

Creating a Custom Task Type for TCEng

Mark Hoover



Who is Hoover & Nebrig?

We are a small boutique Teamcenter Engineering consulting firm located in Seal Beach, California and have been providing consulting and programming services to the Unigraphics and Teamcenter communities since 1992.

Our goals are to educate, offer advice and alternatives, and facilitate the entire Teamcenter Engineering implementation process.

We can be found on the web at: www.HooverNebrig.com

Agenda

- Why would I need one?
- What makes up a Custom Task Type?
- A Custom Task in Action.
- Developing a Custom Task Type.
- How tough is this really?
- Tools Used.
- Summary.

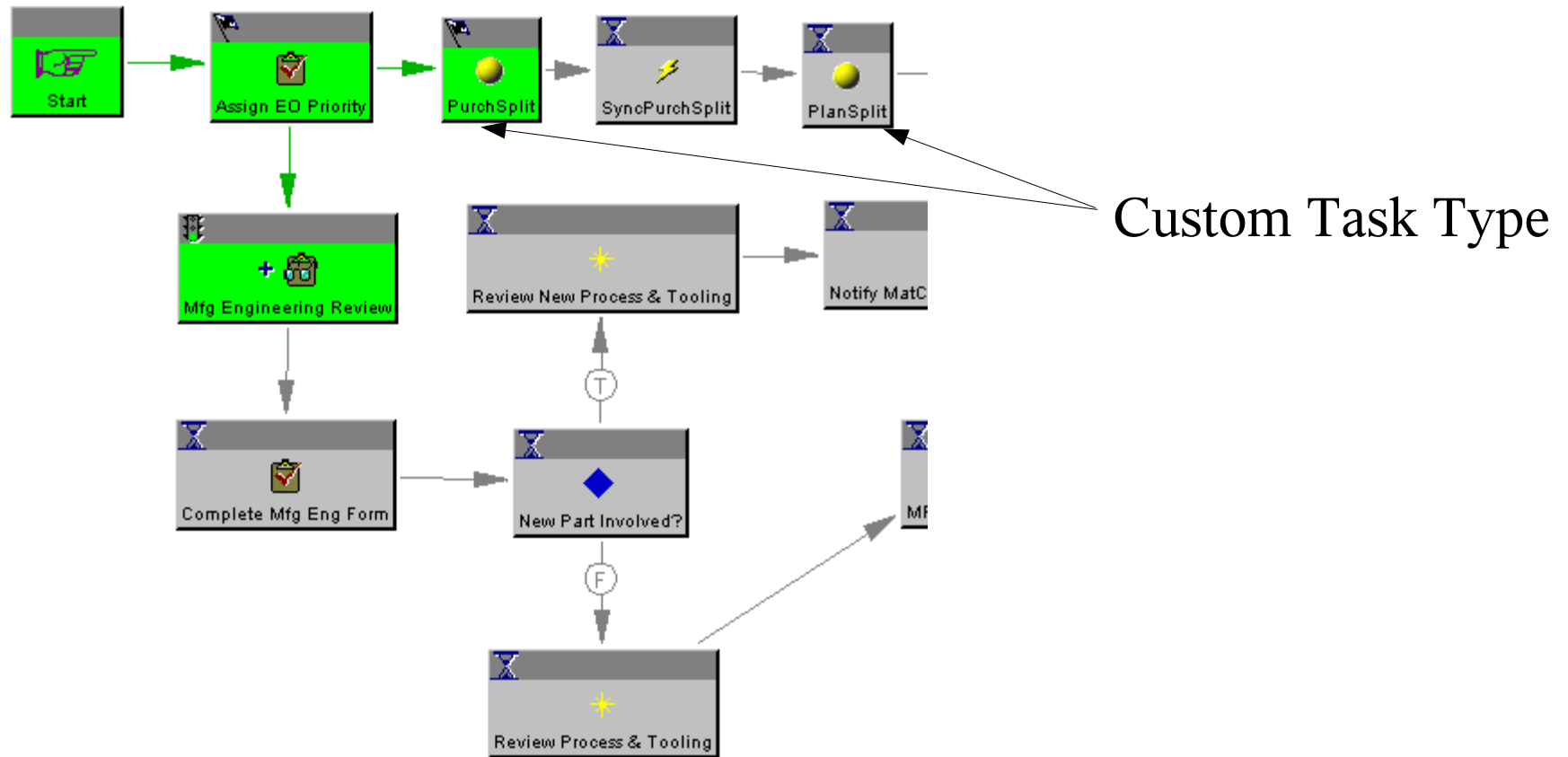
Special Acknowledgement

Thanks to the Workflow Development Team.
Without them answering my incessant
questions, you wouldn't be seeing this
presentation!

