**Designing Process Templates**

This document provides guidelines for a Mechanica expert to design process templates as per the design and simulation requirements of the model. The design engineers can use these templates as guidelines to perform the similar kind of analysis.

The document covers the following topics:

**Topic**

About Process Templates

[Template Structure](#_23DesigningProcessTemplates_2D_302B6A_303C_2FTemplateStructureYouCanDesignAProc_2D_303BF6194)

[Sample Template](#_23DesigningProcessTemplates_2D_302B6A_303C_2FSampleTemplateTheFollowingSampleTe_2D_303C_30115C)

[Language-Specific Limitations](#_23DesigningProcessTemplates_2D_302B6A_303C_2FLanguage_2DSpecificLimitationsMechan_2D_303C_303_309C)

[Object Support in Native and FEM mode](#_23DesigningProcessTemplates_2D_302B6A_303C_2FObjectSupportInNativeAndFEMModesTh_2D_303C_304BF4)

[Model Object and Attribute](#_23DesigningProcessTemplates_2D_302B6A_303C_2FModelObjectAndAttributeToDefineATa_2D_303C_30674C)

**About Process Templates**

You can create [process templates](D:\\TMP\\.._proc_template.dita) for [Process Guide](D:\\TMP\\.._top.dita) that guide user’s through a set of Mechanica tasks. You can design separate templates according to the simulation requirements of your model. In designing a template, you should consider:

● the design requirements of your model and its components

● the actual geometry of a model or component under consideration

● the model’s real world surroundings such as loads and constraints

● type of analysis to perform such as static, modal, prestress, buckling, and vibration analyses

● the way you want to evaluate the results

● your design goal

Mechanica allows you to create process templates in any language provided you use the proper encoding in the template header corresponding to the language-specific characters in the template. Consider these [limitations](D:\\TMP\\pg_design.dita) while designing templates in languages other than English.

**Template Structure**

You can design a process template in XML format (Extensible Markup Language). It follows a predefined structure and uses a certain set of tags to define all the tasks involved in any model analysis.

The template exposes a flexible tree-based object model whose hierarchy depends on the order in which the tasks are defined. A template should include the following components:

● Header provides an identification line that is used to recognize the process template.

● Process encloses the set of Mechanica tasks involved in any simulation process. The task list follows a particular sequence based on the [Mechanica workflow](D:\\TMP\\.._work.dita) but may not always include all the tasks.

A template must contain a set of tags in the flow as shown in the following example:

<file name>  
  
<header product="GuidingUI" fileVersion="1.0" />  
  
<process id="1" name="------------">  
<task id="1" object="-------" label="----------">  
<description>------------- </description>  
</task>  
<task id="2" object="-------" label="----------">  
<description>------------- </description>  
</task>  
  
.  
  
.  
  
.  
  
<task id="n" object="-------" label="----------">  
<description>-------------- </description>  
</task>  
</process>  
</file name>

You can refer to a sample template, staticAnalysis.xml, available in the system path $PTCSRC/mech/html/process\_guide for details. It includes separate templates for Native and FEM mode.

**Header Element**

The process templates must contain the <ProcessGuideProcess> elements to define a header of the process template.

The following example shows a fragment of the XML schema from the sample template. This fragment shows header information that you can use to describe the template.

<ProcessGuideProcess>  
<header product="GuidingUI" fileVersion="1.0" />

**Process Element**

The process templates defines a simulation process using the process element. The process element provides details about the template with the name attribute. Consider the analysis type or the model name as an input value for the name attribute. For example, <process id="1" name="OilPanTemplate">.

You must enclose all the tasks involved a simulation process in the <process> tag. A process is described through a list of tasks as follows:

<process id="1" name="StaticAnalysis">  
  
<task id, label, object, type, visibility, name, ...>  
  
<description>Text, actionlink(1), Text, infolink(0+), Text</description>  
  
</task>  
  
<task id, object, type, name, ...>  
  
<description>Text, actionlink(1), Text, infolink(0+), Text</description>  
  
</task>  
  
.  
  
.  
  
.  
  
<task>....</task>  
  
</process>

**Task**

For the simulation of a model, you need to perform various tasks that follow a particular order. In the process template, all the tasks except nested tasks are associated with an object and are identified by a unique number. A nested task is a parent task containing subtasks where subtasks may or may not have an object attribute and are used only to provide information.

The task function provides details about the operation to be performed on an object.For example, <task id="2" object="Info" label="General Info">.

A task must contain the following attributes:

● The id attribute assigns a unique number to each task. You can list the tasks based on the order in which the tasks need to be performed in the process of a model simulation.

● The object attribute specifies a model object associated with the task. Although Mechanica supports most of the Structure objects in native and FEM modes, a few objects are supported only in a single mode. For details, refer to [Object Support in Native and FEM mode](D:\\TMP\\pg_design.dita).

● The label attribute is used to identify the task. The value that you specify for the label attribute appears as a separate step in the Navigation panel of the [Process Guide dialog box](D:\\TMP\\.._dialog.dita). You can specify label in languages other than English using the proper [encoding type](D:\\TMP\\pg_design.dita) in the template header.

