

Autodesk® PLM 360 Developer's Delight: A Hands-on PLM 360 Scripting Overview

Jared Sund

Autodesk PLM 360 Product Manager

PLM 360 Scripting Basics

Server-Side JavaScript

aka: Mocha, LiveScript, Jscript, ECMAScript

JavaScript 1.5 (Standard ECMA-262 3rd edition)

- NOT JAVA
- C construct language
- Not Strongly Typed

Server Side

- No Access to client side DOM (document object model)
 - navigator *such as: navigator.userAgent*
 - event *such as: event.button()*
 - window *such as: window.navigate()*
 - document *such as: document.getElementsByName().value*
- No client side functions *such as: alert(), document.write()*

JavaScript Basics

<http://w3schools.com/js/default.asp>



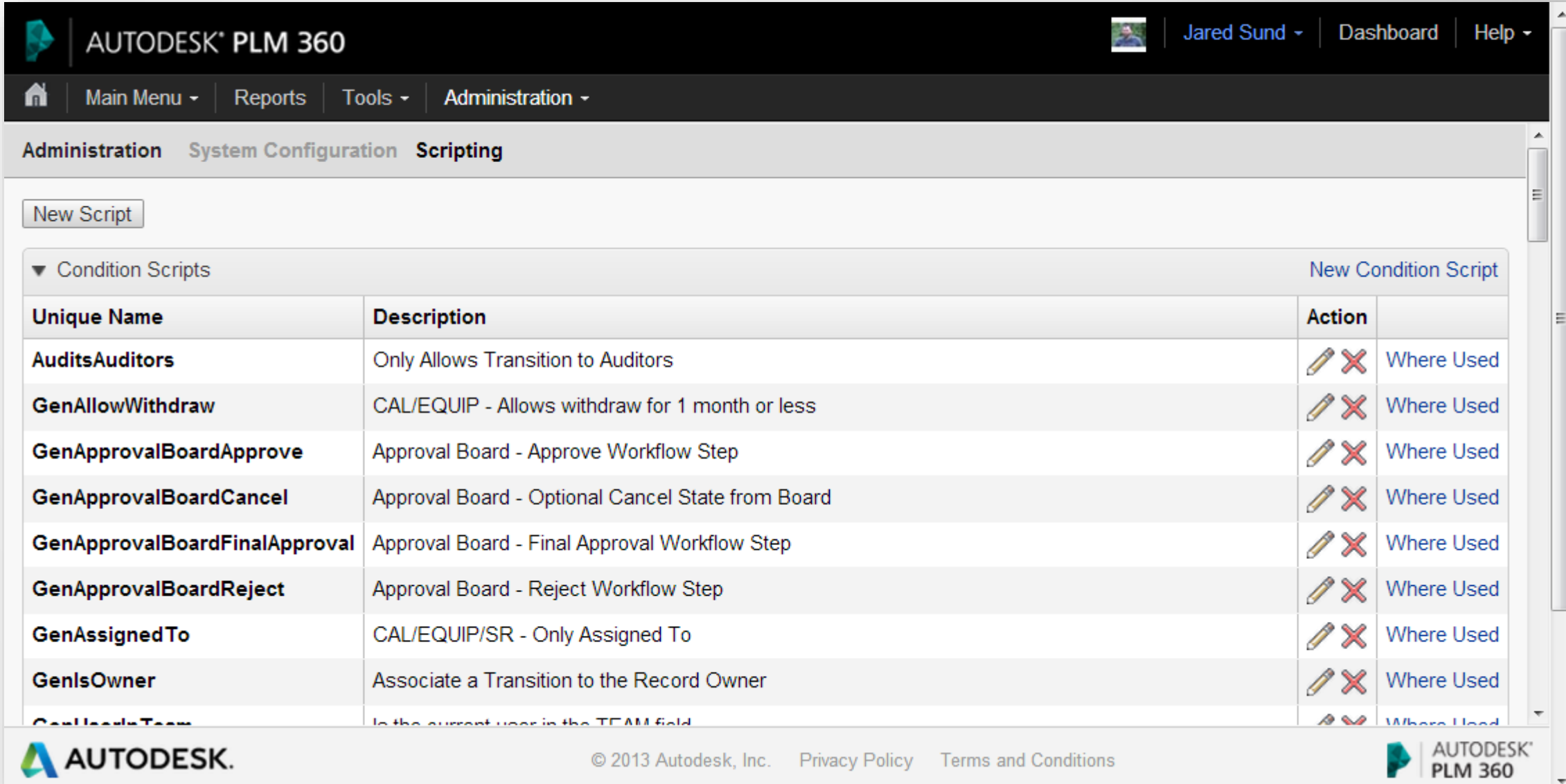
Comments	//Single line comment	Comments are a great way to document your scripts and are ignored by the scripting engine
	<pre>/* Multiple line comments Multiple line comments */</pre>	A multiline or block comment is great for creating a header at the top of your script. Everything inside <code>/* */</code> is ignored by the scripting engine
Variables	<pre>var instructor = 'Jared Sund';</pre>	The variable instructor contains the string – Jared Sund
	<pre>var qty = 7;</pre>	The variable qty holds the integer number 7
	<pre>var price = 3.25;</pre>	The variable price hold the decimal number 3.25
	<pre>var subTotal = qty * price;</pre>	The variable subTotal holds the decimal number 22.75 (7*3.25)
Array	<pre>var students = []; students[0] = 'Tom'; students[1] = 'Jane';</pre>	An array is a list of items
Conditions	<pre>If(subTotal > 150) { tax = .05; }</pre>	Conditions are questions. When the question is true (the value in the variable subTotal is greater than the number 150), perform the instructions with the body (<code>{ }</code>)
Loops	<pre>for(var i =0; i < students.lenght; i++){ println(students[i]); }</pre>	Loops are used to iterate through arrays (lists), or can be used to iterate until a condition (question) is true (see while loops)

