Usability Features in SAP NetWeaver PI 7.1 Development and Configuration Times

Applies to:

SAP NetWeaver PI 7.1

Summary

SAP NetWeaver PI 7.1 includes a wealthy set of development utilities that together with the runtime stability and performance make this SAP NetWeaver component the natural choice for your process integration scenarios.

This document will detail some of those development and configuration time utilities -many of them newly introduced in PI 7.1- and show how they help to increase productivity or administration capabilities to support large development teams.

You might be already familiar with some of them, but I think it's worth taking a look.

Author: Daniel Horacio Bianchin

Company: SAP

Created on: October 2007

Author Bio

Daniel has been working with SAP technology since 1996, and in 1997 he joined SAP. Over the last 10 years he has worked on more than 35 projects plus consulting, training and presales activities.

At SAP, he started as an ABAP and RFC (C++) consultant later began to work with ALE interfaces, doing quality assurance, development optimization, in-house software development and design, and then CRM Internet Sales implementation and interface architecture design. After that, he has been leading development and integration teams using SAP BC and XI. In 2005 he joined the SAP NetWeaver Regional Implementation Group.

Table of Contents

Introduction

Usability and Development and Configuration Administration

Good development administration tools result in more stable software (reducing downtimes) and reducing maintenance efforts.

Also a lack of administration tools makes a central development environment nearly impossible to use.

These tools are completely necessary in terms of eSOA, because the number of integration objects is big, and numbers of diverse organizational units that interact in a unique development and configuration environment are pretty heterogeneous.

Consider that PI 7.1 also introduces decentralized processing, what makes its processing capabilities unbound at runtime, so the possibilities that important companies share a global modeling, development and configuration environment across different subsidiaries, branched, etc., to later uniformly apply the patterns in are high.

Before starting with the list of items, it might be interesting to refresh the PI implementation methodology, so as to also to remember the steps and roles involved. Consider that this methodology is still not updated to PI 7.1, but still useful as reference: http://service.sap.com/~sapidb/011000358700006503452006E.

Objects Tree Filtering and Positioning

One important set of features is the "objects tree" and its positioning and filtering capabilities.
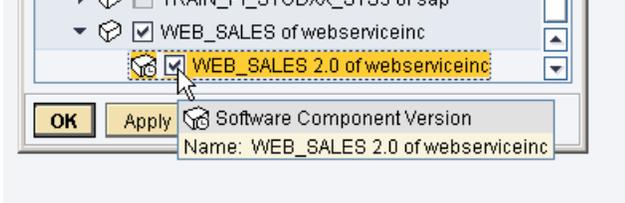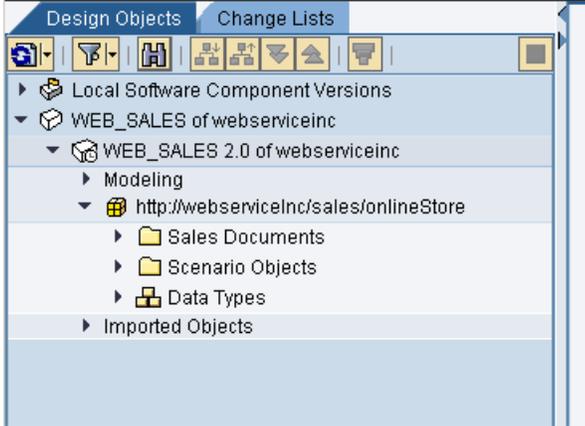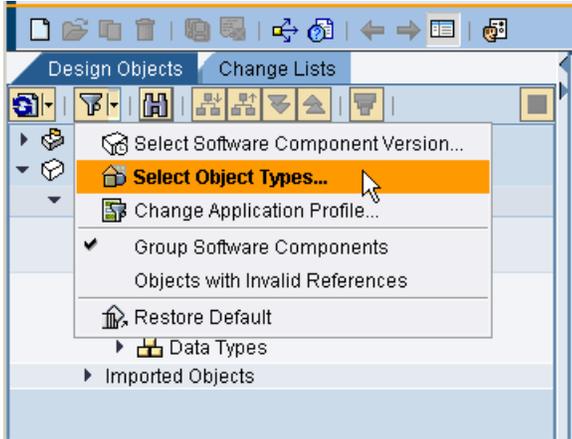
Just consider a list of software components like this one. Every time you start the ES Repository, you should scroll and open your particular software component and software component version.

