SYNTAX

IPP-export-taskinfo-spantime-to-external-schedule

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

IPP-export-to-external-schedule

DESCRIPTION

This handler transfers the full project WBS structure to an external system for scheduling, typically when the external scheduling system is populated for the first time.

SYNTAX

IPP-export-to-external-schedule

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement. Typically placed on the **Start** or **Complete** action of a task.

RESTRICTIONS

None.

IPP-incorporate-contingency

DESCRIPTION

This handler incorporates contingency cost in the relevant cost elements of the direct cost on the functional work package revision. If the parent WBS work package and functional work package revision containing the direct cost to modify is released, they are revised and the cost is updated on the revised object.

SYNTAX

IPP-incorporate-contingency

ARGUMENTS

-contingency_type

The type of contingency, Risk or Opportunity.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-initiate-estimation-process

DESCRIPTION

This handler iterates through the target project WBS root revision. It initiates the workflow on all the WBS work elements with the same estimation state as specified in the **allowed_estimation_state** argument.

SYNTAX

IPP-initiate-estimation-process

ARGUMENTS

-allowed_estimation_state

The estimation state of the WBS work package, which may be Created, In Work, IPT Review or Approved.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-process-cost-update-form

DESCRIPTION

This handler:

- Iterates through the cost update request forms attached as targets.
- Obtains the values of the **Change Impact** and **Change Percentage** properties to identify whether to reduce cost or add cost based on the **release_status argument**.
- Decides whether to update the direct cost and distributed cost tables of all FWPs or only those of released FWPs.

If the **release_status** argument is set to **Approved**, only the direct cost and distribution cost tables for released FWPs are updated. If the **release_status** argument is set to **All**, the handler updates the cost of both released and WIP FWPs.

SYNTAX

IPP-process-cost-update-form

ARGUMENTS

-release_status

Approved or **All**.

PLACEMENT

Place on the **Start** or **Complete** action of the task.

RESTRICTIONS

None.

IPP-process-fwp-request-approval

DESCRIPTION

Iterates through the target functional work package requests and creates a functional work package for the functional organization mentioned in each functional work package request.

After the functional work package is created, the system attaches it under the WBS work package mentioned in the functional work package request.

Before the functional work package is actually created for the WBS work package, the system checks if a functional work package exists for the functional organization mentioned in the functional work package request. If it exists, the new functional work package is not created.

SYNTAX

IPP-process-fwp-request-approval

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement. Typically placed on the **Start** action of any task.

RESTRICTIONS

None.

EXAMPLES

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-revise-fwp-for-estimation

DESCRIPTION

This handler checks if the target FWP and its parent WBS work package are released. If so, it revises the FWP and/or the WBS, and then attaches the revised object as the target.

SYNTAX

IPP-revise-fwp-for-estimation

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-send-alert

DESCRIPTION

This handler sends the alert to the logged user. These alerts can be viewed in the **Alerts** section of the Active Workspace. Alert messages are configured in the handler argument.

SYNTAX

IPP-send-alert

ARGUMENTS

-subject

The subject text for the alert.

-msg_text

The message body text for the alert.

-object_from_attach

Specifies the type of attachment (**TARGET** or **REFERENCE**) from which the attached object should be selected.

-object_type

Specifies the object type which should be selected. Alert message will be sent with reference to this object.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-subject	Clone WBS Root
-msg_text	Clone WBS Root completed
-object_from_attach	REFERENCE
-object_from_attach	Wbs0RootRevision

In this example, the handler will find the **Wbs0RootRevision** objects from the **REFERENCE** attachments and send an alert with reference to this object with the alert subject **Clone WBS Root** and alert body text **Clone WBS Root completed**.

IPP-set-active-taskinfo

DESCRIPTION

This handler finds a WBS revision or resource package revision and Taskinfo or ExecutionInfo from the workflow attachment. It then typerefs the Taskinfo or ExecutionInfo onto the WBS revision or resource package revision.

SYNTAX

IPP-set-active-taskinfo

ARGUMENTS

-owning_object_type

The WBS work package or functional work package revision for which the **owning_object_property** will be set.

-owning_object_property

The property name to set.

-reference_object_type

The TaskInfo or ExecutionTaskInfo to set as the active task information.

-from_attach

The EPM attachment type for the specified objects, either target or reference.

PLACEMENT

Typically place on the **Start** or **Complete** action of a task.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-owning_object_type	**Wbs0ElementRevision**
-owning_object_property	**ipp0ActiveTaskInfo**
-reference_object_type	**Ipp0ExternalTaskInfo**
-from_attach	**target**

In this example, the handler will find objects of type **Wbs0ElementRevision** and **Ipp0ExternalTaskInfo** from **target** attachments and will set the **ipp0ActiveTaskInfo** property of the **Wbs0ElementRevision** type object to the **Ipp0ExternalTaskInfo** type object found from target attachments.

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-set-date-on-task-info

DESCRIPTION

This handler sets the current system date on the specified property.

SYNTAX

IPP-set-date-on-task-info

ARGUMENTS

-property

The property to set.

-include_type

The object type whose property will be set.