While defining dependent tasks that have a dependency on other tasks, you need to define additional attributes than the standard id and label attributes and can link to the related task through its TaskID where TaskID is the value of the id attribute of the corresponding task. For example, while defining the Load task, the loadset attribute allows you to link to previously defined LoadSet through its TaskID.

You can choose to hide certain tasks in a process template by setting the value of the visibility attribute to OFF. The tasks for which the visibility attribute is set to OFF, will be executed without them appearing in the task list of the **Process Guide** dialog box. For example, you can set the value of the visibility attribute to OFF for the LoadSet object task so that the specified LoadSet will be automatically created without using the user interface.

You may need to specify a number of additional attributes than the standard set of attributes based on the task object. For details about the objects and their corresponding attributes refer to [Model Object and Attributes](D:\\TMP\\pg_design.dita). Make sure that you define all the necessary attributes while defining a task for an object.

**Description**

The description attribute provides detail about the current task. You can provide some practical information using this attribute such as tips, tricks, values to be applied, model references to be selected while executing the current task.

For example,

<description>Create a Total load at Point using <actionlink>this dialog</actionlink>simulate the transmission of torque through the crankshaft. <br />   
the load of 400,000 mmN of torque in the x-direction to the circular cross-sectionthe crank. The magnitude is calculated at the center of the shaft, PNT10. <br  
>   
name of the constraint is specified by the process template ("torque"). View  
<infolink href="http://rdweb.ptc.com/promec/start.htm">help</infolink> </description>

The information provided for description appears in the Instruction panel of the [Process Guide dialog box](D:\\TMP\\.._dialog.dita). You can provide an <action link> as well as an <information link>. For each visible task in the Navigation panel, you must provide an action link for a user to invoke the corresponding dialog box and complete the operation. The text that you include in the <action link> tags is linked to the corresponding dialog box and appears as hyperlink in the Instruction Panel of the **Process Guide** dialog box.

When you complete all the activities related to the dialog box, **Process Guide** records your action and marks the status of the current step as complete. To know when the Process Guide marks that status of the task as complete, refer to the Description column of the table in the Model Object and Attributes section. You can use <information link> to point users to the local documentation.

Instead of text, you can also include images in an <action link> and <information link>. You must use the image attribute to include image in the template.

For example,

<task id="2" Object= "LoadSet" Label="Loadset" name="MyLoadSet">  
  
 <description>   
  
 Define a <actionlink image="d:\test\_picture.gif" > loadset. </actionlink>  
  
 </description>  
  
</task>

The image is linked to the corresponding dialog box and appears as hyperlink in the Instruction Panel of the **Process Guide** dialog box.

A parent task must be an information task and consists of sub-tasks. In other words, a parent task cannot have an action link but its sub-tasks can include an action link and information links. Process Guide marks the status of the main task as complete only after all the sub-tasks are complete.

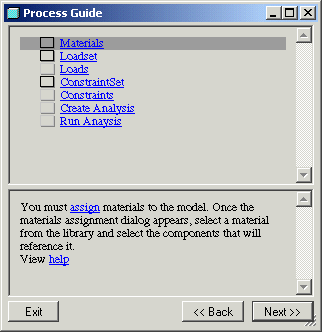
You can specify description in languages other than English using the proper [encoding type](D:\\TMP\\pg_design.dita) in the template header.

**Sample Template**

The following sample template is designed to conduct the static analysis of a model.

<ProcessGuideProcess>  
  
<header product="GuidingUI" fileVersion="1.0"></header>  
  
<!-- Process Definition for StaticAnalysis -->  
  
<process id="1" name="StaticAnalysis">  
  
<!-- Assign or create materials -->  
  
<task id="0" Object="Material" Label="Materials">  
  
<description> You must <actionlink>assign</actionlink> materials to the model. Oncematerials assignment dialog appears, select a material from the library and selectcomponents that will reference it. <br></br>View <infolink href="http://rdweb.ptc.com/dock01/html/usascii/proe/default.htm?promec/start.htm">help</infolink></description>  
  
</task>  
  
<!-- Use default loadset -->  
  
<task id="2" Object="LoadSet" Label="Loadset" name="MyLoadSet">  
  
<description> Define a <actionlink> loadset. </actionlink></description>  
  
</task>  
  
<!-- Create new load in current loadset -->  
  
<task id="3" Object="Load" Label="Loads" Name="LoadOne" type="Force" LoadSet="2">  
  
<description> Define <actionlink> loads </actionlink> that will be applied to the. Multiple loads can be created. After you have completed defining the loads,OK from the dialog.</description>  
  
</task>  
  
<!-- Create a new Constraint Set -->  
  
<task id="4" Object="ConstraintSet" Label="ConstraintSet" name="MyConstraintSet">  
  
<description> Define the <actionlink> constraints </actionlink> sets.</description>  
  
</task>  
  
<!-- Create a new contraint in the Constraint Set task -->  
  
<task id="5" Object="Constraint" Label="Constraints" Name="MyConstraint" ConstraintSet="4">  
  
<description> Define the <actionlink> constraints </actionlink> that will be appliedthe model. Multiple constraints can be created. After you have completed definingconstraints, select OK from the dialog. </description>  
  
