

UGS CONNECTION



AMERICAS 2008



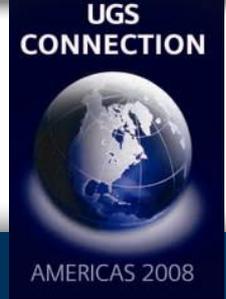
Siemens PLM Software

SIEMENS

A Journey From Model Organizational Requirements to Custom Checker Development

Tom Donovan
Processes & Tools
Boeing, St. Louis

A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Our Environment
 - ▶ Multiple programs serving different customers.
 - ▶ Some variation in model organizational requirements among programs.
 - ▶ Model validation authority is at the program level.
 - ▶ Profile authors are program engineering leads who are not familiar with detailed Check-Mate functionality and KF concepts.

A Journey From Model Organizational Requirements to Custom Checker Development



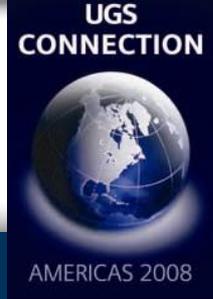
- ▶ Our Check-Mate Deployment Requirements
 - ▶ Impose minimal application knowledge overhead on the profile authors.
 - ▶ No configuring of checkers.
 - ▶ No manual editing of dfa files.
 - ▶ Focus on relatively few, very capable, highly customizable checkers to be heavily reused.
- ▶ Status
 - ▶ The custom checkers presented here are in production use and are modified and enhanced as needed.

A Journey From Model Organizational Requirements to Custom Checker Development

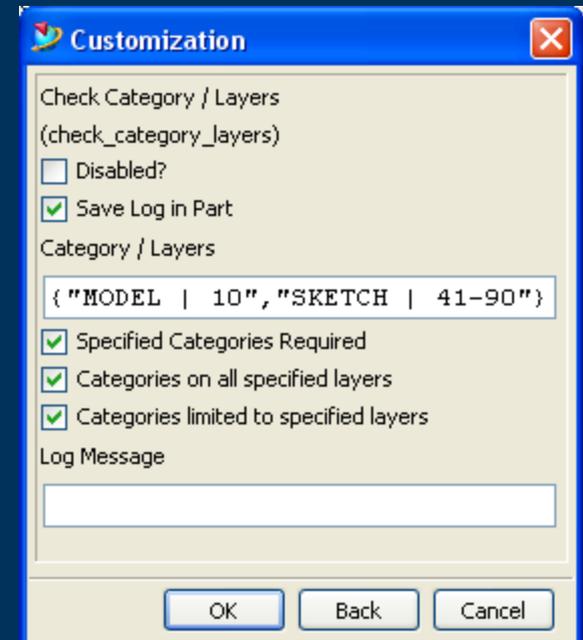


- ▶ Model Organizational Requirements
 - ▶ Some Variables:
 - ▶ Category, Layer, Entity Type, Entity Quantity, Entity Name, Reference Set
 - ▶ We translated our requirements into use cases for checking the relationships between these variables.
 - ▶ We found that Category is kind of a primary key because it is common throughout many model organizational requirements.
 - ▶ We developed several custom checkers, some of which are described here.

A Journey From Model Organizational Requirements to Custom Checker Development



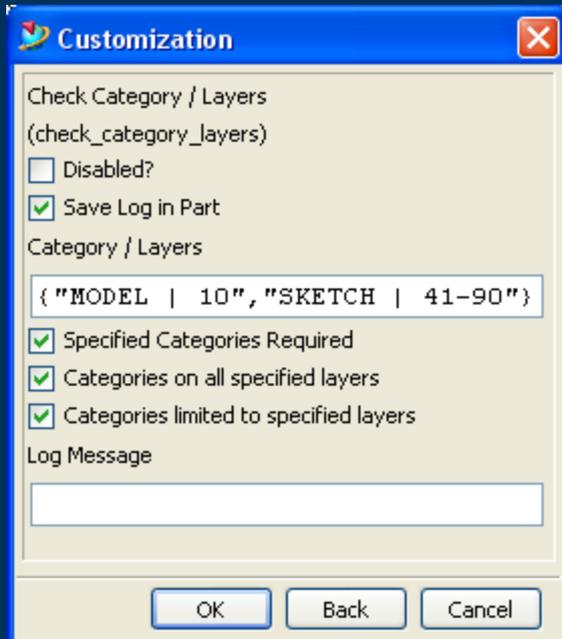
- ▶ Check Category / Layers
 - ▶ Evolved from the need to combine various category / layer requirements into one checker.
 - ▶ Direct entry of category / layer pairs.
 - ▶ First checkbox controls whether or not the categories are required.
 - ▶ Two other checkboxes allow more specific requirements when the categories are present.