Why would I need one?

- To do something that is not provided in the standard Workflow task types
 - Add additional information to a perform UI
 - Perform additional processing on targets

A Process Designer Snippet

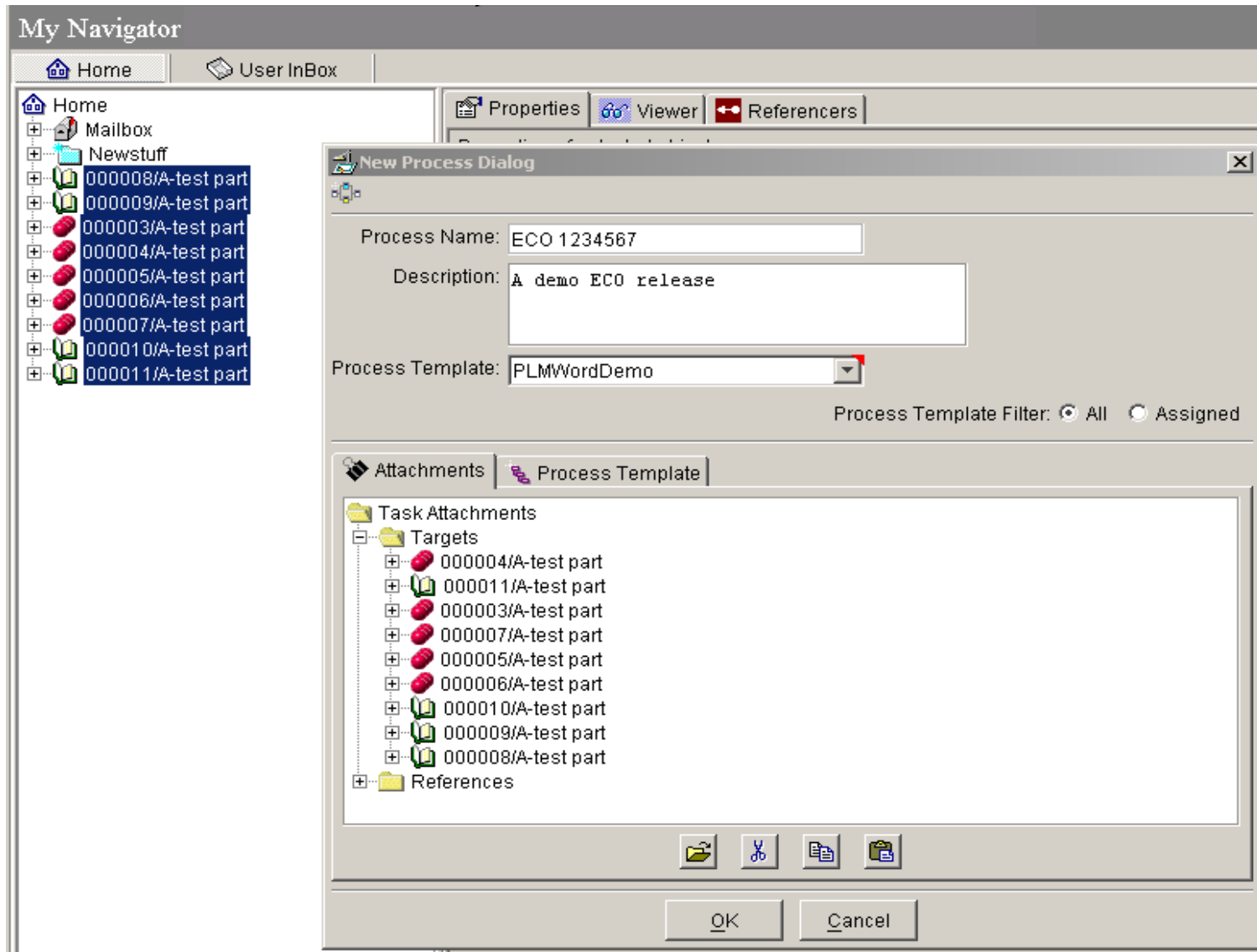


A Complex Custom Task in Action

The following 6 slides and avi file show a complex task type in action.