If the list keeps on growing, it is not unusable, but the time spent in positioning and the possibility or error increases.
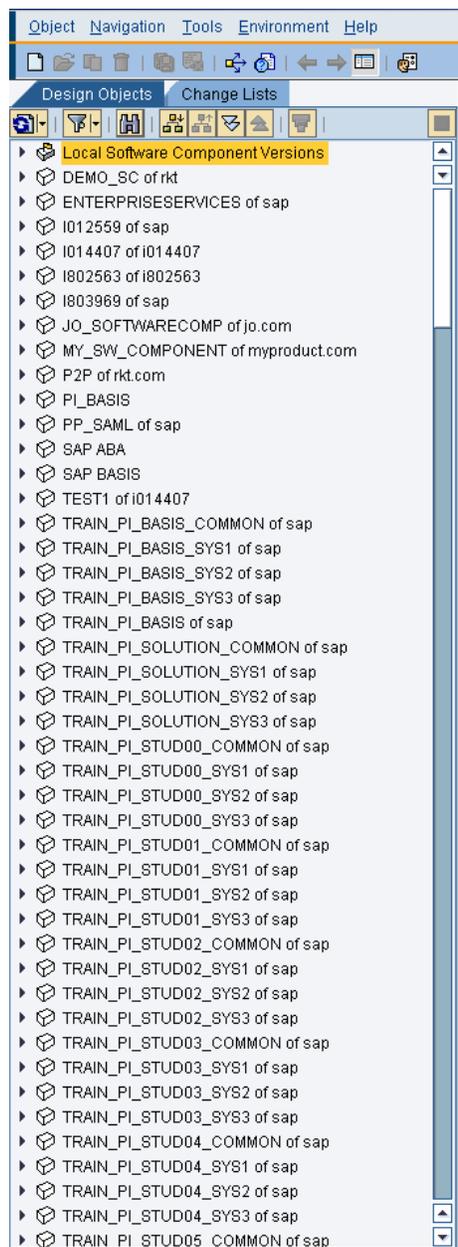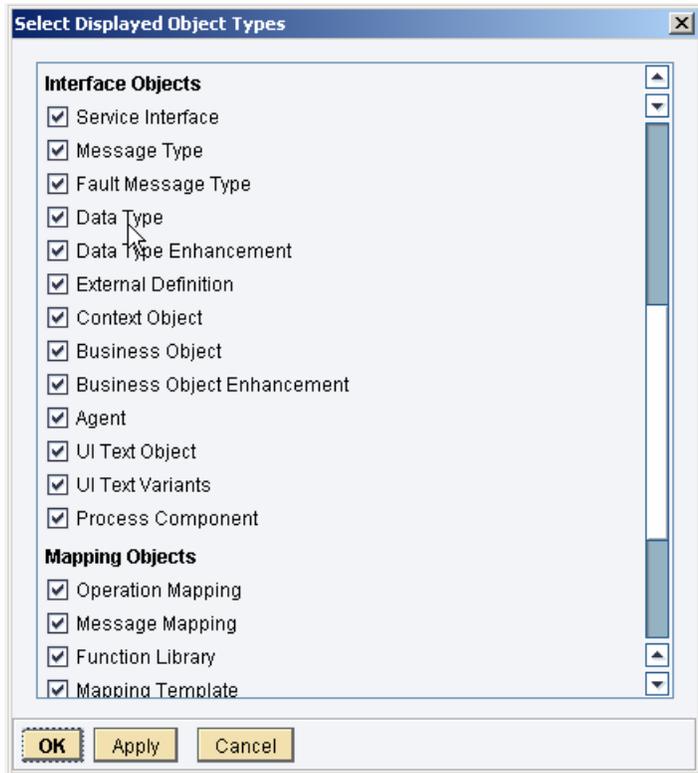
A nice feature to avoid it, is applying the filter functionality. It allows you to select just the set of software component and software component version you are interested in, in a particular moment.
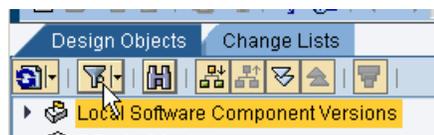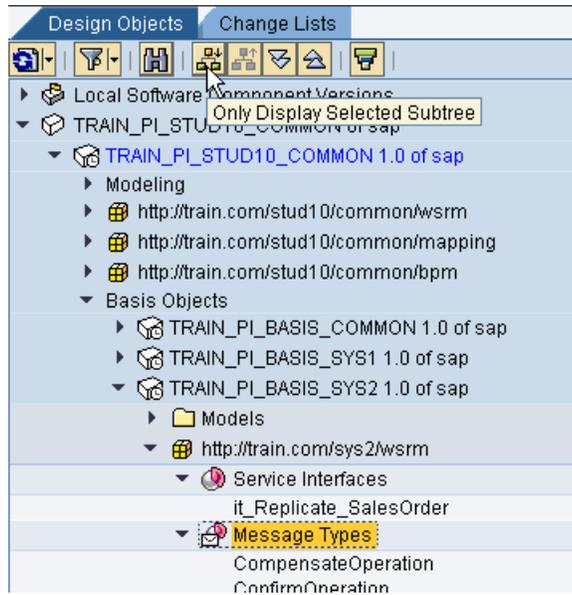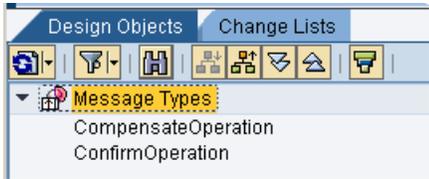
Execute the function and simply unselect the whole tree…	**Select Software Components** ☒ ▼ ☐ Local Software Component Versions ▶ ☐ Local Software Component Versions ▶ ☐ DEMO_SC of rkt
Then, select the software components or particular software component versions you are interested in.	▶ ☐ TRAIN_PI_STODXX_ST33 of sap ▼ ☑ WEB_SALES of webserviceinc ☑ WEB_SALES 2.0 of webserviceinc **OK** **Apply** Software Component Version Name: WEB_SALES 2.0 of webserviceinc
As result, you get your "virtual" view of the repository, at software component and software component version level.	Design Objects Change Lists ▶ Local Software Component Versions ▼ WEB_SALES of webserviceinc ▼ WEB_SALES 2.0 of webserviceinc ▶ Modeling ▼ http://webserviceinc/sales/onlineStore ▶ Sales Documents ▶ Scenario Objects ▶ Data Types ▶ Imported Objects
Another nice idea is to filter the object types you are interested in.	Design Objects Change Lists ▶ Select Software Component Version… ▼ **Select Object Types…** Change Application Profile… ✔ Group Software Components Objects with Invalid References Restore Default ▶ Data Types ▶ Imported Objects