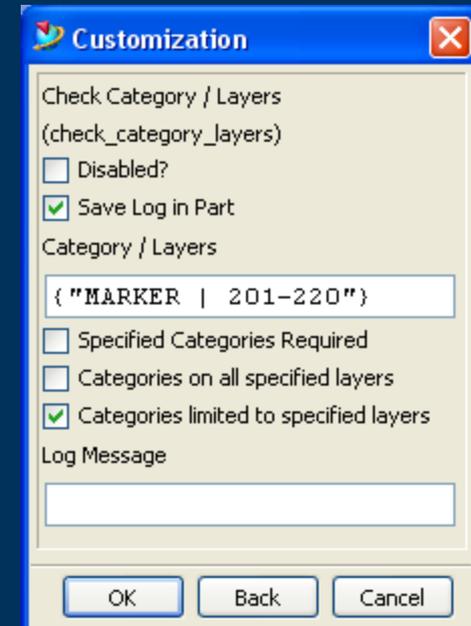
A Journey From Model Organizational Requirements to Custom Checker Development



Check Category / Layer examples:

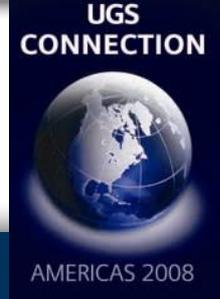


This customization is most typical and passes when the specified categories are on all their respective specified layers and not on any others.

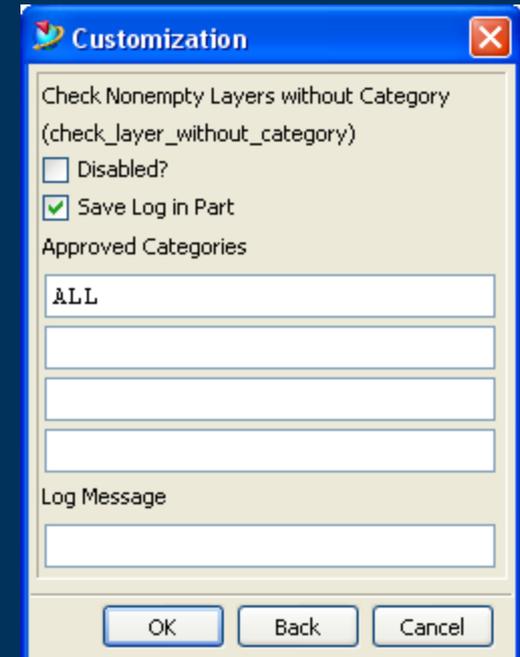


This customization passes when category MARKER is not present. It also passes when MARKER is present on at least one layer in the range 201-220 but not on any layers other than 201-220.

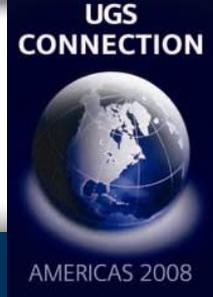
A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Nonempty Layers without Category
 - ▶ Evolved from the need to check for particular approved categories on nonempty layers.
 - ▶ Optionally, keyword ALL can be used and means that all categories are approved.



A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Category / Entity / Name
 - ▶ 3 stage checker that evolved from the need to check for the category and entity type of named entities.
 - ▶ Serial logic flow through the three stages. The check terminates at the first failure.
 - ▶ Collection of checkboxes in each stage are primarily for specifying the required entity quantity.
 - ▶ Entity name is optional.