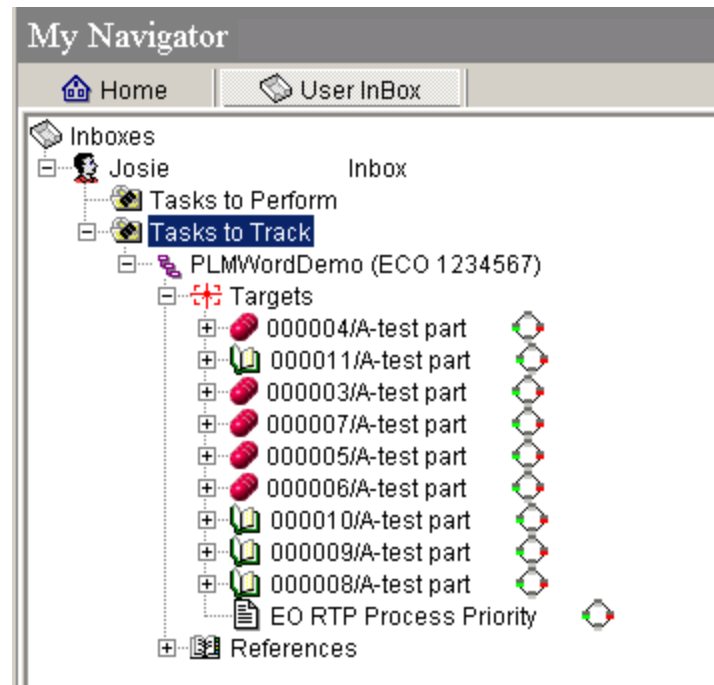
This complex task type presents the user with an interface which allows targets to be split from a parent job and sent off in zero or more child jobs.

User initiates a job with multiple targets



Initiator's Tasks to Track

Before the custom task



Note that all 9 targets are in a single Job.

Purchasing's Tasks to Perform

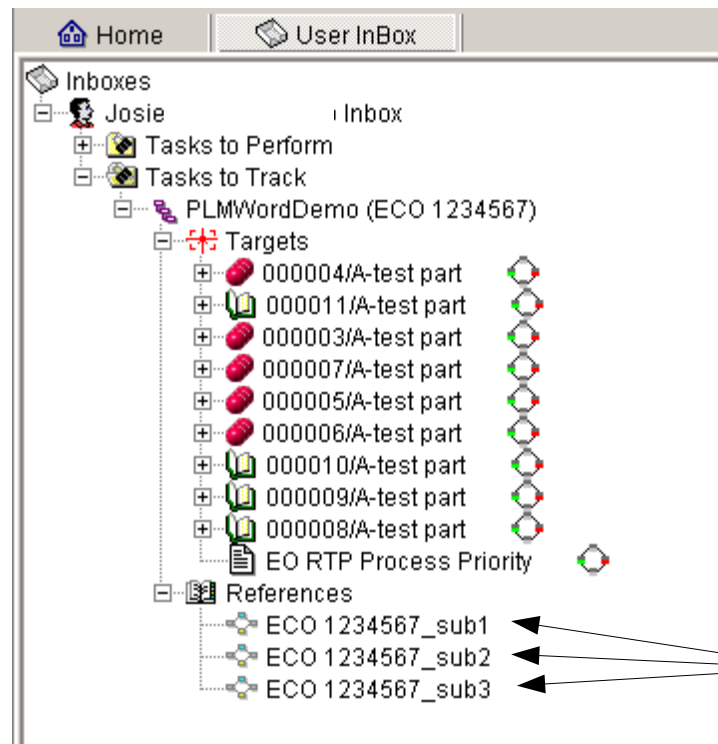
Splitting the Targets using the Custom Task

The screenshot shows a software interface with a left-hand navigation pane and a main content area. The navigation pane, titled "My Navigator", shows a tree structure under "Inboxes" for "Austin". The selected task is "PurchSplit (ECO 1234567)", which has a sub-section for "Targets". The main content area shows details for "EO #: ECO 1234567", initiated by "Josie" on "02-Mar-2004 08:44". A large white box with the text "AVI file not available..." is overlaid on the main content area. Below this, a table lists targets and responsible parties.