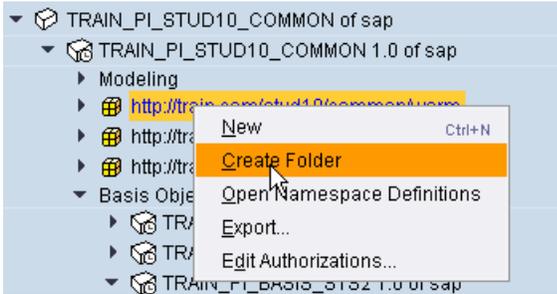|	
Easily click on the objects you are not interested in, and they will also be filtered from the tree for you.	 **Select Displayed Object Types** **Interface Objects** ☑ Service Interface ☑ Message Type ☑ Fault Message Type ☑ Data Type ☑ Data Type Enhancement ☑ External Definition ☑ Context Object ☑ Business Object ☑ Business Object Enhancement ☑ Agent ☑ UI Text Object ☑ UI Text Variants ☑ Process Component **Mapping Objects** ☑ Operation Mapping ☑ Message Mapping ☑ Function Library ☑ Mapping Template OK Apply Cancel

Of course, the well known tree positioning capability is still there…	Design Objects Change Lists
	
	▶ Local Software Component Versions
	Only Display Selected Subtree
	▼ TRAIN_PI_STUD10_COMMON of sap
	  ▼ TRAIN_PI_STUD10_COMMON 1.0 of sap
	    ▶ Modeling
	    ▶ http://train.com/stud10/common/wsrm
	    ▶ http://train.com/stud10/common/mapping
	    ▶ http://train.com/stud10/common/bpm
	    ▼ Basis Objects
	      ▶ TRAIN_PI_BASIS_COMMON 1.0 of sap
	      ▶ TRAIN_PI_BASIS_SYS1 1.0 of sap
	      ▼ TRAIN_PI_BASIS_SYS2 1.0 of sap
	        ▶ Models
	        ▼ http://train.com/sys2/wsrm
	          ▼ Service Interfaces
	            it_Replicate_SalesOrder
	          ▼ Message Types
	            CompensateOperation
	            ConfirmOperation

For example, it is possible to select just the Message Types, if you are not going to work with any other objects..	
By default, you will have your navigation configuration considered after Integration Builder restart. You can avoid it if you want.	

Folders in the Enterprise Services Repository and Integration Directory

As you might have realized, in PI 7.1 the default object grouping by object types is not created until either an object of that particular type or a folder is created. Folders are more flexible and allow you to structure in a more convenient way in accordance with your requirements. Suppose you want to create a business object / object type hierarchy, or group determined data types under another particular folder, everything is possible. To create the folder, simply right click on the parent object (namespace or folder) and execute "Create Folder", then name it.	