PLM 360 Script Management

*Single location to manage
all your PLM 360 scripts*

Administration → System Configuration → Scripting

- ✓ Create New
- ✓ Edit Existing
- ✓ Delete Existing
- ✓ Where Used



The screenshot shows the Autodesk PLM 360 web interface. The top navigation bar includes the Autodesk logo, 'AUTODESK PLM 360', a user profile for 'Jared Sund', and links for 'Dashboard' and 'Help'. Below this is a main menu with 'Main Menu', 'Reports', 'Tools', and 'Administration'. The 'Administration' menu is expanded to show 'System Configuration' and 'Scripting'. The 'Scripting' page has a 'New Script' button and a table of 'Condition Scripts'. The table has columns for 'Unique Name', 'Description', 'Action', and 'Where Used'. Each row includes edit and delete icons. The footer contains the Autodesk logo, copyright information for 2013, and links for 'Privacy Policy' and 'Terms and Conditions'.

Unique Name	Description	Action	Where Used
AuditsAuditors	Only Allows Transition to Auditors		Where Used
GenAllowWithdraw	CAL/EQUIP - Allows withdraw for 1 month or less		Where Used
GenApprovalBoardApprove	Approval Board - Approve Workflow Step		Where Used
GenApprovalBoardCancel	Approval Board - Optional Cancel State from Board		Where Used
GenApprovalBoardFinalApproval	Approval Board - Final Approval Workflow Step		Where Used
GenApprovalBoardReject	Approval Board - Reject Workflow Step		Where Used
GenAssignedTo	CAL/EQUIP/SR - Only Assigned To		Where Used
GenIsOwner	Associate a Transition to the Record Owner		Where Used
GenIsInTeam	Is the current user in the TEAM field		Where Used