Target	Responsible Party
000003/A	
000004/A	
000005/A	
000006/A	
000007/A	
000008/A	
000009/A	
000010/A	
000011/A	

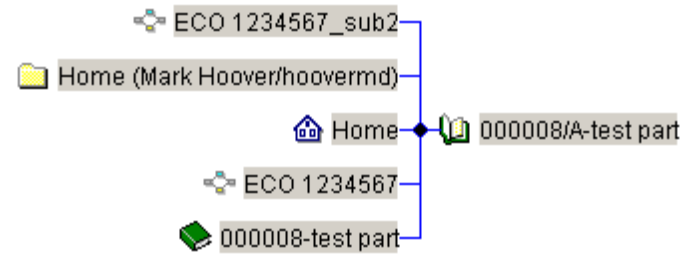
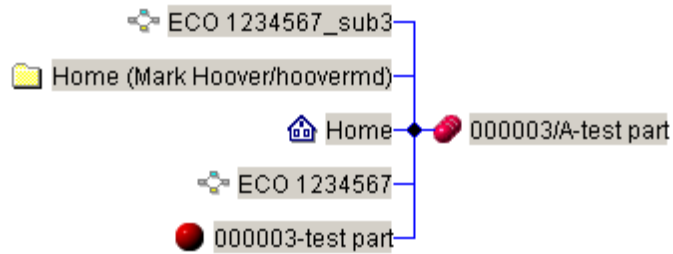
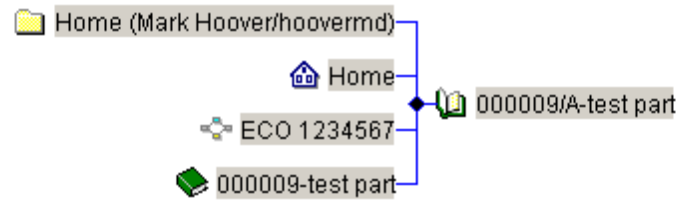
User's Tasks to Track

After the split

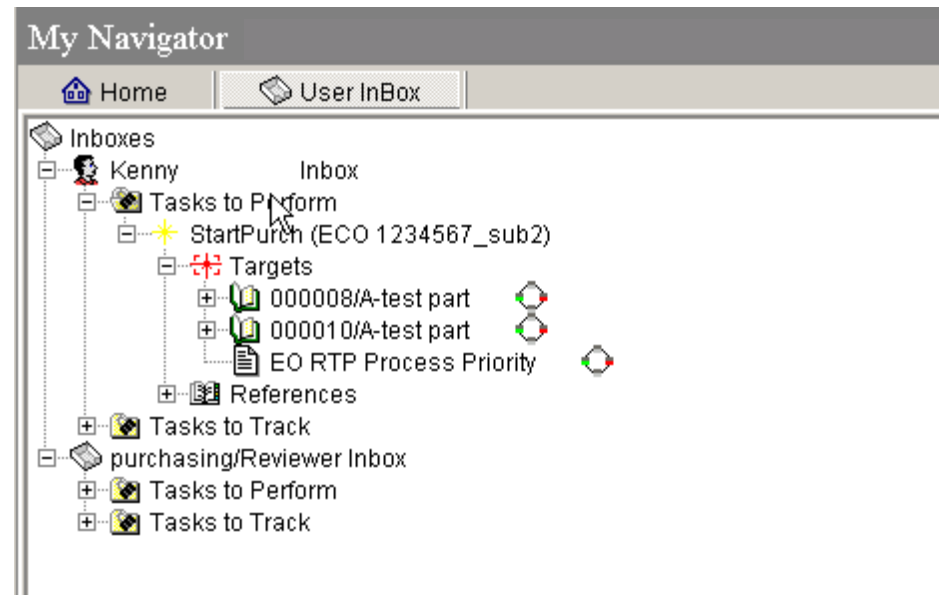


Note the three tasks as reference targets

Referencers



Another Purchasing Reviewer Tasks to Perform



What makes up a Custom Task Type?

- A new Iman Type (no schema mod required)
- An addition to the Task Definitions file
- A set of Icons
- JAVA code
 - Properties files
 - Code for the perform action

Sample Custom Task

Demonstrating creation of a custom task type in a simple fashion presents a challenge.

Keeping the sample code simple enough to be understood by someone unfamiliar with the concepts forces the sample to be trivial.