Here you see a business object / object type hierarchy. Also the service interfaces, since they share most of the object types, are assigned to intermediate folders. That is, not only leaf nodes are allowed to contain objects.	
There are two additional good features regarding folder in ES Repository and Integration Directory. One is the option to execute massive commands at folder level for the objects contained. The other is the possibility to work with authorizations at folder level, what increases the granularity of roles.	
Suppose only some roles are allowed to work on particular mapping objects, you just execute "Edit Authorization" on this folder and change them as required.	
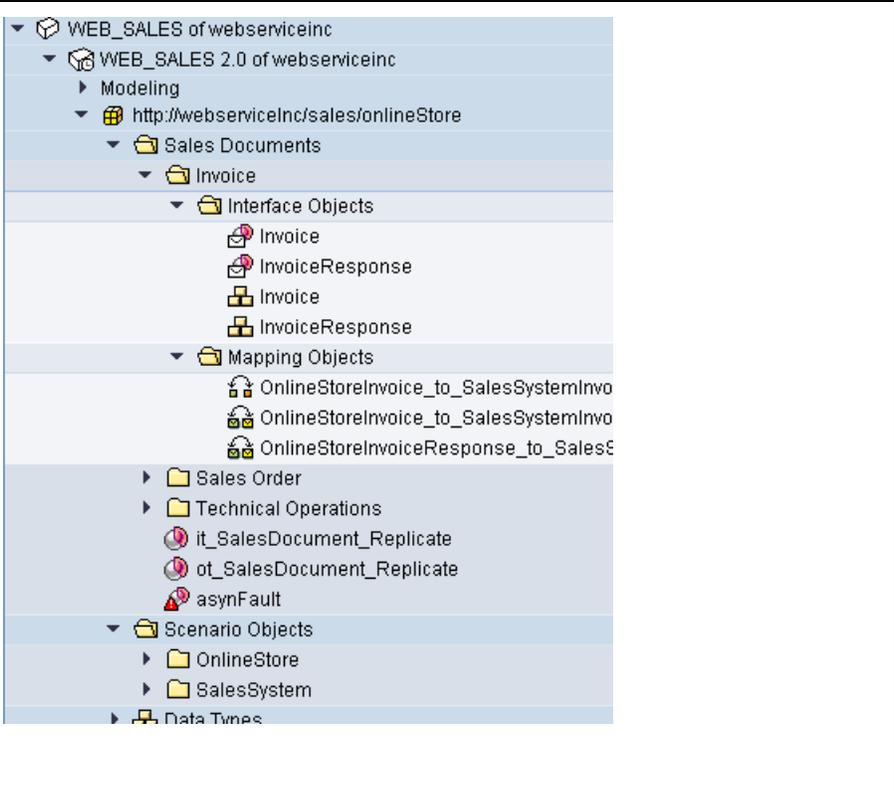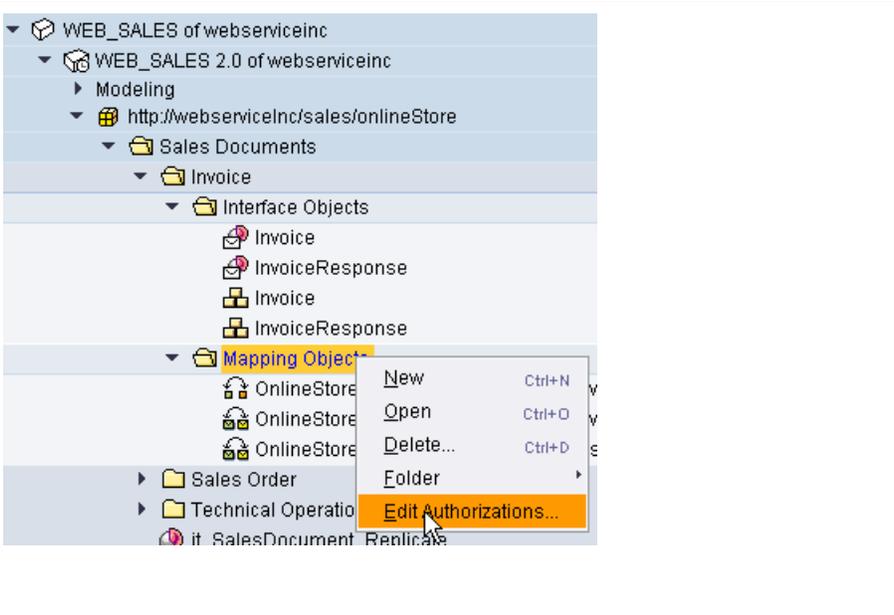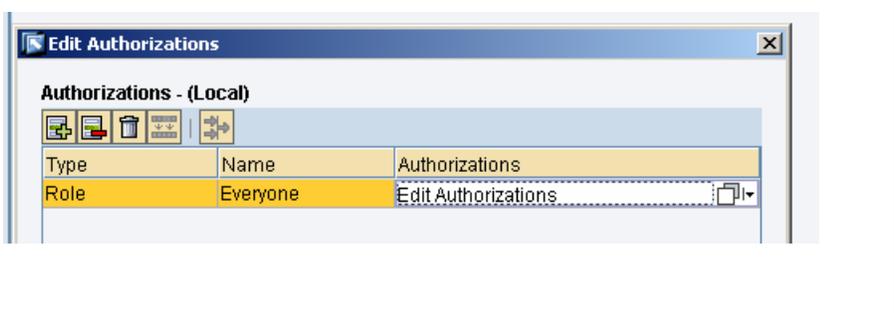
In the Integration Directory folder are optional. You can either show or hide them.