-attachment

The EPM attachment type for the specified objects, either target or reference.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

IPP-set-next-approving-level-from-IPT

DESCRIPTION

This handler sets the **Next Approving Level** attribute on the functional work package revision. It also creates an array of approvers (one of whom approved the task) as the **Estimation Approver** attribute.

For each target object, it does the following:

1. Checks if the target object is a WBS element revision.

2. Checks if the value of the **Next Approving Level** attribute is set or not.

   - If **Next Approving Level** is NULL, it gets the value of the IPT revision attached to the target WBS element revision with an **Applicable IPT** relation.

   - If **Next Approving Level** is already set, it:

     a. Gets the IPT revision attached to the WBS root of the WBS element revision with the **Applicable IPT** relation.

     b. Configures the IPT revision obtained in the previous step.

     c. Gets all the parents (IPTs) of the **Next Approving Level** IPT set on the target WBS element revision.

     d. Iteratively gets the IPT lead user for each parent IPT revision.

     e. Sets the **Next Approving Level** as any parent IPT revision whose IPT lead user is same as the current session user.

SYNTAX

IPP-set-next-approving-level-from-IPT

ARGUMENTS

-include_type

The object type whose property will be set.

-property

The property whose value will be set.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-set-next-approving-level-from-OBS

DESCRIPTION

This handler sets the value of the **Next Approving Level** attribute on a functional work package revision.

For each target object, it does the following:

1. Checks if the target object is a functional work package revision.
2. Checks if the value of the **Next Approving Level** attribute is set.
3. If **Next Approving Level** is NULL, it gets the value of the functional organization of the resource package revision, otherwise it skips this step.
4. If **Next Approving Level** is already set, the handler:
    a. Gets the OBS container attached to the WBS root of the resource package revision.
    b. Gets the revision and variant rules set in the **IPPRevisionRuleForOBSConfiguration** and **IPPVariantRuleForOBSConfiguration** preferences respectively.
    c. Configures the OBS container with the revision and variant rules obtained in the previous step.
    d. Get all the parents (OBS elements) of the next approving level OBS element set on the target resource package revision.
    e. Iteratively gets the functional organization lead user for each parent OBS element.
    f. Sets the next approving level as any of the parent OBS elements whose functional organization lead user is same as the current session user.
5. Creates an array of approvers (one of whom approved the task) for the **Estimation Approvers** property.

SYNTAX

IPP-set-next-approving-level-from-OBS

ARGUMENTS

-include_type

The object type whose property will be set.

-property

The property whose value will be set.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-set-originating-wbs-element-for-recovery

DESCRIPTION

This handler attaches a failed WBS task to the new WBS task that is created to allow recovery.

SYNTAX

IPP-set-originating-wbs-element-for-recovery

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

IPP-set-risk-on-mitigated-wbs-element

DESCRIPTION

This handler performs risk or opportunity mitigation where the risk or opportunity is attached using the source property on the target.

SYNTAX

IPP-set-risk-on-mitigated-wbs-element

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

IPP-set-value-to-null

DESCRIPTION

This handler sets the value of the specified property to NULL.

SYNTAX

IPP-set-value-to-null

ARGUMENTS

-include_type

The object type whose property will be set to NULL.

-property

The property to set to NULL.

-attachment

The EPM attachment type from which the objects are considered. This value may be target, reference or both.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Argument	Value
-include_type	**Wbs0ResourcePkgRevision**
-property	**ipp0NextApprovingLvl**
-attachment	**target**

In this example, the handler sets the **NULL** value to the **ipp0NextApprovingLvl** property of the Functional Resource Package Revision (**Wbs0ResourcePkgRevision**) object related as a target attachment to the workflow process.

IPP-set-wbs-element-completion-status

DESCRIPTION

This handler sets the Completion Status property on the TaskInfo attached as the target. It sets this property based on the Completion Status value of FWPs attached to the WBS work package.

SYNTAX

IPP-set-wbs-element-completion-status

ARGUMENTS

This handler has no arguments.

PLACEMENT

Typically placed on the **Start** or **Complete** action of a task.

RESTRICTIONS

None.

EXAMPLES

WBS1 has two FWPs, FWP1 and FWP2:

- If the completion status on both FWP1 and FWP2 is **Successful**, the handler sets the **Completion** status on the TaskInfo attached to the WBS element to **Successful**.
- If the completion status on FWP1 is **Successful** and FWP2 is **Failed**, the handler sets the **Completion** status on the TaskInfo attached to the WBS element as **Failed**.
- If the completion status on FWP1 is **Failed** and FWP2 is also **Failed**, the handler sets the **Completion** status on the TaskInfo attached to the WBS element as **Failed**.

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-update-ref-property-to-latest-rev

DESCRIPTION

This handler updates the OBS element revision specified on a functional work package with the latest revision.

It is used when the OBS element is revised after the functional work package is created and not approved.

SYNTAX

IPP-update-ref-property-to-latest-rev

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-read-and-saveas-wbs

Description

This handler is used for cloning a WBS structure.

Syntax

IPP-read-and-saveas-wbs

Arguments

This handler has no arguments.

Placement

Typically placed on the **Start** action of a task.

Restrictions

None.