How to Create the Sample

- Install the new imantype
 - `install_types -f=add -t=plmworldTask -c=EPMTask`
- Add the new template types to the procedure definitions
 - `append_procedure -f=custom_task_template.txt`
- Copy the `uninstantiable_templates.dat` into your iman data directory.
- Build the Java Code

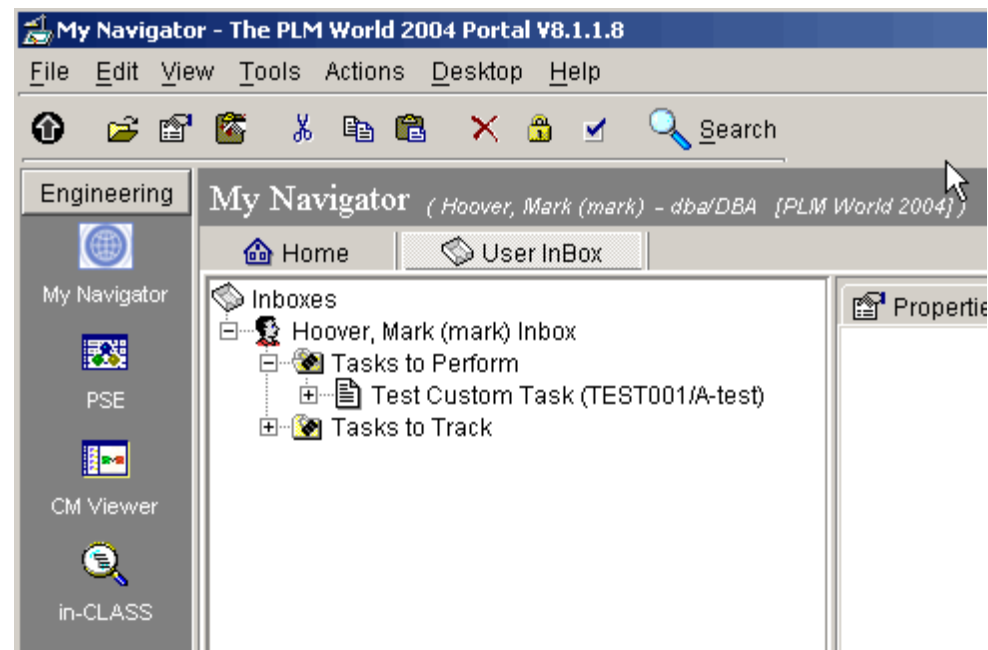
PLM World Sample Custom Task

The Task Type Icon

The screenshot displays the Process Designer interface. The title bar reads "Process Designer - The PLM World 2004 Portal V8.1.1.8". The menu bar includes "File", "Edit", "View", "Go", "Desktop", and "Help". The toolbar contains various icons for navigation and editing. The left sidebar shows a tree view with categories: Engineering, Manufacturing, Admin, Organization, Business Modeler, Type, List Of Values, and Query Builder. The main workspace shows a process template named "PLM World Test" with a task list containing "Test Custom Task" and "Hold Here". The workflow diagram at the bottom consists of four nodes: a green "Start" node, a grey "Test Custom Task" node, a grey "Hold Here" node, and a red "Finish" node, all connected by arrows. An arrow points from the text "The Task Type Icon" to a blue globe icon in the toolbar. Another arrow points from the text "The Task" to the "Test Custom Task" node in the workflow.

The Task

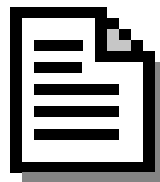
PLM World Sample Tasks to Perform



Required Java files

(com/ugsolutions/iman/common)

- `common_user.properties`
 - Define the icon `plmworldTask.ICON`
 - The icon used for display of the task in the tree on the Left Hand side of the process viewer.
 - If you do not define this icon here, the fallback will be to display the one defined in:
 - `workflow/common/common_user.properties`



Required Java files

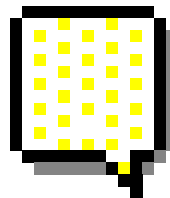
(com/ugsolutions/iman/common/imanviewer)

- imanviewer_user.properties
 - Add to TaskPerformViewer.TYPES
 - TaskPerformViewer.TYPES=plmworldTask,...
 - Add your custom UI for the task perform action
 - plmworldTask.VIEWPANEL=org.plmworld.iman.workflow.commands.task.CustomTaskPanel

Required Java files

(com/ugsolutions/workflow/common)

