



**UGS**

*Transforming the  
process of innovation*



## Teamcenter Systems Engineering and Interoperability within Teamcenter

**Jacob Davies**  
**TcSE Implementation Engineer**  
**[jacob.davies@ugs.com](mailto:jacob.davies@ugs.com)**



- ▶ Introduction
- ▶ Understanding the Integration
  - ▶ What Linking Can and Cannot Do
- ▶ WOLF Linking
- ▶ Proxy Linking
  - ▶ Proxy Linking Explained
  - ▶ Understanding the Link Direction
- ▶ How to Set Up & Install the Integration
  - ▶ TcSE
  - ▶ TcEnterprise
  - ▶ TcEngineering
- ▶ Summary



# Process Interoperability



- ▶ Design engineers working on large projects must verify that each one of the thousands of requirements are being met
- ▶ Requirement engineers must ensure that designers have the latest, most up-to-date requirements available
- ▶ They may both be using Teamcenter products, but “integration” usually consist of importing, exporting, attaching, and printing documentation for verification along each stage of Product Lifecycle
  - ▶ **What if we could digitally link the Requirements process work products with the Concept & Product Engineering process work products? – building knowledge**
  - ▶ **What if we could simply click on a requirement in Teamcenter for Systems Engineering and allocate the requirement or function to a configuration item in Teamcenter Engineering/Enterprise?**



## TcEngineering or TcEnterprise

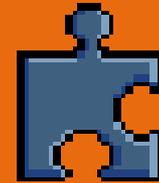


### **PART**

- ▶ Workflows
- ▶ Assemblies
- ▶ Models
- ▶ Revisions
- ▶ Properties

*Object-Oriented*  
**Oracle Database**

## TcSystems Engineering



### **Requirement**

- ▶ Related Objects
- ▶ Attachments
- ▶ Content
- ▶ Versions
- ▶ Properties

*Relational*  
**Versant Database**

How do we link objects with  
fundamentally different  
architecture in completely  
different databases?



# Understanding the Integration



- ▶ You CANNOT link...
  - ▶ Processes unless they are modeled in TcSE
  - ▶ Toolsets
  - ▶ Tool Views
- ▶ In other words, you will not be able to search in the TcSE module for objects that exist in a TcExx database, and you cannot edit the properties of a TcSE requirement from TcExxx
- ▶ You can only link OBJECTS to OTHER OBJECTS



# Understanding the Integration



- ▶ Because there is little common ground between the applications, an object must be created in one application that reflects information about an object in the other application
- ▶ The objects that reflect this information are called *Proxy objects*
- ▶ Each application can associate its objects with these Proxy objects to create relationships that show requirement traceability
- ▶ Creating a link is as easy as a “Copy & Paste”
- ▶ **Each link is 1-way, and focused toward the users of one application or the other, depending on how the link is created** (more on that later)



- ▶ There are 2 different link types:
  - ▶ WOLF Linking
  - ▶ Proxy Linking
  
- ▶ **UGS Services uses the more robust Proxy links.**
  - ▶ **The primary reason is that WOLF links do not dynamically update to reflect changes that are made to either of the objects being linked. Proxy links do not have this limitation.**



# 1<sup>st</sup> Type: WOLF Linking



- ▶ WOLF (Web Object Linking Facility)
- ▶ These links are created by selecting an object in one application and using the appropriate menu option to find and select an object in another application and link to it
  - ▶ Pros:
    - ▶ Consistent links that do not vary
    - ▶ Easily configured
  - ▶ Cons:
    - ▶ Limited information displayed in Enterprise/Engineering
    - ▶ Cumbersome menu interface
    - ▶ **Links are not dynamic (they do not reflect changes)**