The screenshot shows a dialog box titled 'Customization' with a close button in the top right corner. The dialog box contains the following fields and options:

- Check Category / Entity / Name (check_category_entity_name)
- Category Name: MODEL
- Disabled?
- Zero entities required in Category or
- Save Log in Part
- One entity required in Category or
- More than one required in Category
- Entity Type: SOLID_BODY (dropdown menu)
- Zero entities of type required or
- One entity of type required or
- More than one of type required
- No other entity types allowed
- Entity Name: MODEL
- One entity required with name or
- More than one required with name
- All entities required
- Log Message: (empty text box)

At the bottom of the dialog box are three buttons: OK, Back, and Cancel.

A Journey From Model Organizational Requirements to Custom Checker Development



Check Category / Entity / Name examples:

This customization passes when category MODEL contains only one entity and that one entity is a solid_body entity type and that one entity has the name MODEL.

A screenshot of a software dialog box titled 'Customization'. The dialog box has a blue title bar with a close button (X) in the top right corner. The main content area is light yellow and contains the following fields and options:

- Check Category / Entity / Name (check_category_entity_name)
- Category Name: A text box containing 'MODEL'.
- Disabled?:
- Zero entities required in Category or:
- Save Log in Part:
- One entity required in Category or:
- More than one required in Category:
- Entity Type: A dropdown menu showing 'SOLID_BODY'.
- Zero entities of type required or:
- One entity of type required or:
- More than one of type required:
- No other entity types allowed:
- Entity Name: A text box containing 'MODEL'.
- One entity required with name or:
- More than one required with name:
- All entities required:
- Log Message: An empty text box.

At the bottom of the dialog box, there are three buttons: 'OK', 'Back', and 'Cancel'.

A Journey From Model Organizational Requirements to Custom Checker Development



Check Category / Entity / Name examples:

This customization passes when category LOFT contains zero, one or more than one entity and zero, one, or more than one of those entities is a sheet_body entity type and the category has no other entity types.

In other words, it passes when there are only sheet_body entities in category LOFT.

A screenshot of a "Customization" dialog box. The title bar says "Customization" with a close button. The main area contains the following fields and options:

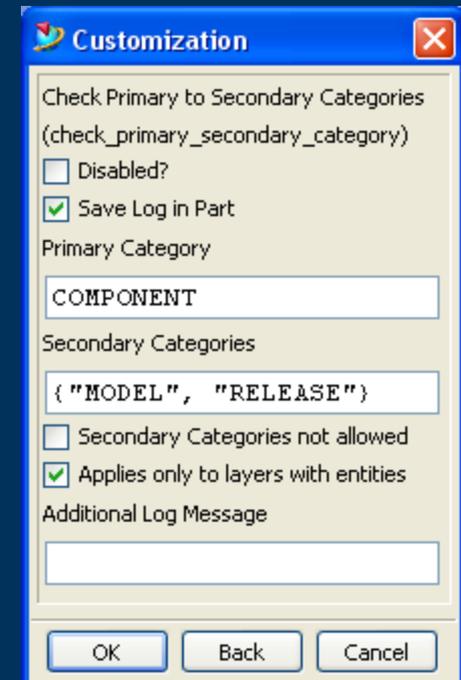
- Check Category / Entity / Name (check_category_entity_name)
- Category Name: LOFT
- Zero entities required in Category or
- Disabled?
- Save Log in Part
- One entity required in Category or
- More than one required in Category
- Entity Type: SHEET_BODY (dropdown menu)
- Zero entities of type required or
- One entity of type required or
- More than one of type required
- No other entity types allowed
- Entity Name: (empty text box)
- One entity required with name or
- More than one required with name
- All entities required
- Log Message: (empty text box)

At the bottom are three buttons: OK, Back, and Cancel.

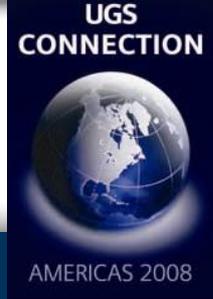
A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Primary to Secondary Categories
 - ▶ Evolved from the need to check that a particular category is always accompanied by other particular categories.
 - ▶ Optionally checks only layers with entities.
 - ▶ Optionally checks for the absence of secondary categories.