To manage your objects, you can use folders, configuration scenarios or objects type groupings.

Objects types grouping have been available for a long type, also configuration scenarios.

There have been a conceptual change regarding objects representation in the tree; they do not appear in the tree more than once anymore. That means that configuration scenarios do not include references to objects and list them in the tree, but rather show them in a list in the main screen area when you open the scenario.

Also the folders don't have a reference to an object, but the object itself is contained in the folder. So if you try to copy an object from one folder to another, the source folder does not contain the object anymore.
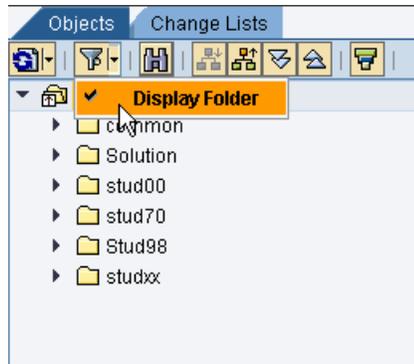
For that reason, it is a good practice to create "common" or "basis" folders to accommodate shared objects like business systems, channels, or whatever makes sense in your installation.

Later you will be able to give particular authorization, for example to administrators, to work on that folder and maintain authentication information, while other roles will be able to deal with logical routing configuration only.
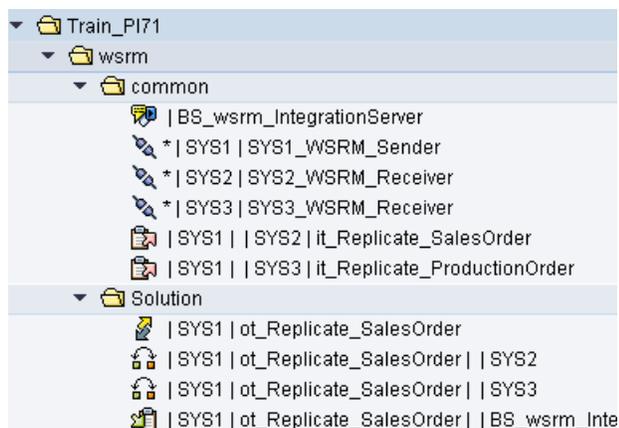
As mentioned before the configuration scenario is not representing the "referenced" object list in the tree anymore. To simulate that functionality you have to open the scenario and detach the window.

Configuration scenarios and folders are not the same thing, there is a very good blog on SDN regarding folder, you should go and take a look:

https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/7010
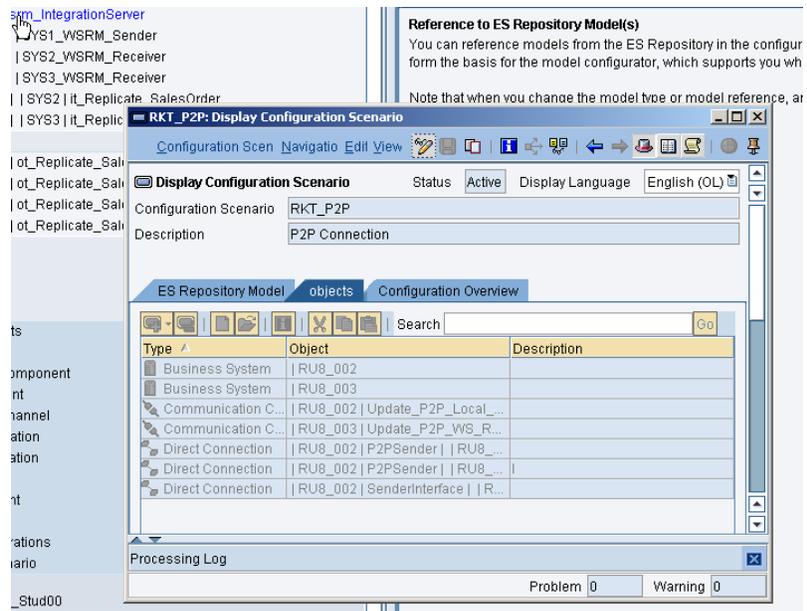
Software Component Version Relationships