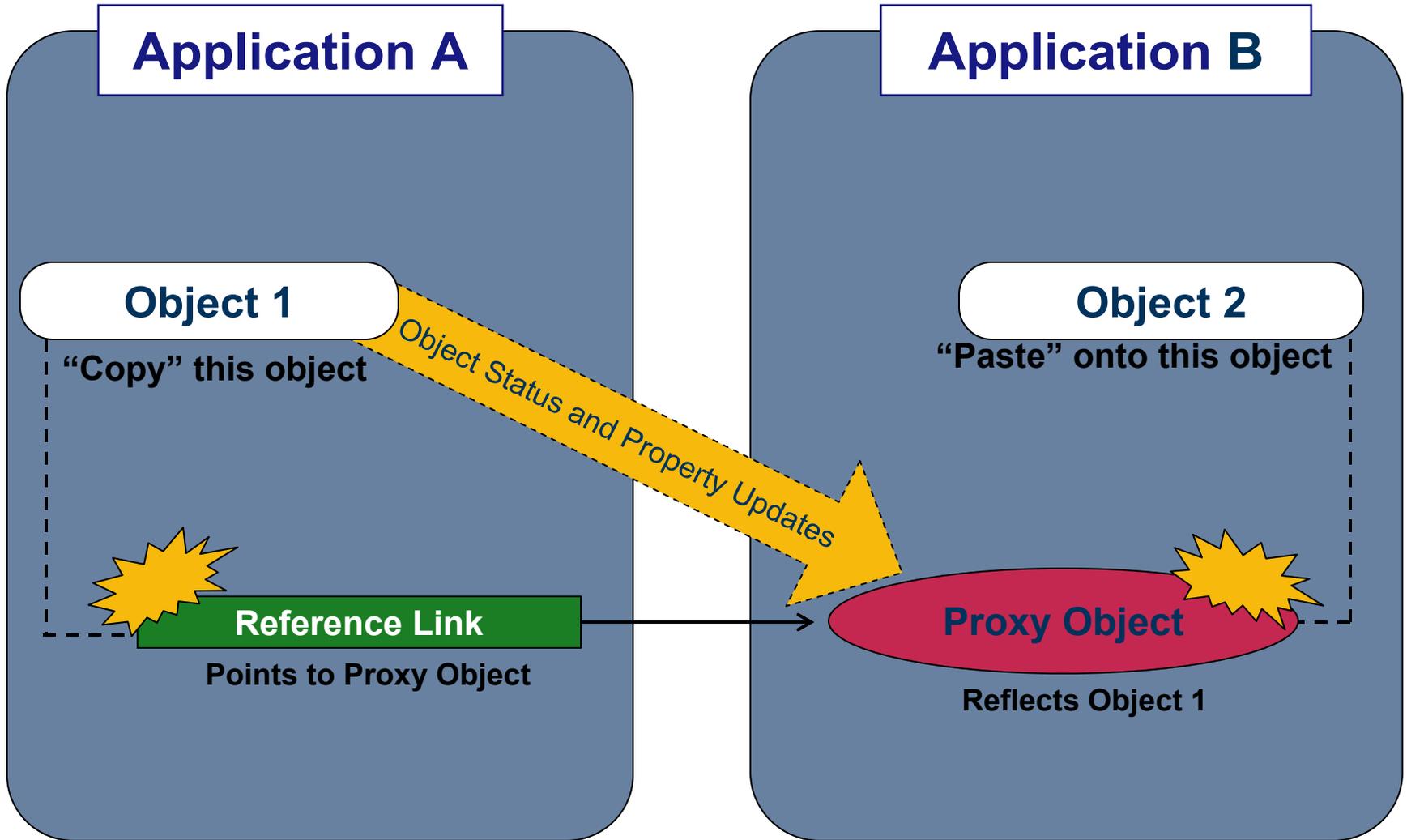
## 2<sup>nd</sup> Type: Proxy Linking



- ▶ Proxy links are created by performing a “Copy” on an object in one application, followed by a “Paste” onto an object in another application
- ▶ Upon creating a Proxy link, a Proxy object gets created in the second application where the “Paste” was performed. This Proxy object represents the original object in the first application.
- ▶ The Proxy object reflects certain properties of the original object, some of which can be configured