</task>  
  
<!-- Creates/Edit MeshControl Task with the name -->  
  
<task id="6" Object="AutoGEMControl" Label="AutoGEMControl" name="MyMeshCtrl">  
  
<description> <actionlink> Execute </actionlink> the AutoGEMControl.</description>  
  
</task>  
  
<!-- Creates/Edit Mesh Task -->  
  
<task id="7" Object="AutoGEM" Label="AutoGEM">  
  
<description> <actionlink> Execute </actionlink> the AutoGEM mesh.</description>  
  
</task>  
  
<!-- Create a new analysis with the LoadSet specified in task 2 and ConstraintSettask 4 -->  
  
<task id="8" Object="Analysis" Label="Create Analysis" Name="MyAnalysis" type="Static">  
  
<description> Create a new <actionlink> analysis </actionlink> with default name.</description>  
  
<reference Object="LoadSet">  
  
<id>2</id>  
  
</reference>  
  
<reference Object="ConstraintSet">  
  
<id>4</id>  
  
</reference>  
  
</task>  
  
<!-- Run the analysis created in task 6 -->  
  
<task id="9" Object="Run" Label="Run Anaysis" analysis="6">  
  
<description> <actionlink> Execute </actionlink> the analysis, this may take some.</description>  
  
</task>  
  
<!-- Show results for run of the analysis created in task 6 -->  
  
<task id="10" Object="ResultTemplate" Label="View Results" analysis="6" visibility="off"="TestTemplate.rwt" mode="combine">  
  
<description> <actionlink> View </actionlink> results for run of the analysis.</description>  
  
</task>  
  
</process>  
  
</ProcessGuideProcess>

A design engineer can use this template to create a new Process Guide session. When you create a session, Mechanica invokes the **Process Guide** dialog box as shown in the figure below:



All the steps listed in the **Process Guide** dialog box are based on the associated template. The upper section of the **Process Guide** dialog box is the Navigation panel while the lower section of the dialog box is the Instruction panel.

If you try to associate the steps defined in the template with the steps listed in the dialog box, you can notice that:

● The navigation panel displays the tasks in the same order as defined in the template.

● The name of the task displayed in the Navigation panel is according to the value specified for the label attribute.

● The text displayed in the Instruction panel is according to the information provided for the description attribute of a task.

● The tasks for which the value of the visibility attribute is set to Off are not displayed in the Navigation panel. For example, see task 10 in the sample for the ResultTemplate object. Process Guide does not display this task in the Navigation Panel.

● The tasks that have a dependency on other tasks refer to those related tasks using the reference object and the TaskID. For example, task 8 in the sample defined for the object Analysis uses the reference objects to link to the corresponding loadset and constraintset tasks through their task ids.

● The tasks for which the type attribute is defined, Process Guide invokes the corresponding dialog box. For example, task 3 in the sample for the Load object specifies the value of the type attribute as force. The description attribute for this task provides an action link for a user to invoke the Force/Moment Load dialog box.

If you do not specify the type attribute while defining the task for the Load object, Process Guide invokes the [Loads Manager](D:\\TMP\\.._manage_loads.dita) dialog box for managing loads in the model. In addition to the Load object, the Constraint, Material, and MaterialOrientation objects observe the same rule while defining the type attribute.

● To define certain tasks, you need to specify more than the standard set of attributes, that is, id, label, and description. For example, task 8 in the sample for the ResultTemplate object specifies the template, mode, analysis, and visibility attributes.

For details about the Process Guide user interface, refer to [Process Guide](D:\\TMP\\.._dialog.dita) dialog box.

**Language-Specific Limitations**

Mechanica allows you to create process templates in any language. While defining a task in a process template, you can add a description and label for that task in any language that Mechanica supports. Pro/ENGINEER supports only a limited number of language encoding types. If your template contains the task description or label in a language other than English, you need to specify the encoding in the file header. Valid values for encoding the Japanese language are SJIS (on Windows only) and UTF-8.

There are third party tools to convert encoding types, for example, iconv on UNIX. To set the encoding type in a process template file containing Japanese characters, add the following line at the top of the XML template file:

<?xml version="1.0" encoding="SHIFT\_JIS"?>

or

<?xml version="1.0" encoding="UTF-8"?>

**Object Support in Native and FEM Modes**

The following table lists objects and their supported modes:

| **Object** | **Type** | **native**  **(P Mode)** | **FEM (H Mode)** |
| --- | --- | --- | --- |
| Model Type | Pstructural | x |  |
| Hstructural |  | x |
| Info |  | x | x |
| Mapkey |  | x | x |
| UnitInfo |  | x |  |
| DatumPoint |  | x | x |
| DatumCurve |  | x | x |
| DatumPlane |  | x | x |
| CoordSystem |  | x | x |
| DatumAxis |  | x | x |
| SurfaceRegion |  | x | x |
| VolumeRegion |  | x | x |
| Shell |  | x | x |
| ShellProperty |  | x | x |
| Midsurface |  | x | x |
| Beam |  | x | x |
| BeamSection |  | x | x |
| BeamOrientation |  | x | x |
| BeamRelease |  | x | x |
| Mass |  | x | x |
| MassProperty |  | x | x |
| Spring |  | x | x |
| SpringProperty |  | x | x |
| Constraint | Displacement | x | x |
| Symmetry | x |  |
| AlongSurface |  | x |
| ConstraintSet |  | x | x |
| Load | Force | x | x |
| Pressure | x | x |
| Bearing | x |  |
| Gravity | x | x |
| Centrifugal | x | x |
| GlobalTemperature | x | x |
| MEC/TTemperature | x |  |
| ExternalTemperature | x |  |
| StructuralTemperature | x | x |
| Preload | x |  |
| LoadSet |  | x | x |
| Material |  | x | x |
| MaterialOrientation | Surface | x | x |
| Part | x |  |
| Measure |  | x |  |
| AutoGEM |  | x |  |
| AutoGEMControl |  | x |  |
| AutoGEMSettings |  | x |  |
| Mesh |  |  | x |
| MeshControl |  |  | x |
| Analysis | Static | x | x |
| Modal | x | x |
| Buckling | x |  |
| Fatigue | x |  |
| PrestressStatic | x |  |
| PrestressModal | x |  |
| DynamicTime | x |  |
| DynamicFrequency | x |  |
| DynamicShock | x |  |
| DynamicRandom | x |  |
| Run |  | x | x |
| RunSettings |  | x |  |
| ResultTemplate |  | x | x |

**Model Object and Attribute**

To define a task for any model object, you must specify attributes specific to that model object. The model objects and their attributes used in various modeling and analysis activities are categorized as follows:

● [General Modeling](D:\\TMP\\pg_design.dita)

● [Simulation Features](D:\\TMP\\pg_design.dita)

● [Modeling Entities](D:\\TMP\\pg_design.dita)

● [Idealizations](D:\\TMP\\pg_design.dita)

● [Properties](D:\\TMP\\pg_design.dita)

● [Meshes](D:\\TMP\\pg_design.dita)

● [Analyses and Results](D:\\TMP\\pg_design.dita)

**General Modeling**

You can define tasks associated with the following modeling objects in the process template:

● [ModelType](D:\\TMP\\pg_design.dita)

● [Info](D:\\TMP\\pg_design.dita)

● [UnitInfo](D:\\TMP\\pg_design.dita)

The following table lists the object and its attributes for defining general modeling tasks:

| **Object** | **Attributes** | **# of values** | **Valid Values** | **Description** |
| --- | --- | --- | --- | --- |
| ModelType | id | 1 | integer | Specifies the model as a native mode model or FEM mode model. If you use this object, Mechanica restricts the template to the specified mode only, and you cannot use the template in the other mode. Considerations:  ● Add this task at the top of the task list.  ● A template can have only one task with a ModelType object.  ● This task does not invoke a specific UI action but specifies that the configuration file is for a specific mode.  ● If this task is defined and the model type is specified in a template, error checking controls the import of the configuration file.  ● This task also allows validation of tasks and their attributes in a template. |
| type | 1 | PStructural, HStructural, |  |
| visibility | 1 | Off | Process Guide completes the task without displaying it in the Navigation panel of the **Process Guide** dialog box. |
| Info | id | 1 | integer | Specifies the details about the current process template.  Considerations:  ● The task is complete when Process Guide displays the information specific to this template. |
| label | 1 | text |  |
| description | 1 | text, executable descriptor | The task cannot contain any action link to display in the Information panel of the **Process Guide** dialog box. |
| Mapkey | Id | 1 | Integer | The action for this object is to run an existing mapkey with the given name |
| Label | 1 | text |  |
| name | 1 | text | Key-sequence of the mapkey |
|  | description | 1 | text, executable descriptor |  |
| UnitInfo | id | 1 | integer | Displays the system of units.  Considerations:  ● The task is complete when Process Guide displays the unit information. |
| label | 1 | text |  |
| description | 1 | text, executable descriptor |  |

**Simulation Features**

You can define tasks associated with the following simulation features in the process template:

● [DatumPoint](#_23DesigningProcessTemplates_2D_302B6A_303C_2FDatumPoint_2D_3043_3028AC)

● [DatumCurve](#_23DesigningProcessTemplates_2D_302B6A_303C_2FDatumCurve_2D_3035495B9)[pg\_design.dita](D:\\TMP\\pg_design.dita)

● [DatumPlane](#_23DesigningProcessTemplates_2D_302B6A_303C_2FDatumPlane_2D_30354A171)[pg\_design.dita](D:\\TMP\\pg_design.dita)

● [CoordSystem](#_23DesigningProcessTemplates_2D_302B6A_303C_2FCoordSystem_2D_30354B111)[pg\_design.dita](D:\\TMP\\pg_design.dita)

● [DatumAxis](#_23DesigningProcessTemplates_2D_302B6A_303C_2FDatumAxis_2D_30354C_30B1)[pg\_design.dita](D:\\TMP\\pg_design.dita)

● [SurfaceRegion](D:\\TMP\\pg_design.dita)

● Volume regions using the following different creation methods:

○ [VolumeRegionExtrude](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionExtrude_2D_303456AE9)[pg\_design.dita](D:\\TMP\\pg_design.dita)

○ [VolumeRegionRevolve](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionRevolve_2D_303457A89)

○ [VolumeRegionSweep](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionSweep_2D_303458A29)

○ [VolumeRegionBlend](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionBlend_2D_3034599C9)

○ [VolumeRegionUseQuilt](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionUseQuilt_2D_30345A581)

○ [VolumeRegionVarSectionSweep](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionVarSectionSweep_2D_30345BCF1)

○ [VolumeRegionSweptBlend](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionSweptBlend_2D_30345C8A9)

○ [VolumeRegionHelicalSweep](#_23DesigningProcessTemplates_2D_302B6A_303C_2FVolumeRegionHelicalSweep_2D_30345D849)

Consider the following points when you define the task associated with any simulation feature object:

● The task sets menus ready for creating the corresponding simulation feature.