- common_user.properties
 - Define the icon plmworldTask.ICON
 - This icon is the one displayed within task box as seen in the process view.
 - If you do not define this icon here, the fallback will be to display the one defined in:
 - /iman/common/common_user.properties



Required Java files

(com/ugsolutions/workflow/common/actions)

- actions_user.properties
 - Define the task template icon
 - This icon is the one displayed in the Process Designer



Required Java files

(com/ugsolutions/workflow/common/actions)

- actions_user.properties
 - Define the task template action key names
 - This is the name of your task template
 - plmworldTaskTemplate
 - Add the template Action
 - **org.plmworld.iman.workflow.commands.task.CustomTaskCommand**
 - Add the Perform Command (`performCommand_plmworldTask`)
 - **org.plmworld.iman.workflow.commands.task.CustomTaskCommand**

Required Java files

(com/ugsolutions/workflow/common/actions)

- actions_user.properties
 - Add the template Icon, name and mnemonic
 - plmworldTaskTemplate.POPUP=true
 - plmworldTaskTemplate.NAME=PLM World Task
 - plmworldTaskTemplate.TIP=PLM World Custom Task Template
 - PlmworldTaskTemplate.MNEMONIC=P

Required Java files

(org/plmworld/iman/workflow/commands/task)

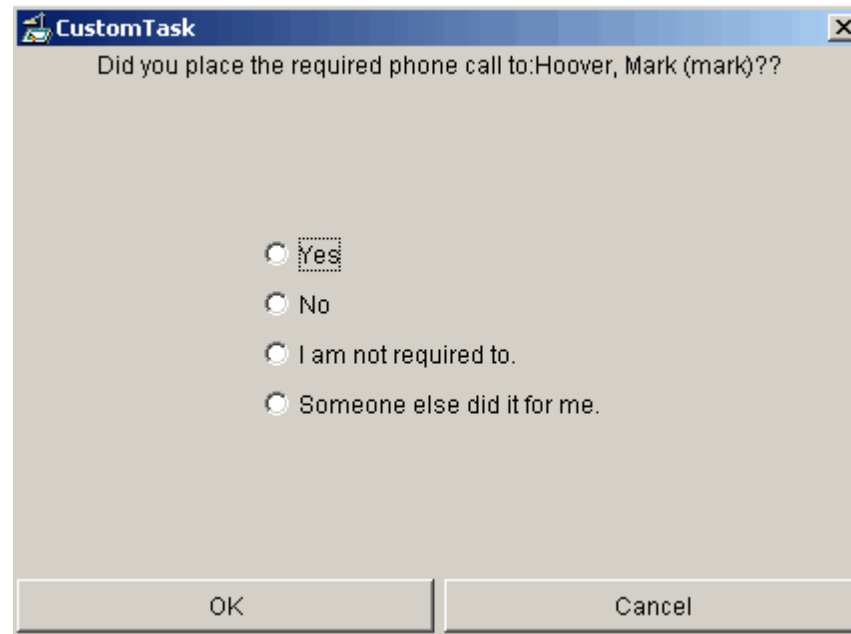
- The three components of the Perform UI
 - Command
 - The command is called by the action defined in the properties file.
 - Dialog
 - The dialog is displayed by the Command
 - Panel
 - The panel is contained within the Dialog

Required Java files

(com/ugsolutions/workflow/common/imanviewer)

- Imanviewer_user.properties
 - Add to TaskPerformViewer.TYPES
 - `TaskPerformViewer.TYPES=plmworldTask,...`
 - Add your custom UI for the task perform action
 - `plmworldTask.VIEWPANEL=org.plmworld.iman.workflow.commands.task.CustomTaskPanel`

Sample Tasks to Perform Custom UI



How tough is this really?

- Requires NO schema modification.
- Requires understanding of properties files.
- Requires understanding of basic JAVA.
- May or may not require User Service.

Tools Used

- JBuilder 7
- iMAN 8.1.1.9
 - JAVA 1.3.1-b24
- Teamcenter Engineering 9.0.0.2
 - JAVA 1.4.2_04-b05

Summary

- Don't be afraid to hack!
- Think about what you might do with your new knowledge.
- Complete Sample code is available at:

`www.HooverNebrig.com/samplecode.shtml`

Creating a Custom Task Type for TCEng

Questions?



Mark Hoover
mark@HooverNebrig.com