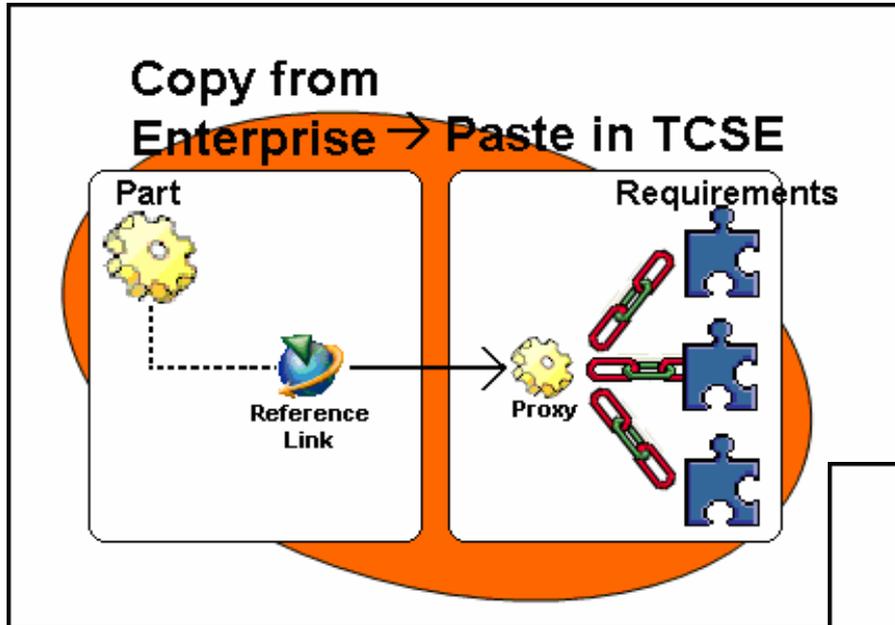


# Proxy Linking Diagram

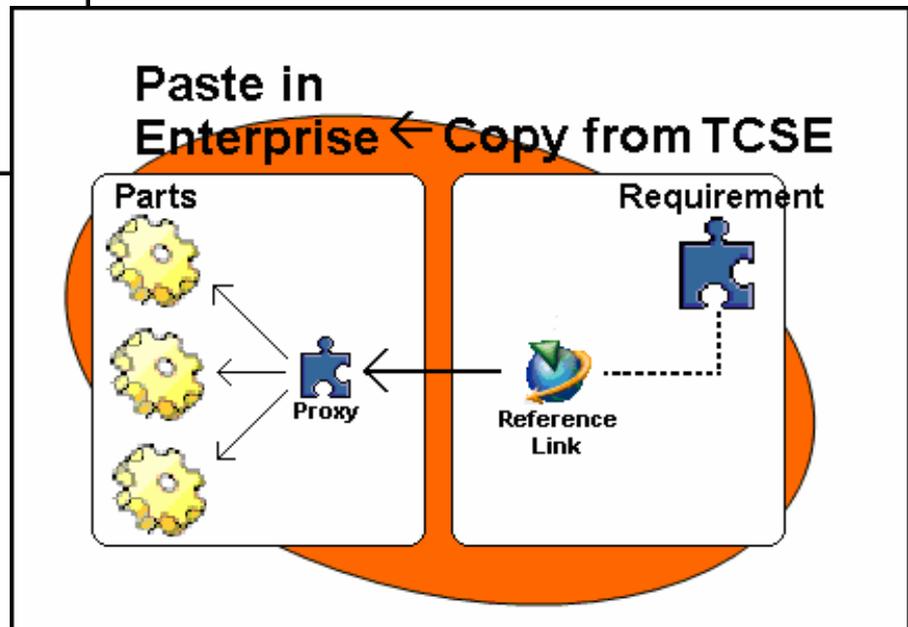




# Proxy links are dependant on which application the link is initiated from



The application where “*Paste*” is performed gets the Proxy object



The application where “*Copy*” is performed only gets a reference link



- ▶ Pros:
  - ▶ **Dynamic Links (objects are updated to reflect changes)**
  - ▶ Very simple “Cut & Paste” usability
  - ▶ More flexibility
- ▶ Cons:
  - ▶ More complex
  - ▶ Properties are displayed in only one application or the other, depending on direction of the link
  - ▶ Requires additional setup