This feature has also been available for a while. It is the possibility to describe relationships at SLD level. They are required if you want to comply with the naming and somehow with the methodology too: (ref: https://weblogs.sdn.sap.com/pub/wlg/5356)
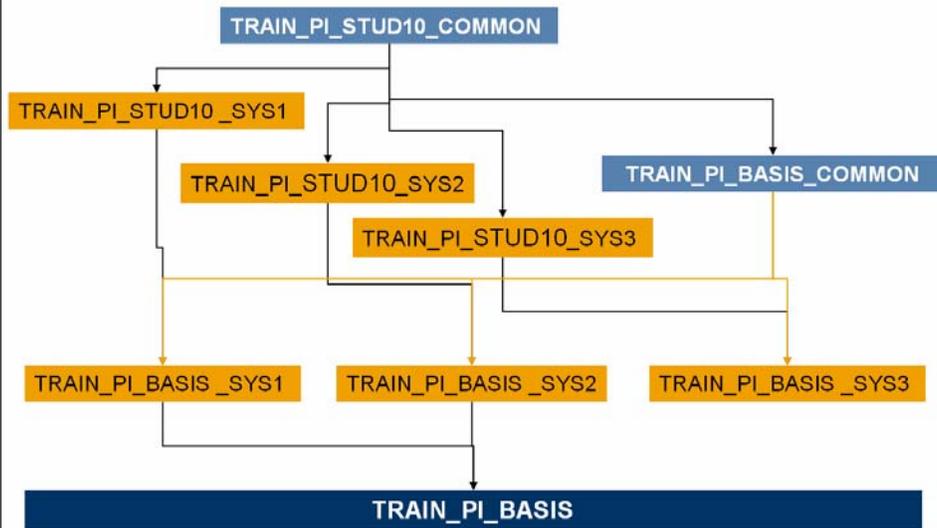
In the ES Repository, you will see the list of basis software component versions, and will be able to select object from them every time you need something for your root software component version.

Version handling is another powerful feature here; if you upgrade a basis one, you can generate a new version for the root to include the new functionality, while you keep the other basis relationships, in the same version as before.
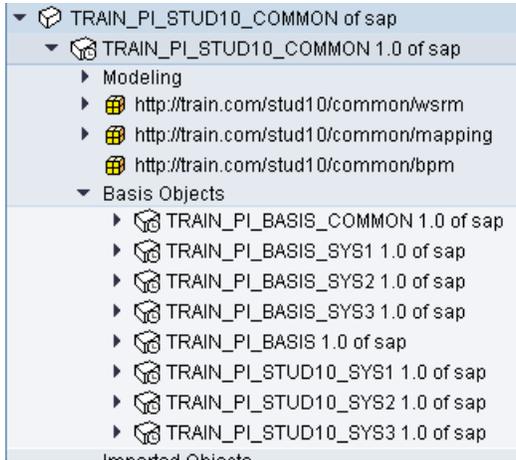
Object search, usage reference and objects navigation

In an eSOA architecture, you need powerful tools to manage all the objects. One of the tools is the well known SAP "Where-Used List" It allows you to see where a particular object is referenced, directly or indirectly also.

Another tool is the search utility. It is very comprehensible, supports wildcards and extended attributes.

Both the "Where-used List" and the search utility (actually, most of the ES Repository and Integration Directory objects) support the navigation from the listed object to its definition.

You simply double-click on it…

And you get the requested object in a new tabstrip.

Wildcards, context reference and Keyboard shortcuts

Wildcards become more important when the number of objects increases. It is pretty error prone to receive a long listing of similar objects from where you have to choose one. It is better to simply restrict the search with key information first.	**Copy Service Interface** **Target Object** **Service Interface** Name * it_Replicate_SalesOrder Namespace * http://train.com* Software Component Version * TRAIN_PI_STUD10_COMMON 1.0 of sap Folder ☐ With All Dependent Objects
Then, using the dropdown box with the mouse or simply using the "F4" key from your keyboard you get the list of objects.	F3 F4 F5 $ % ^
The search criteria will be shown first and then the results from where you can type an addition search criterion or easily double-click on the desired entry.	**Choose Namespace** ✕ Hierarchical Search Help Tabular Search Help **Search Criteria** | Attribute	Value
Software Component Version	TRAIN_PI_STUD10_COMMON 1.0 of sap
Namespaces for RFC or IDoc Also	No
NAMESPACE	http://train.com* | ▲ ▼ **Search Result** 🔁	Search [ ] [Go] Namespace ▲ http://train.com/stud10/common/bpm http://train.com/stud10/common/mapping http://train.com/stud10/common/wsrm

