

*Tivoli Now!*

# WELCOME TO TIVOLI NOW!

IBM would like to acknowledge the support of our sponsors.



*Tivoli Now!*

# Extending IBM Tivoli Identity Manager 4.6 Using JavaScript



*Tivoli Now!*

# David Edwards

Consulting IT Specialist  
Tivoli Global Response Team AP  
Tivoli Security  
davidedw@au1.ibm.com



## Abstract

This session will cover how to customize IBM Tivoli Identity Manager (ITIM) 4.6 using JavaScript in Provisioning Policies and Workflows. Come learn scripting techniques and best practices for writing JavaScript in ITIM.

# Objectives

Identify where JavaScript can be used.

Describe frequently used functions.

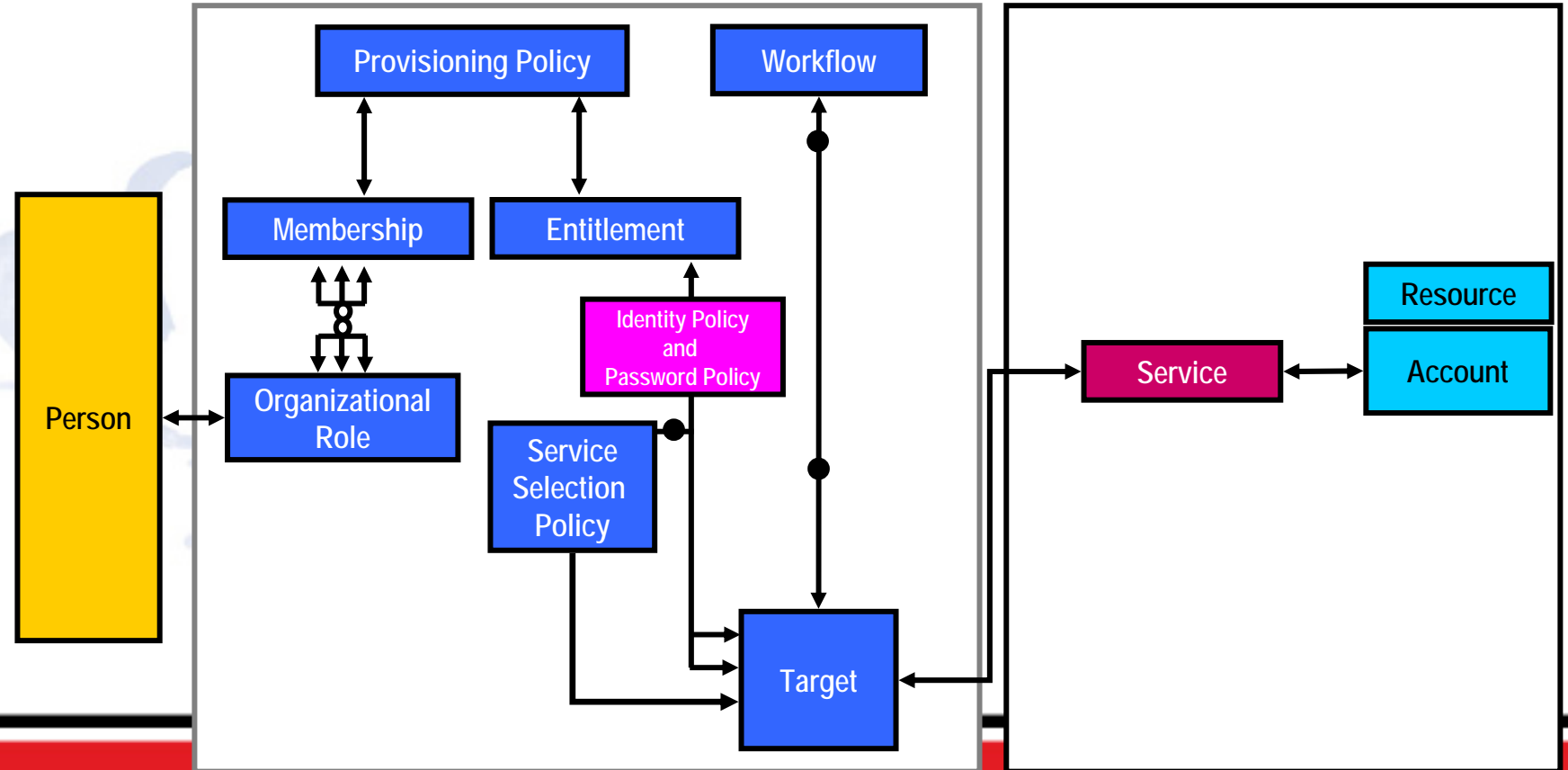
Use JavaScript functions in Provisioning Policy parameters.

Create advanced Workflows.

Troubleshoot common script errors.

# Account Provisioning Review

┌ People ─┤ ─────────── Authorization ───────────┤ ─────────── Resources ───────────┤



# JavaScript Writing Tips

Leverage existing functions.