# Proxy Object Properties



## A Requirements Proxy

 **REQ0024,A,1**

Summary Assignment Properties

Name: REQ00  
 Requirement Id: REQ00  
 External Application Name: Teamc  
 Creator: davies  
 Official?: false  
 Rev: A  
 Checked Out?: false  
 Seq: 1  
 Owner: Enterp  
 Project Name:  
 Frozen?: false  
 Superseded?: false  
 Life Cycle State: Autho  
 Responsible Party: davies  
 View Network:  
 Starting View Name:  
 Frozen View List:  
 Created: 11/28  
 Last Modified: 11/28  
 External ROIN: 0043  
 External Name: REQ2e  
 External Owner: integr  
 External Project: ERI te  
 External Text:  
 External Change Time: 11/28  
 External Change User: davies  
 External Defining Requirements:

.ssoapplet.SsoApplet started

Proxy objects have 2 types of properties:

- ▶ **Native** properties are assigned by the application in which the Proxy was created
- ▶ **External** properties are copied from the original object
- ▶ **Some of the External properties that get copied over can be configured, but ONLY when the user Copies from TcE and Pastes into TcSE**

## An Enterprise Proxy

Name	
Attachment Count	0
Change Time	11/28/05 8:49 AM
Change User	davies
Create Time	11/28/05 8:48 AM
Create User	davies
External: Class	Component
External: CreationDate	2005/11/28-14:44:43:09
External: LastUpdate	2005/11/28-14:49:01:18
External: Nomenclature	ENT2reqcopytopaste
External: obj_name	ENT2req_CopytoPaste,A
External: OwnerName	davies
External: PartNumber	ENT2req_CopytoPaste
External: RespParty	davies
External: Revision	A
External: Sequence	1
External: StdPartIndicator	false
External: Subtype	Teamcenter Enterprise P
Full Name	VENT2req_CopytoPaste,y
GUID	tcentERI
Name	ENT2req_CopytoPaste,A
OID	MTIObjectHandle-0002-1
Project	ERI testing
Reserved By	
Security Profile	
Subtype	Proxy
Trace Link Count	1
Type Name	Proxy
WhereUsed Object Count	0



# Installing the Integration



- ▶ Detailed installation instructions can be found on each Teamcenter CD
- ▶ Integration requires that the Teamcenter Application Registry be installed on the application server
- ▶ Each Teamcenter instance registers itself with the Application Registry and communicates with it to find other installed Teamcenter applications
- ▶ Simply deploy the *ApplicationRegistry.war* file, also available on each Teamcenter CD
- ▶ Variables can be configured, such as log file path and .xml storage directory



# Installation: TcSE



http://daldavies01:8080/tcr/ugs/tc/req/configtcr.jsp - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print

Links Reference Tableture Stuff UGS Accounts

Address http://daldavies01:8080/tcr/ugs/tc/req/configtcr.jsp

Go

Simply set these variables on the Web Application Configuration page to configure TcSE for linking