● The task is complete when a user creates one or more simulation features. Process Guide associates these newly created simulation features with the task.

● When the user deletes all the simulation features associated with the task, Process Guide automatically marks the task as not valid.

The following table lists the object and its attributes for defining simulation feature tasks:

| **Object** | **Attribute** | **# of values** | **Valid Values** | **Description** |
| --- | --- | --- | --- | --- |
| DatumPoint | id | 1 | integer | Create [datum points](D:\\modstr\\features\\reference\\datpt_top.dita) on your model. |
| label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |
| DatumCurve | id | 1 | integer | Create [datum curves](D:\\TMP\\.._top.dita) on your model. |
| label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |
| DatumPlane | id | 1 | integer | Creates [datum planes](D:\\modstr\\features\\reference\\datplane_top.dita) on your model. |
| label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |
| CoordSystem | id | 1 | integer | Create [coordinate systems](D:\\TMP\\.._sys_types.dita) on your model. |
| label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |
| DatumAxis | id | 1 | integer | Creates [datum axis](D:\\TMP\\.._top.dita) on your model. |
|  | label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |
| SurfaceRegion | id | 1 | integer | Create [surface regions](D:\\TMP\\.._reg_top.dita). |
| label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |
| VolumeRegionExtrude  VolumeRegionRevolve  VolumeRegionSweep  VolumeRegionBlend  VolumeRegionUseQuilt  VolumeRegionVarSectionSweep  VolumeRegionSweptBlend  VolumeRegionHelicalSweep | id | 1 | integer | Create [volume regions](D:\\TMP\\.._reg_top.dita) based on the different volume creation methods. You can create a volume region task using these objects. |
| label | 1 | text |  |
| description | 1 | text,  executable descriptor |  |

**Modeling Entities**

You can define tasks associated with the following modeling entity objects in the process template:

● [Constraint](D:\\TMP\\pg_design.dita)

● [ConstraintSet](D:\\TMP\\pg_design.dita)

● [Load](D:\\TMP\\pg_design.dita)

● [LoadSet](D:\\TMP\\pg_design.dita)

● [Measure](D:\\TMP\\pg_design.dita)

Consider the following points when you specify the modeling entity object tasks:

● The task enables creation of the corresponding modeling entity object. The task is complete when a user creates one or more modeling entity objects. Process Guide automatically creates the ConstraintSet or LoadSet if it is not available in the model.

● If the task is to edit a specific modeling entity, the task is complete when a user invokes the corresponding dialog box and closes it after making changes.

● Process Guide associates this created or edited entities with the task. When a user deletes all the entities associated with the task, Process Guide marks the task as not valid.

Consider the following points when you specify the name attribute for the modeling entity object tasks:

● If a user specifies a new name, Mechanica creates a new modeling entity with the specified name.

● If a user creates a modeling entity without specifying any name, Mechanica creates a new modeling entity with the default name.

● If a user specifies the name of an existing modeling entity, Process Guide edits the existing modeling entity with the specified name.

Consider the following points when you specify the type attribute for the modeling entity object tasks:

● If you do not specify the type attribute while defining the tasks associated with the Constraint and Load objects, **Process Guide** invokes the [Constraints Manager dialog box](D:\\TMP\\.._manage_constraints.dita) or [Loads Manager dialog box](D:\\TMP\\.._manage_loads.dita) for a user to create one or more constraints or loads

● If the task specifies the type for constraint or load, **Process Guide** invokes the dialog box to create the specified type of constraint or load.

The following table lists the object and its attributes to define tasks for creating modeling entities:

| **Object** | **Attributes** | **# of values** | **Valid Values** | **Description** |
| --- | --- | --- | --- | --- |
| Constraint | id | 1 | integer | Define [constraints](D:\\TMP\\.._top.dita) for your model. |
| label | 1 | text |  |
| type | 0,1 | Displacement,  Symmetry,  AlongSurface |  |
| constraintset | 0,1 | TaskID |  |
| name | 0,1 | ConstraintName |  |
| description | 1 | text, executable descriptor |  |
| ConstraintSet | id | 1 | integer | Group various constraints in a [constraint set](D:\\TMP\\.._under.dita) that act together, and at the same time, on your model. |
| label | 0|1 | text | Specify this attribute, if the value of the visibility attribute is set to ON. |
| visibility | 0, 1 | On, Off | Process Guide displays the ConstraintSet by default. If the task specifies the value of this attribute to Off, Process Guide completes the task without displaying it in the Navigation panel of the **Process Guide** dialog box. |
| name | 0, 1 | ConstraintSetName |  |
| description | 0|1 | text, executable descriptor | Specify this attribute, if the visibility attribute is set to ON. |
| Load | id | 1 | integer | Define loads on your model. |
| label | 1 | text |  |
| type | 0, 1 | Force, Pressure, Bearing, Gravity, Centrifugal, GlobalTemperature, MEC/TTemperature, ExternalTemperature, StructuralTemperature, Preload |  |
| loadset | 0, 1 | TaskID |  |
| name | 0, 1 | LoadName |  |
| description | 1 | text, executable descriptor |  |
| LoadSet | id | 1 | integer | Group various loads in a [load set](D:\\TMP\\...dita) that act together on your model. |
| label | 0|1 | text | Specify this attribute, if the value of the visibility attribute is set to ON. |
| visibility | 0,1 | On, Off | The LoadSet is displayed by default in the **Process Guide** dialog box. If the task sets the value of this attribute to Off, Process Guide completes the task without displaying it in the Navigation panel of the **Process Guide** dialog box. |
| name | 0,1 | LoadSetname |  |
| description | 0|1 | text, executable descriptor | Specify this attribute, if the visibility attribute is set to ON. |
| Measure | id | 1 | integer | Create [measures](D:\\TMP\\.._top_s.dita). |
| label | 1 | text |  |
| name | 0, 1 | MeasureName |  |
| description | 1 | text, executable descriptor |  |

**Idealizations**

You can define tasks associated with the following idealization objects in the process template:

● [Shell](D:\\TMP\\pg_design.dita)

● [Midsurface](D:\\TMP\\pg_design.dita)

● [Beam](D:\\TMP\\pg_design.dita)

● [Mass](D:\\TMP\\pg_design.dita)

● [Spring](D:\\TMP\\pg_design.dita)

Consider the following points when you specify the idealization object tasks:

● Process Guide associates the created or edited idealization with the task.

● The task is complete when creation or editing of idealization is complete.

● When a user deletes the idealization associated with the task, Process Guide marks the task as not valid.

Consider the following points when you specify the name attribute for the idealization object tasks:

● If a user specifies a new name for idealization, Mechanica creates a new idealization with the specified name.

● If a user creates an idealization without specifying any name, Mechanica creates a new idealization with the default name.

● If a user specifies the name of an existing idealization, Process Guide edits the existing idealization with the specified name.

The following table lists the object and its attributes to define tasks for creating modeling entities and idealizations:

| **Object** | **Attributes** | **# of values** | **Valid Values** | **Description** |
| --- | --- | --- | --- | --- |
| Shell | id | 1 | integer | Create [shells](D:\\TMP\\.._top.dita) for your model. |
| label | 1 | text |  |
| name | 0, 1 | ShellName |  |
| description | 1 | text, executable descriptor |  |
| Midsurface | id | 1 | integer | Creates a [midsurface](D:\\TMP\\.._pair_defn_dialog.dita) by pairing two surfaces. |
| label | 1 | text |  |
| name | 0, 1 | ShellPairName |  |
| description | 1 | text, executable descriptor |  |
| Beam | id | 1 | integer | Create [beams](D:\\TMP\\.._top.dita) as model idealization. |
| label | 1 | text |  |
| name | 0, 1 | BeamName |  |
| description | 1 | text, executable descriptor |  |
| Mass | id | 1 | integer | Creates [mass](D:\\TMP\\.._top.dita) as an idealization that you can use to represent a concentrated mass without a specified shape. |
| label | 1 | text |  |
| name | 0, 1 | MassName |  |
| description | 1 | text, executable descriptor |  |
| Spring | id | 1 | integer | Create [springs](D:\\TMP\\.._top.dita) in your model. |
| label | 1 | text |  |
| name | 0, 1 | SpringName |  |
| description | 1 | text, executable descriptor |  |

**Properties**

You can define tasks associated with the following properties in the process template:

● [ShellProperty](D:\\TMP\\pg_design.dita)

● [BeamSection](D:\\TMP\\pg_design.dita)

● [BeamOrientation](D:\\TMP\\pg_design.dita)

● [BeamRelease](D:\\TMP\\pg_design.dita)

● [MassProperty](D:\\TMP\\pg_design.dita)

● [SpringProperty](D:\\TMP\\pg_design.dita)

● [Material](D:\\TMP\\pg_design.dita)

● [MaterialOrientation](D:\\TMP\\pg_design.dita)

Consider the following points when you specify the property object tasks:

● Process Guide associates the newly created or edited property with the task.

● The task is complete when creation or editing of property is complete.

● When a user deletes the property associated with the task, Process Guide marks the task as not valid.

Consider the following points when you specify the name attribute for the property object tasks:

● If a user specifies a new name for property, Mechanica creates a new property with the specified name.

● If a user creates a property without specifying any name, Mechanica creates a new property with the default name.

● If a user specifies the name of an existing property, Process Guide edits the existing property with the specified name.