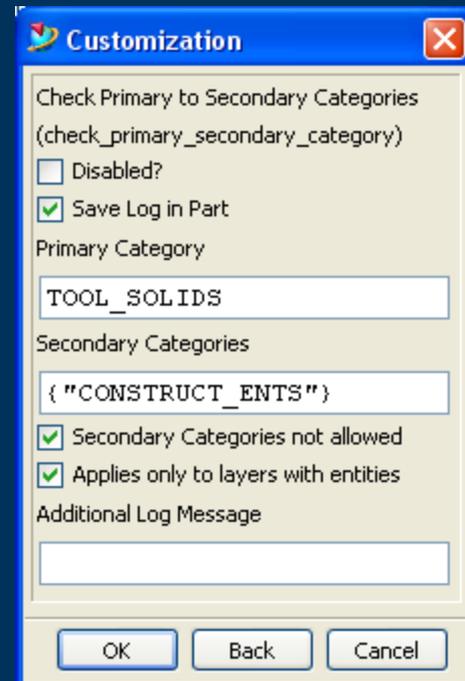


A Journey From Model Organizational Requirements to Custom Checker Development

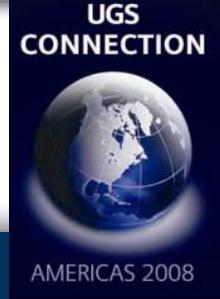


Check Primary to Secondary Categories example:

This customization passes when category CONSTRUCT_ENTS does not accompany category TOOL_SOLIDS on layers with entities.



A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Reference Sets with Categories
 - ▶ Evolved from the need to check that a reference set contains all the entities in a category with the exception of entities in another category.
 - ▶ The Exception Categories are optional.

A screenshot of a software dialog box titled "Customization". The dialog has a blue title bar with a close button (X) in the top right corner. The main area is light beige and contains the following fields and controls:

- Text: "Check Reference Set with Categories (check_refset_with_categories)"
- Checkbox: "Disabled?" (unchecked)
- Checkbox: "Save Log in Part" (checked)
- Text field: "Reference Set Name" with the value "RELEASE"
- Text field: "Primary Category Name" with the value "RELEASE"
- Text field: "Exception Categories" with the value {"LOFT"}
- Text field: "Additional log message" (empty)

At the bottom, there are three buttons: "OK", "Back", and "Cancel".

A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Entity / Layer / Category
 - ▶ 3 stage checker that evolved from the need to check for particular entity types on a particular layer range, like sketches or components.
 - ▶ Optionally checks for a particular category on the layers or keyword All can be used.
 - ▶ Optionally checks for the entity name as a category on the layer with the named entity.

A screenshot of a software dialog box titled "Customization". The dialog is for configuring a checker named "Check Entity / Layer / Category" (check_entity_layer_category). It contains several sections with checkboxes and input fields:

- Check Entity / Layer / Category (check_entity_layer_category)**
 - Disabled?
 - Save Log in Part
- Entity Type**: A dropdown menu set to "COMPONENT".
- Entity Requirements**
 - Zero entities required or
 - One entity required or
 - More than one required
- Eligible Layers**: A text input field containing "12-249".
- Layer Restrictions**
 - Entities of specified type not allowed on other layers
 - Nothing else allowed on layers with entities of specified type
 - Only one entity of specified type allowed per layer
- Category**: A text input field containing "COMPONENT".
- Category Options**
 - Category is exclusive to eligible layers with the specified entity type
 - Check for a category matching the entity name
- Log Message**: An empty text input field.

At the bottom right, there are three buttons: "OK", "Back", and "Cancel".

A Journey From Model Organizational Requirements to Custom Checker Development



Check Entity / Layer / Category examples:

This customization passes when more than one component entity are on layers 12 – 249 and not on any others and all the layers with the component entities have the category COMPONENT and all the layers without the component entities do not have the category COMPONENT.