<input type="checkbox"/>	4999	trying to find an available port on which to start it's socket service.
<input type="checkbox"/>	4000	Start of the port range the client will search when trying to find an available port on which to start it's socket service.
<input type="checkbox"/>	TCRequirements	ID that the single sign-on service(SSO) uses to identify Requirements.
<input type="checkbox"/>	false	If true, Teamcenter 2005 for systems engineering will redirect login requests to the SSO login service.
SSO.Enabled	false	
SSO.LoginURL	http://ssoserver:8080/tcs	http://ssoserver:8080/tcssols URL to the SSO login service.
SSO.ServiceURL	http://ssoserver:8080/tcs	http://ssoserver:8080/tcssoservice URL to the SSO service.
WOLF.AppGUID	TcSE7	Unique Teamcenter Requirements application instance ID - used when registering with the WOLF directory service. Note, if this ID is changed, existing WOLF links to objects in this application instance will be broken.
WOLF.AppIP	DALDAVIES01	Application address - used when registering TcR with the WOLF directory service.
WOLF.AppPort	8080	Application port (e.g. 8080) - used when registering Teamcenter Requirements with WOLF directory service.
WOLF.AppProtocol	http	Application protocol (e.g. http) - used when registering Teamcenter Requirements with WOLF directory service.
WOLF.AppRegistryURL	http://DALDAVIES01:8080	URL of WOLF application registry service.

Update

Install Security Files

Clear

Reset to Default

Cancel



# Installation: Enterprise



- ▶ Requires ERI (Enterprise-Requirements Integration) server module and ERI thin client solution
- ▶ Must create new a new vault and background user

Name	Value	Comment	Origin
EPI_CLIENT_APP_NAME	tcEntEPI		Global
EPI_CLIENT_HOST	DALDAVIES01	where Enterprise	Global
EPI_CLIENT_PORT	8080		<Corporate>
ERI_CLIENT_APP_NAME	http://DALDAVIES01		<Corporate>
ERI_CLIENT_HOST	DALDAVIES01	where Enterprise	Global
ERI_CLIENT_PORT	8080		<Corporate>
ERI_PROTOCOL	http		Global
EVENT_LOG_BASE_DIRECTORY	D:\EntLogs\		Global

**Server Module Variables**

Req	Name	Value
<input type="checkbox"/>	showGraphicHistory	true
<input type="checkbox"/>	adminEditorDeploymentURL	http://DALDAVIES01:8080/tc...
<input checked="" type="checkbox"/>	TeamcenterApplicationRegistryURL	http://DALDAVIES01:8080/A...
<input checked="" type="checkbox"/>	TeamcenterEnterpriseInstanceGUID	tcEnt50
<input checked="" type="checkbox"/>	TeamcenterProjectInterfaceIntegration	change_me
<input checked="" type="checkbox"/>	TeamcenterRequirementInterfaceIntegration	true
<input checked="" type="checkbox"/>	TcEntIntegrationUser	backgrounderi
<input checked="" type="checkbox"/>	TcEntIntegrationUser_Pwd	backgrounderi
<input type="checkbox"/>	RMC_Sendet_Encoding	8859_1

**Client Solution Variables**



# Installation: TcEngineering



- ▶ Must have web server running Web Access (TcEng 2005 requires 4-tier installation)
- ▶ Deploy the EngChooser servlet for WOLF linking
- ▶ Configure AIWS (Application Interface Web Service) to accept invocations from remote Teamcenter applications
- ▶ Configure Teamcenter Linking
- ▶ Set **portal\_user.properties**
  - ▶ enableDragAndDrop=true (for rich client only)
- ▶ Set dedicated properties in **iman.env**



# Summary: Integration Capabilities



- ▶ 2 possible types of links:
  - ▶ WOLF links
  - ▶ Proxy links
- ▶ UGS urges linking by Proxy
- ▶ Proxy linking is one-way, and focused toward the users of one application or the other -- crucial information is only displayed in the application where the “Paste” action is performed.
- ▶ Proxy objects mirrors certain properties of the original object they reflect
  - ▶ Properties are updated dynamically
  - ▶ They can be customized for client-specific properties, but **ONLY** when the Proxy resides in TcSE



**UGS**

*Transforming the  
process of innovation*



[www.ugs.com](http://www.ugs.com)

**Jacob Davies**  
TcSE Implementation Engineer  
[jacob.davies@ugs.com](mailto:jacob.davies@ugs.com)