The following table lists the object and its attributes for defining tasks to create model properties:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Object** | **Attributes** | **# of values** | **Valid Values** | **Description** |
| ShellProperty | id | 1 | integer | Defines a [shell property](D:\\TMP\\.._props_top.dita) to a face, region, or datum surface. |
| label | 1 | text |  |
| name | 0, 1 | ShellPropertyName |  |
| description | 1 | text, executable descriptor |  |
| BeamSection | id | 1 | integer | Defines a [beam section](D:\\TMP\\.._sec.dita). |
| label | 1 | text |  |
| name | 0, 1 | BeamSectionName |  |
| description | 1 | text, executable descriptor |  |
| BeamOrientation | id | 1 | integer | Specifies the [beam orientation](D:\\TMP\\.._orient.dita). |
| label | 1 | text |  |
| name | 0, 1 | BeamOrientationName |  |
| description | 1 | text, executable descriptor |  |
| BeamRelease | id | 1 | integer | Specifies the degrees of freedom you want to [release](D:\\TMP\\.._rel.dita) for a beam end or ends. |
| label | 1 | text |  |
| name | 0, 1 | BeamReleaseName |  |
| description | 1 | text, executable descriptor |  |
| MassProperty | id | 1 | integer | Defines [mass property](D:\\TMP\\.._prop.dita) for your model. |
| label | 1 | text |  |
| name | 0, 1 | MassPropertyName |  |
| description | 1 | text, executable descriptor |  |
| SpringProperty | id | 1 | integer | Defines [properties](D:\\TMP\\.._spring_stiff.dita) for spring elements. |
| label | 1 | text |  |
| name | 0, 1 | SpringPropertyName |  |
| description | 1 | text, executable descriptor |  |
| Material | id | 1 | integer | Assigns [material](D:\\TMP\\.._top.dita) to your model.  Considerations:  ● This task enables a user to create or change the material for a model by specifying type and name.  ● If you define this task without specifying the type attribute, the task is complete when a user assigns the material to the model. |
| label | 1 | text |  |
| type | 0,1 | Isotropic, TransIsotropic, Orthotropic | ● If you specify a material type, Process Guide opens the corresponding **Material Definition** dialog box.  ● If you do not specify this attribute, Process Guide invokes the [Materials](D:\\TMP\\.._df.dita) dialog box. |
| name | 0,1 | MaterialName | ● If the type is specified and a user creates a new material by specifying a name, Mechanica creates a new material with specified name.  ● If the type is specified and a user creates a material without specifying a name, Mechanica creates a new material with the default name.  ● If a user specifies the name of an existing material, Process Guide allows you to edit the material ignoring the type of material specified. |
| description | 1 | text, executable descriptor |  |
| MaterialOrientation | id | 1 | integer | Defines surface or Part type [material orientation](D:\\TMP\\.._orient_top.dita). |
| label | 1 | text |  |
| type | 0, 1 | Surface, Part | ● If you define this task without specifying the type attribute, the task is complete when at least the material orientation is defined.  ● If you specify a type, Process Guide invokes the dialog box to create a new or to edit the existing type of material orientation.  ● If you do not specify the type attribute, Process Guide invokes the [Material Orientations](D:\\TMP\\.._orient_df.dita) dialog box. |
| name | 0, 1 | MaterialOrientationName | ● If the type is specified and a user creates a new material orientation by specifying a name, Mechanica creates a new material orientation with the specified name.  ● If the type is specified and a user creates a material orientation without specifying a name, Mechanica creates a new material orientation with the default name.  ● If a user specifies the name of an existing material orientation, Process Guide allows you to edit the material orientation ignoring the type of material orientation specified. |
| description | 1 | text, executable descriptor |  |

**Meshes**

You can define tasks to create or control meshing of your model using the following objects in the process template:

● [AutoGEM](D:\\TMP\\pg_design.dita)

● [AutoGEMControl](D:\\TMP\\pg_design.dita)

● [AutoGEMSettings](D:\\TMP\\pg_design.dita)

● [Mesh](D:\\TMP\\pg_design.dita)

● [MeshControl](D:\\TMP\\pg_design.dita)

Consider the following points when you specify the AutoGEMControl and MeshControl object tasks:

● Process Guide associates the created or edited control with the task.

● The task is complete when creation or editing of control is complete.

● When a user deletes the control associated with the task, Process Guide marks the task as not valid.

Consider the following points when you specify the name attribute for the AutoGEMControl and MeshControl object tasks:

● If a user specifies a new name for the control, Mechanica creates a new control with the specified name.

● If a user creates a control without specifying a name, Mechanica creates a new control with the default name.

● If a user specifies the name of an existing control, Process Guide edits the existing control with the specified name.