Rule handlers

IPP-check-etc-process-on

DESCRIPTION

This handler checks if the target FWP revision's WBS root element is in the ETC process.

If the **Is ETC Process** property is **true** on the WBS root revision, it allows initiation of the workflow on the target functional work package revision.

SYNTAX

IPP-check-etc-process-on

ARGUMENTS

-Property

The WBS work package property whose value is checked.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-check-for-outdated-distributed-cost

DESCRIPTION

This handler traverses the entire program WBS structure and gives an error if any of the validations fails:

- If any of the WBS work packages does not contain external task information.
- If the distributed cost is not present on the functional work package revision.
- If an **Outdated Distributed Cost** property value is **true** on a functional work package. This value indicates the distributed cost is outdated and must be updated.

SYNTAX

IPP-check-for-outdated-distributed-cost

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

> Note:
>
> A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-check-initiator-from-property

DESCRIPTION

This handler checks if the session user is the same as the functional organization lead user for the target functional work package revision.

SYNTAX

IPP-check-initiator-from-property

ARGUMENTS

-Property

The reference property of the specified object type to check, for example, **ipp0FunctOrgLead**.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-check-obselement-in-structure

DESCRIPTION

This handler finds the item of the functional organization of the target functional work package revision. It then checks if any revisions of the functional organization item exists in the configured organizational breakdown structure.

If the item of the functional organization of the target functional work package revision does not exist in the configured organizational breakdown structure, it throws an error.

SYNTAX

IPP-check-obselement-in-structure

ARGUMENTS

This handler has no arguments.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-check-property-for-values

DESCRIPTION

This handler checks if the specified property of an object has one or more matching values as specified in the **values** argument.

SYNTAX

IPP-check-property-for-values

ARGUMENTS

-property

Specifies the properties to check.

-values

Specifies the required real values to check. This may be a single value or multiple values separated by commas (,).

-attachment

Specifies the type of attachment to check, either target or reference.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

You can use this handler to check the target object for the **Created** value of the **ippEstimationState** property.

IPP-check-target-count

DESCRIPTION

This handler verifies the number of target attachments, which should not be more than one.

SYNTAX

IPP-check-target-count

ARGUMENTS

This handler has no arguments.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

IPP-validate-dependent-fwps

DESCRIPTION

This handler checks if all dependent resource packages of the target functional work package revision have a value specified in the argument.

SYNTAX

IPP-validate-dependent-fwps

ARGUMENTS

-Allowed_state

The state of the functional work package revision to check for,which may be **Created**, **In Work**, **In Review** or **Approved**.

PLACEMENT

Typically placed on the **Start** action of a task.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-validate-next-approving-authority-for-fwp

DESCRIPTION

This handler validates if the current session user is a valid approving authority for the target functional work package revision as follows:

1. Checks if the target object is an FWP revision.
2. Checks if the next approving level is set.
3. Gets the functional organization lead user for the next approving level.
4. Finds the last approved functional organization lead user set on the estimation approvers property of the target resource package revision.
5. Gets the OBS container and its revision and variant rules. Configures the OBS container with the revision and variant rules.
6. From the configured hierarchy, finds all the parents of the last approved Functional organization lead user set on the estimation approver property of the target functional work package revision.
7. Gets the functional organization lead user for all the parent OBS elements.
8. Checks if the functional organization lead user for the next approving level user is the same as the functional organization lead user for the parent OBS element.
9. If step 8 is not satisfied, that is, none of the parent OBS elements has the next approving user as its functional organization lead user, the functional organization lead user for the next approving level is not the correct authority for approving the functional work package revision.

SYNTAX

IPP-validate-next-approving-authority-for-fwp

ARGUMENTS

-project_phase

Determines the target type. If the value is **Estimation**, the target type is functional work package revision. If the value is **Execution**, the target type is **Task Execution Info**.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.

IPP-validate-next-approving-authority-for-wbs-elem

DESCRIPTION

This handler validates if the current session user is a valid approving authority for the target WBS element revision as follows:

1. Checks if the target object is a WBS element revision.
2. Checks if the next approving level is set, then gets the IPT lead user for the next approving level.
3. If the next approving level is already set, gets the IPT root.
4. Finds the last approved user set on the estimation approvers property of the target WBS revision.
5. Configures the IPT root with the **Latest Working** revision rule.
6. Find all the parents of the IPT on the target WBS element from the configured IPT root node.
7. Configures the IPT revision.
8. Gets the IPT lead user for the parent IPT.
9. Checks if the session user is the same as the IPT lead user for the parent IPT.
10. If step 9 is not satisfied, that is, none of the parent IPTs has the current session user as its IPT Lead user, the current session user is not the correct authority for approving the WBS element revision.

SYNTAX

IPP-validate-next-approving-authority-for-wbs-elem

ARGUMENTS

-project_phase

Determines the target type. If the value is **Estimation**, the target type is WBS work package revision. If the value is **Execution**, the target type is **External Task Info**.

PLACEMENT

Requires no specific placement.

RESTRICTIONS

None.

EXAMPLES

Note:

A WBS work package is identified based on the **Work Package** check box on the WBS element revision.