|	
Besides the "Tabular Search" tabstrips there is another "Hierarchical Search" just in case you need to know the object classification information for the one you are looking for.	
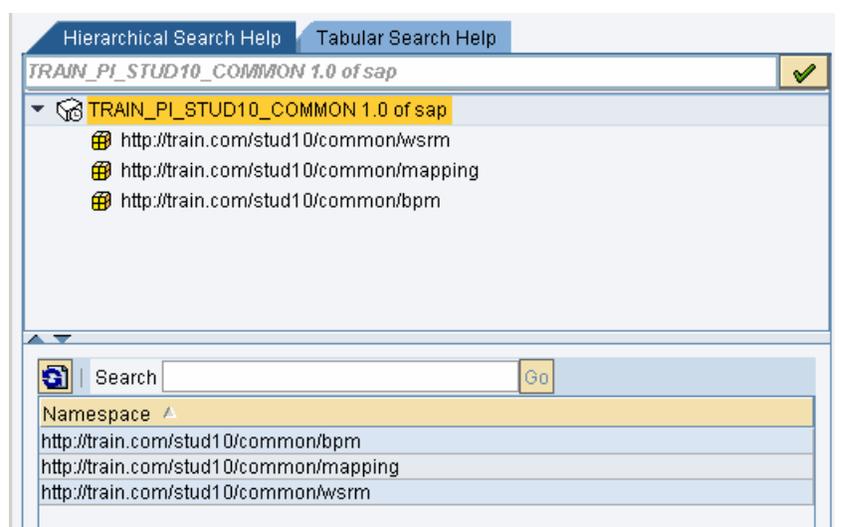
Authorizations

|	
It was possible before to define authorizations at object level for particular roles. This functionality has been enhanced in this version to better support large development teams.	
As shown before, folders allow you to group and handle authorizations on the granularity level that better fits you needs.	
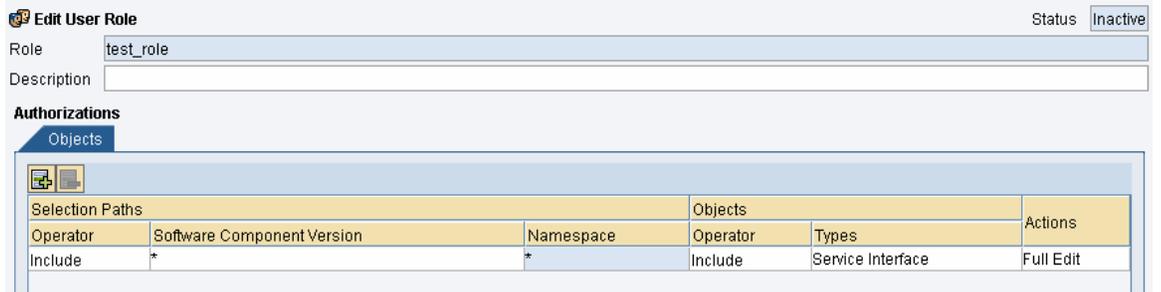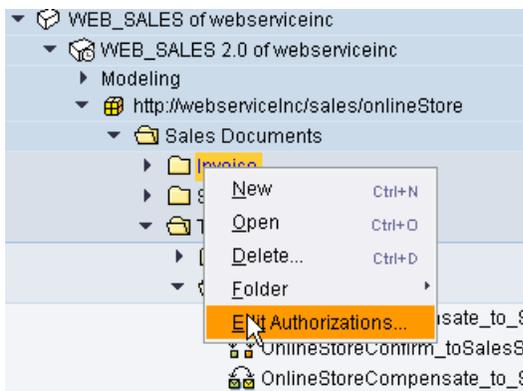
Lifecycle Management: Objects Activation

Integration builder objects allow you to work with inactive objects versions, all other users will see the active version if available.

So it gives you the opportunity to modify a new inactive version while a previous one is used in the meanwhile (without creating a new version).
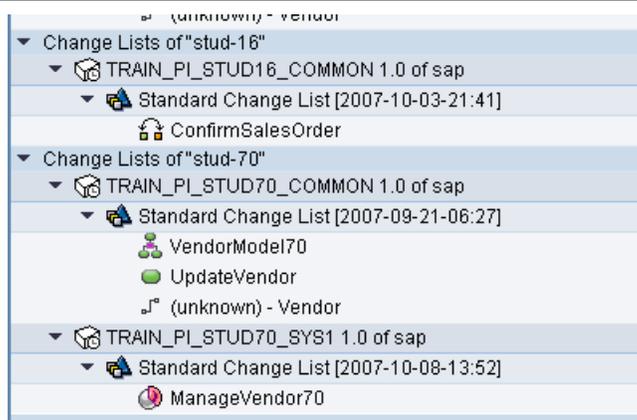
Inactive versions can be transferred among system users.