The following table lists the object and its attributes for defining tasks to create model meshes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | Attributes | # of values | Valid Values | Description |
| AutoGEM | id | info | integer | Meshes your model through [AutoGEM](D:\\TMP\\.._mesh_autogem_top.dita) in Native mode. This object is available only in Native mode.  Considerations:  ● The task is complete when a user creates a p-mesh and saves the mesh file.  ● Process Guide marks the task as incomplete if the p-mesh file is missing. |
| label | 1 | text |  |
| description | 1 | text, executable descriptor |  |
| AutoGEMControl | id | 1 | integer | Specifies mesh distribution in your model through [AutoGEm control](D:\\TMP\\.._mesh_control_top.dita) in Native mode. |
| label | 1 | text |  |
| name | 0, 1 | AutoGEMControlName |  |
| type | 0,1 | MaxElemSize, EdgeDistribution, MinEdgeLength, EdgeLenByCurv, HardPoint, HardCurve, IsolateForExclusion, MappedMesh, PrismaticElem, ThinSolid | This attribute is optional. When this attribute is not specified, MaxElemSize is considered (as default). |
| description | 1 | text, executable descriptor |  |
| AutoGEMSettings | id | 1 | integer | Control various characteristics of element creation during mesh generation by specifying [AutoGEM settings](D:\\TMP\\.._mesh_dform.dita) in Native mode.  Considerations:  ● The task is complete when Process Guide invokes the dialog box and user closes the dialog box after making the changes. |
| label | 1 | text |  |
| description | 1 | text, executable descriptor |  |
| Mesh | id | 1 | integer | Enables you to [mesh](D:\\TMP\\.._mesh_mesh_top.dita) your model in FEM mode.  Considerations:  ● The task is complete when a user creates a H-mesh.  ● Process Guide marks the task as not valid when the H-mesh becomes invalid because of a change in the model. |
| label | 1 | text |  |
| description | 1 | text, executable descriptor |  |
| MeshControl | id | 1 | integer | Specifies mesh distribution in your model through [Mesh control](D:\\TMP\\.._mesh_mesh_control.dita) in FEM mode. |
|  | label | 1 | text |  |
|  | name | 0,1 | MeshControlName |  |
|  | type | 0,1 | MaxElemSize, MinElemSize, EdgeDistribution, HardPoint, HardCurve, DispCoordSys, MappedMesh, ShellElemDirection, MeshNumbering, MeshIDOffset, IgnoredMeshControl | This attribute is optional. When this attribute is not specified, MaxElemSize is considered (as default). |
|  | description | 1 | text, executable descriptor |  |

**Analyses and Results**

You can define tasks associated with the analysis and results phases using the following objects in the process template:

● [Analysis](D:\\TMP\\pg_design.dita)

● [Run](D:\\TMP\\pg_design.dita)

● [RunSettings](D:\\TMP\\pg_design.dita)

● [ResultTemplate](D:\\TMP\\pg_design.dita)

The following table lists the object and its attributes for defining tasks to analyze the model to evaluate the analysis results:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Object | Attributes/ Reference Object | # of values | Valid Values | Description |
| Analysis | id | 1 | integer | Defines [analyses](D:\\TMP\\.._top.dita) for the model for various structural aspects.  Considerations:  ● The created or edited analyses are associated with the task.  ● The task is complete when creation or editing of analysis is complete. |
| label | 0|1 | text | Specify this attribute, if the value of the visibility attribute is set to ON. |
| type | 0, 1 | Static, Modal, Buckling, Fatigue, PrestressStatic, PrestressModal, DynamicTime, DynamicFrequenecy, DynamicShock, DynamicRandom | This attribute is mandatory when creating a new analysis. |
| name | 0,1 | AnalysisName | ● If a user specifies a new name for analysis, Mechanica creates a new analysis with the specified name.  ● If a user creates an analysis without specifying a name, Process Guide creates a new analysis with the default name.  ● If a user specifies the name of an existing analysis, Process Guide allows you to edit the analysis ignoring the type of analysis already specified. |
| reference object="LoadSet" | 0+ | LoadSet id | ● This attribute is optional for creating a new Structural analysis.  ● Refers to the corresponding LoadSet through its task ID.  ● For example, <reference Object="LoadSet"> <id>2</id> </reference> |
| reference object="ConstraintSet" | 1+ | ConstraintSet id | ● This attribute is mandatory for creating a new Structural analysis.  ● Refers to the corresponding ConstraintSet through the its task ID.  ● For example, <reference Object="ConstraintSet "> <id>3</id> </reference> |
| description |  | text, executable descriptor | Specify this attribute if the value of the visibility attribute is set to ON. |
| Run | id | 1 | integer | Specify to [run](D:\\TMP\\.._analstuds.dita) the specified type of analysis. |
| label | 0|1 | text | Specify this attribute if the value of the visibility attribute is set to ON. |
| analysis | 1 | TaskID |  |
| description | 0|1 | text, executable descriptor | Specify this attribute if the value of the visibility attribute is set to ON. |
| RunSettings | id | 1 | integer | Specify various [settings](D:\\TMP\\.._setting_up.dita) for analysis run.  Considerations:  ● The task is complete when the [Run Settings](D:\\TMP\\.._setting_up.dita) dialog box is displayed once. |
| label | 1 | text |  |
| description | 1 | text, executable descriptor |  |
| ResultTemplate | id | 1 | integer | Displays analysis results using the specified template. If the visibility attribute is set to Off, the task automatically shows the result window using the specified result template when the analysis results are available. |
| label | 0|1 | text | Specify this attribute if the value of the visibility attribute is set to ON. |
| visibility | 0,1 | On, Off |  |
| template | 0 | 1 | path with file name |  |
| analysis | 1 | TaskID |  |
| mode | 1 | Repeat, Combine |  |
| description | 0|1 | text, executable descriptor | Specify this attribute if the value of the visibility attribute is set to ON. |

Return to [Process Template](D:\\TMP\\.._proc_template.dita).