Write extensive JavaScript in a text editor and copy into Provisioning Policy or Workflow.

Build and test:

- Use Save as Draft and Preview functions for Provisioning Policies.
- Add elements to Workflows one at a time.

Debug JavaScript using Enrole.log function:

- `Enrole.log("name of component", "message");`

Use the log viewer script.

# Frequently Used JS Functions

`subject.getProperty(String rowAttrName)`

```
subject.getProperty("employeenumber");
```

`service.getProperty(String rowAttrName)`

```
service.getProperty("erServiceName");
```

`PersonSearch.searchByFilter(String profileName, String filter, [int scope])`

```
PersonSearch.searchByFilter("Person", "(cn="Alice Smith")", 2);
```

`ServiceSearch.searchByFilter(String filter, [int scope])`

```
ServiceSearch.searchByFilter("(erServiceName=Trade*)", 1);
```

See Admin Guide

Javascript Extensions Chapter

Identity Policies Chapter

# Advanced Provisioning Parameter List

Allows advanced customization of the service parameters in the account form.

Provides advanced attribute enforcement types:

- Default
- Mandatory
- Optional
- Excluded

## Returning the Name of a Relationship Object

```
{  
    var org = subject.getProperty("parent")[0].name;  
    var title = subject.getProperty("title");  
    if ((title != null) && (title.length > 0)){  
        title = subject.getProperty("title")[0];  
    } else {  
        title = "";  
    }  
    ...  
}
```

## Verifying an Attribute Is Defined

```
...  
var sec = subject.getProperty("xyzsecclearance");  
if ((sec != null) && (sec.length > 0)) {  
    sec = subject.getProperty("xyzsecclearance")[0];  
} else {  
    sec = 0;  
}  
...
```

## Creating a New Array

...

```
var grp = new Array();
grp[grp.length] = "documentation";
if ((org == "Finance") || (title == "CEO")){
    grp[grp.length] = "accounting";
    if ((sec >= 2) || (title == "CEO")){
        grp[grp.length] = "budget";
    }
}
return grp;
}
```

# Instructor Demonstration



# Workflows

## Entitlement

- Specify the process to get approval for account creation.
- Can end with the request either accepted or rejected.
- Specified in the entitlements of the Provisioning Policy.

## Operation

- Modify the behavior of IBM Tivoli Identity Manager during an operation:
  - Add
  - Modify
  - Delete
  - Suspend
  - Restore

# Basic Workflow Elements

## Start

- Defines the beginning of a Workflow sequence

## Request for Information (RFI)

- Identifies information required to complete the Workflow

## Approval

- Defines signature authorities required to proceed with the Workflow

## Subprocess

- Acts as a pointer to a separately defined Workflow

## Script

- Allows a custom defined JavaScript to be executed by the Workflow

## Work Order

- Notifies a Workflow participant of an action that must be completed

## End

- Defines the end of a Workflow sequence

## Transition line

- Used to connect two elements and set status to approved, rejected, or custom

# Additional Workflow Elements

## Loop

- Repeats specified activities for a defined period of time or based on specific conditions to prevent a request from timing out.

## Operation

- Used to call a Lifecycle Operation Workflow.

## Extension

- Specifies a Workflow extension type to execute and input and output parameters.

# Extending Workflow Elements

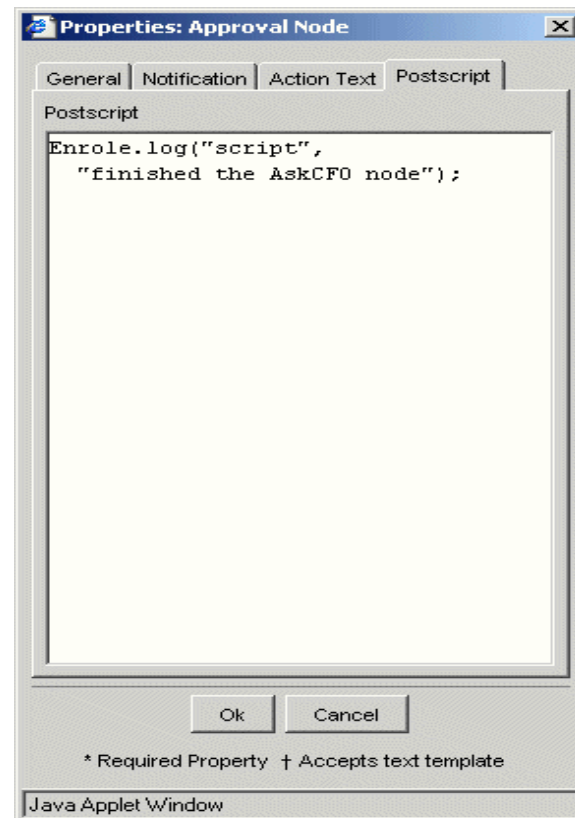
## Postscript

- Not script node

## Custom participant

## Messages

- Notification
- Action Text



# Custom Participants

Several Workflow elements have participants:

- Approval
- RFI
- Work Order

Custom participants are determined using a script.