The screenshot shows a "Customization" dialog box with the following settings:

- Check Entity / Layer / Category (check_entity_layer_category)
 - Disabled?
 - Save Log in Part
- Entity Type: COMPONENT (dropdown menu)
- Zero entities required or
- One entity required or
- More than one required

Eligible Layers: 12-249

- Entities of specified type not allowed on other layers
- Nothing else allowed on layers with entities of specified type
- Only one entity of specified type allowed per layer

Category: COMPONENT

- Category is exclusive to eligible layers with the specified entity type
- Check for a category matching the entity name

Log Message: (empty text box)

Buttons: OK, Back, Cancel

A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Master Model Density and Geometry
 - ▶ Evolved from the need to perform some of the examine geometry checks on a single master model solid.
 - ▶ Optionally checks for a particular density value.

Customization

Check Master Model Density and Geometry
(check_master_model_density_geometry)

Disabled?

Save Log in Part

Density (lb/cu inch): 0.1000

Check Density

Check Tiny objects

Check Misaligned Objects

Check Body Data Structures

Check Body Consistency

Check Face-Face Intersection

Check Sheet Boundaries

Check Face Smoothness

Check Face Self-Intersection

Check Face Spikes/Cuts

Check Edge Smoothness

Check Edge Tolerances

Tolerance Unit: pounds_and_inches

Distance Threshold Tolerance: 0.0250

Angle Threshold Tolerance in degree: 0.5000

Log Options: LOG_ERROR

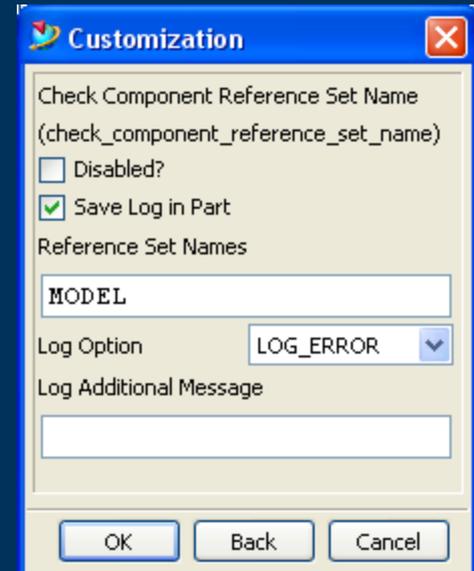
Log Additional Message

OK Back Cancel

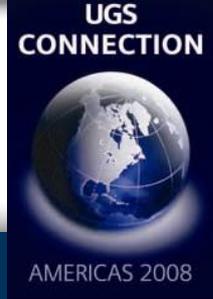
A Journey From Model Organizational Requirements to Custom Checker Development



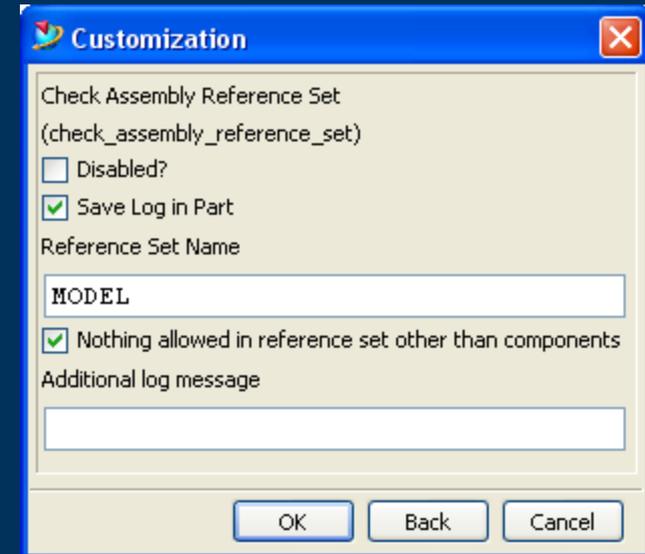
- ▶ Check Component Reference Set Name
 - ▶ Evolved from the need to check that all the components in an assembly are using a particular reference set.
 - ▶ Optionally allows for the specification of multiple reference sets to accommodate the requirement that the components be using any one of them.



A Journey From Model Organizational Requirements to Custom Checker Development



- ▶ Check Assembly Reference Set
 - ▶ Evolved from the need to check that all the components in an assembly are a member of a particular assembly level reference set.
 - ▶ Optionally checks that nothing is in the specified reference set other than components.



UGS CONNECTION



AMERICAS 2008



Siemens PLM Software

SIEMENS

Thank You
2008