PLM 360 Script Editor

Script Management

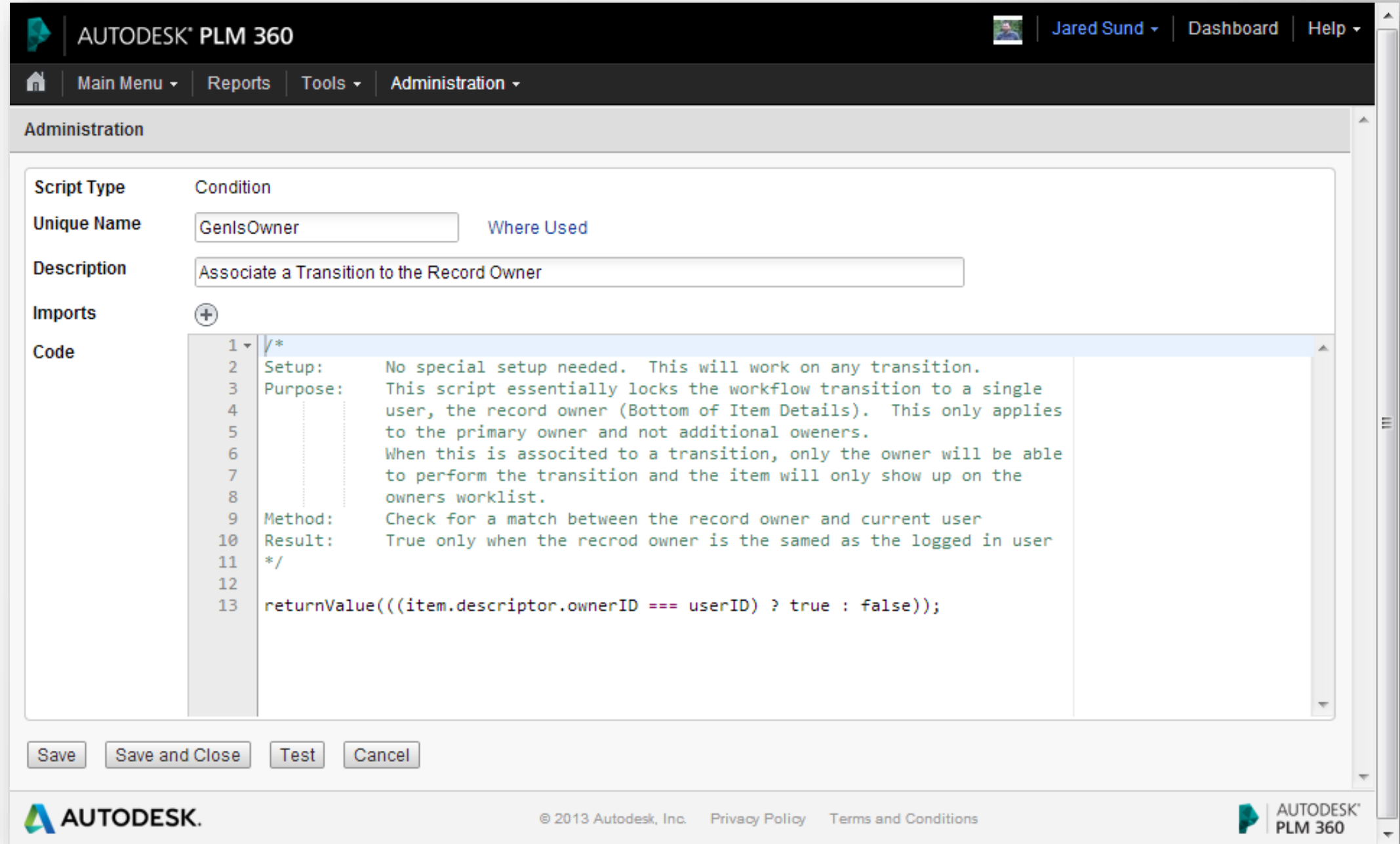
- Unique Name
- Description
- Imports

Code

- Embedded editor

Tools

- Save
- Close
- Testing
- Error Log



PLM 360 Script Debugging/Errors

Syntactic Errors (design time)

- In editor warnings
- On save messaging

Semantic Errors (run time)