Properties: Approval Node

General | Notification | Action Text | Postscript

\* Activity ID: TimeBasedApproval

Activity Name:

\* Participant: Custom

Escalation Participant: Participant Type

Escalation Limit: 2 Days 0 Hours 0 Minutes 0 Seconds

Join Type:  AND  OR Split Type:  AND  OR

Entity Type: Person

Input Parameters: Search Relevant Data

ID	Type	Relevant Data ID
entity	Person	
container	OrganizationalContainer	

Ok Cancel

\* Required Property + Accepts text template

Java Applet Window

# Time-based Custom Participant

```
var now = new Date();
var hour = now.getHours();
var day = now.getDay();
var approverName;

if ((day == 0) || (day == 6) || (hour < 8)
    || (hour > 16)) {
    approverName = "Alice Smith";
} else {
    approverName = "John Doe";
}

...
```

## Time-based Custom Participant (cont.)

...

```
Enrole.log("", "Day:" + day + " Hour:" + hour + "  
  Approver:" + approverName);  
var personSearch = new PersonSearch();  
var searchResults =  
  personSearch.searchByFilter("Person",  
  "(cn=" + approverName + ")", 2);  
var approverDN = searchResults[0].dn;  
  
return new Participant(ParticipantType.USER,  
  approverDN);
```

# Messages

Several Workflow elements have messages:

- Notifications that are sent using e-mail
- Action text that is shown in the to-do list

Messages can contain JavaScript, enclosed between `<JS>` and `</JS>` tags.

The JavaScript can be an expression, in which case the result will be placed in the message.

```
<JS>type =
    process.getParent().requestorType;

if (type == "S") return
    "The Workflow engine";
else if (type == "P")
    return "The system";
else return "A person
    called " +
        process.getParent(
    ).requestorName;
</JS> requested an
account for
<JS>process.requesteeName</JS>
```

## Required Escape Characters

Certain characters have to be escaped in the XHTML notification template:

- Less than (<)—&lt;
- Greater than (>)—&gt;
- Quotation mark (")—&quot;
- Apostrophe (')—&apos;
- Ampersand (&)—&amp;

# Workflow Properties

## Parameters of the Workflow:

- Input Parameters
- Output Parameters
- Relevant Data

Input Parameters

ID	Type	Relevant Data ID
entity	Account	entity
service	Service	service
owner	Person	owner

Output Parameters

ID	Type	Relevant Data ID
entity	Account	entity

Relevant Data

	ID	Type
S	entity	Account
	service	Service
R	owner	Person

S: Subject R: Requestee B: Both

Java Applet Window

# Data Types

Parameters support a variety of data types:

- IBM Tivoli Identity Manager data types
  - ITIM objects – account, service, person
- Standard data types
  - String etc.

# Using Parameters in WF Elements

Some Workflow elements have input parameters.

These input parameters can be mapped to a Relevant Data ID of the same type.

ID	Type	Relevant Data ID
entity	Account	Entity
service	Service	service
owner	Person	acctOwner

Ok Cancel

\* Required Property † Accepts text template

Java Applet Window

## Parameter-based Custom Participant

```
var ownerDN =  
    Entity.get().getProperty("owner")[0]  
    ;  
return new Participant  
    (ParticipantType.USER, ownerDN);
```

# Script Elements

Add more sophisticated processing

More readable than putting a long script in the postscript of other elements

## Setting an Account Attribute

```
var acct = Entity.get();  
var exp = acct.getProperty ("expirationdate")[0];  
var milliDate = Enrole.toMilliseconds(exp);  
var newmilliDate = milliDate + 30*24*3600*1000;  
var expireDate = Enrole.toGeneralizedTime(  
new Date(newmilliDate));  
acct.setProperty ("expirationdate",  
new Array(expireDate));  
Entity.set(acct);
```

# Where Else Can We Use JavaScript in ITIM?

## Adoption Rules (Services)

Define how accounts are adopted

## HR Feed

Placement Rule

## Service Selection Policies

Which service

## Dynamic Org Role

Membership definition

## ITDI – Infinite applications!

# Summary

You should now be able to:

- Identify where JavaScript can be used.
- Describe frequently used functions.
- Use JavaScript functions in Provisioning Policy parameters.
- Create advanced Workflows.
- Troubleshoot common script errors.



## Disclaimers and Trademarks

No part of this document may be reproduced or transmitted in any form without written permission from IBM Corporation.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements (e.g. IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided.

IBM customers are responsible for ensuring their own compliance with legal requirements. It is the customer's sole responsibility to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws.

The following terms are trademarks or registered trademarks of the IBM Corporation in either the United States, other countries or both: DB2, e-business logo, eServer, IBM, IBM eServer, IBM logo, Lotus, Tivoli, WebSphere, Rational, z/OS, zSeries, System z.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

ITIL® is a Registered Trade Mark, and a Registered Community Trade Mark of the Office of Government Commerce, and is Registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library® is a Registered Trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.