- Testing
- Execution

Debugging

- Errors
- Script Log
- print() and println()

The screenshot displays the Autodesk PLM 360 Administration interface. At the top, the user 'Jared Sund' is logged in, with navigation links for 'Main Menu', 'Reports', 'Tools', and 'Administration'. A green banner at the top of the page reads 'Test of script GenIsOwner was successful'. The main content area shows the configuration for a script named 'GenIsOwner'. The 'Condition' is set to 'Associate a T...'. The 'Code' section contains the following script:

```
1 /*
2 Setu
3 Purp
4
5
6
7
8
9 Meth
10 Resu
11 */
12
13 println(item.descriptor.ownerID);
```

The 'Debug Output' field shows 'jared.sund@autodesk.com' and the 'Script Result' is 'false'. A 'Test script...' dialog box is open in the foreground, prompting the user to enter parameters for testing. The dialog contains the following fields:

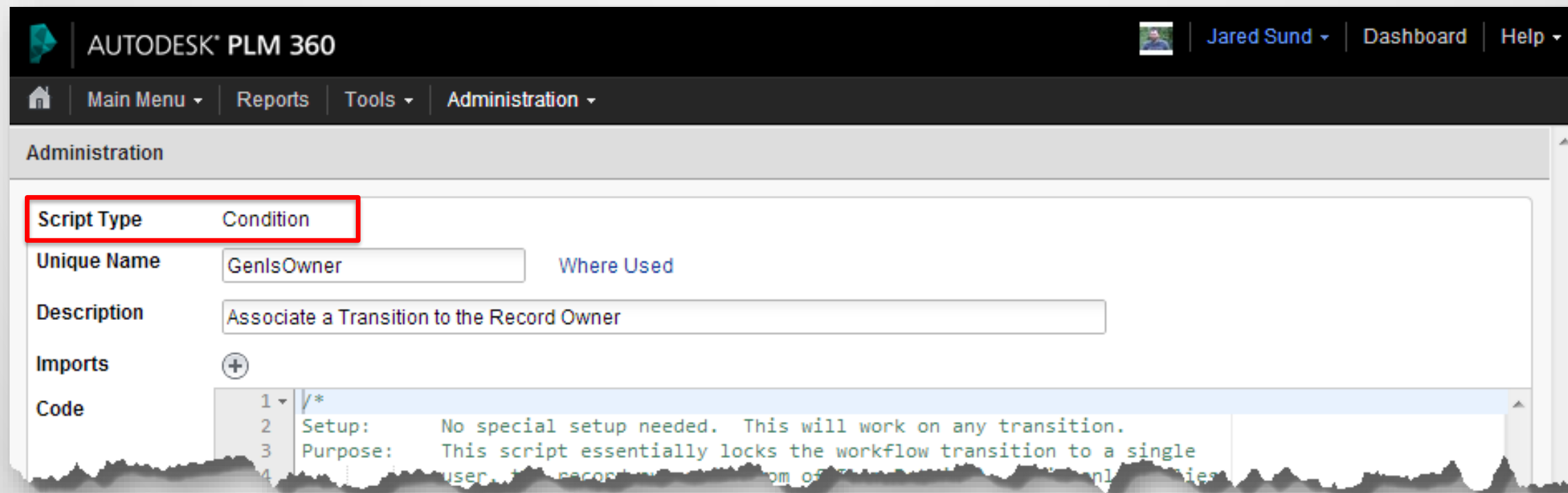
Parameter	Type	Value
dmsID	Integer	2886
userID	String	jared.sund
transID	Integer	
customTransID	String	

The dialog also includes 'Test' and 'Cancel' buttons. At the bottom of the interface, there are buttons for 'Save', 'Save and Close', 'Test', and 'Cancel'. The footer contains the Autodesk logo, copyright information for 2013, and links to 'Privacy Policy' and 'Terms and Conditions'.

PLM 360 Script Types and Events

PLM 360 Script Types

- Condition – returns true/false (boolean)
- Validation – returns a list (array) – empty or populated
- Action – does not return a value
- Library – holds one or more JavaScript functions or objects



PLM 360 Script Events

Scripts DO NOT maintain PLM 360 permissions

Workflow Transitions

- Precondition
 - Script Type: Condition (boolean)
 - Hide or show WF Actions based on the outcome of a condition script
- Validation
 - Script Type: Validation (array)
 - Validates requirements before allowing transition to the next state
- Action
 - Script Type: Action
 - Performs some action, sends an email

Workspace Behaviors (action scripts)

- Add item
 - Performs some action, sets the value of a default field
- Edit item details
 - Performs some action, calculates the total of values

Transition Properties

Name: [Text Field]

Description: [Text Field]

Hide in online map

Permission: [Dropdown Menu]

[Create new permission]

Precondition: [Dropdown Menu]

Validation: [Dropdown Menu]

When this transition is performed

Action: [Dropdown Menu]

Notify by e-mail on occurrence

When this transition becomes available

Display in Outstanding Work

Notify users who have permission to perform it

Password required

[Optional]

[Show IDs]

State Properties

General Escalation

Enabled:

After: [2 Day(s)]

Include Weekends

Type: [Run a script]

Script: [CO_Approval_Escalation]

Notify

New Run A Script Escalations!

PLM 360 Scripting Object and Functions

PLM 360 API item object

The item object is prepopulated from the workspace item associated with the script

Property	Description	Examples
descriptor	Properties for the item object (meta-fields)	<ul style="list-style-type: none">item.descriptor.workflowStateitem.descriptor.ownerID
fields	Read/write access to custom fields added to item details or the grid tab	<ul style="list-style-type: none">item.TITLEitem.QTY
grid	Read/write access to the rows and columns of a grid[row][column]	<ul style="list-style-type: none">item.grid[1].QTYitem.grid[2].TITLEitem.grid.addRow(row)item.grid[1].remove()
milestones	Read/write access to items in the milestones tab	<ul style="list-style-type: none">item.milestones[1].progressitem.milestones[1].milestoneDate
attachments	Read access to items in the attachments tab	<ul style="list-style-type: none">item.attachments[1].fileStatusitem.attachments[1].fileSize
workflowActions	Read only access to workflow actions history (first is the most recent)	<ul style="list-style-type: none">item.workflowActions[0].userIDitem.workflowActions[0].timeStamp
functions	Additional operations that can be performed on an item	<ul style="list-style-type: none">item.performWorkflowTransitionitem.addMilestoneitem.deleteItem

PLM 360 API functions

PLM 360 API functions outside of the item object

function	Description	Example
createItem	Creates a new record in a given workspace	<ul style="list-style-type: none">var newItem = createItem(workspace ID);
getPrintView	Returns the rendered html body of an Advanced Print View	<ul style="list-style-type: none">var body = getPrintView(APV name);
loadItem	Returns an existing item by dmsID	<ul style="list-style-type: none">var relatedItem = loadItem(dmsID);
security	A set of functions to return user/group/role information	<ul style="list-style-type: none">var user = Security.loadUser(userID);var inGroup = Security.inGroup(group name);
Email	Create and send emails from scripts	<ul style="list-style-type: none">var email = new Email();email.to = 'jared.sund@autodesk.com';email.subject = 'This is a test email';email.body = getPrintView('Item Details');email.send();
print/println	Used when testing scripts, writes the debug section	<ul style="list-style-type: none">println(item.descriptor.workflowState);
Logger	Writes to the item's Change Log	<ul style="list-style-type: none">Logger.log('Owned by: ' + item.descriptor.ownerID);
XMLHttpRequest	Access external web services	

Class Links

Lab Files: <http://tinyurl.com/PL2760>

PLM 360 Test Tenant Request:

Your Tenant:

<https://PL2760<lastname>.autodeskplm360.net>

For Questions, discussion on this class:

<http://forums.autodesk.com/t5/PLM-360-General/AU2013-PLM-Scripting-Lab-PL2760/td-p/4675979>