Select the change list you are interested in, and transfer the inactive versions to your own user.
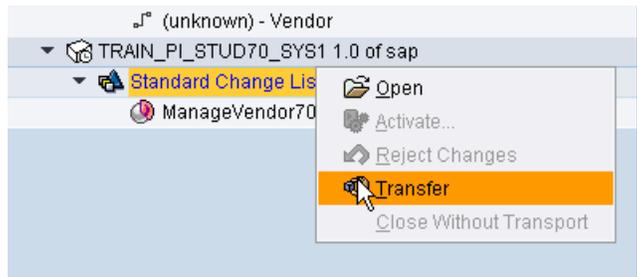
The activation can be executed from different places, you can right-click on the inactive object and choose the option.
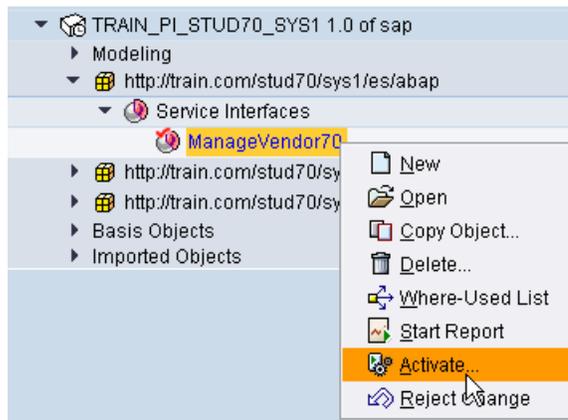
Also the activation is present in the object display or modification functionality.

Of course it is still possible to select you change list and do the massive activation as before.
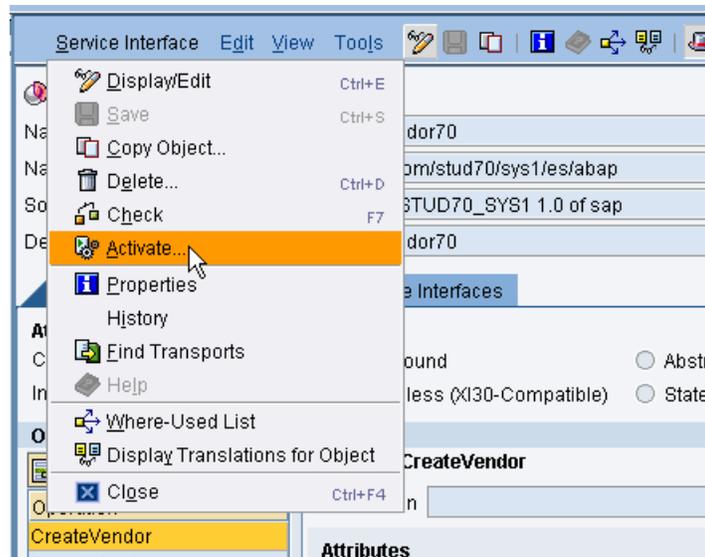
Even when it won't be probably available with the first PI 7.1 general availability release, it will be possible later using CTS+ to include most of SAP NetWeaver objects in a common transport orders instead of using different technologies for each. For example, Enterprise Services Repository objects will be transported together with ABAP mapping classes

Local Objects

Local objects allow you to create software components for ES Repository testing proposes, you don't need to create the object in SLD. You directly create them from the general object creation form.
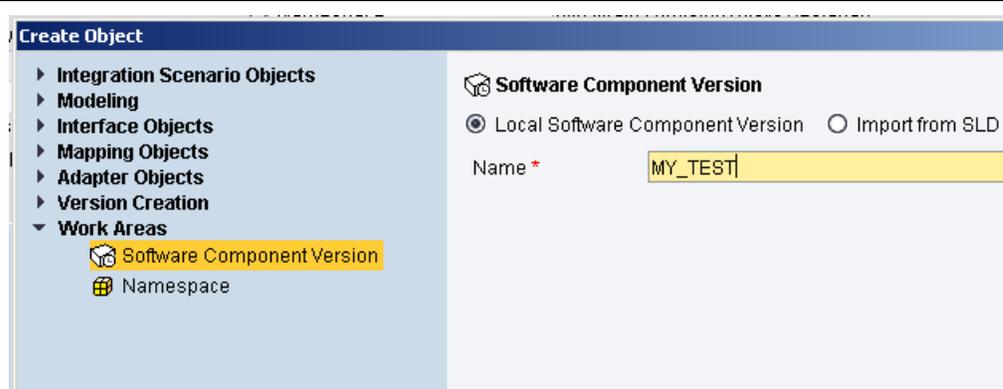
Summary

Keeping a simple virtual environment for each developer

For sure the list of available utilities (considering the new ones included) will simplify development and maintenance. Consider how easy it is to select objects form a list of five, instead of browsing long listing of objects with nearly the same name! Also, the functionalities are completely intuitive and don't require further explanations to understand how to use them.

Customer Engagement Lifecycle support

Usability, simplification, naming, methodology, stability, scalability, performance, etc. Yes, there are many factors to take into account when you choose your process integration application, and development and

configuration topics are not the exception. A well understood eSOA methodology strategy requires powerful tools and the evolution of the Integration Repository (the ES Repository) and Integration Directory have the capability to handle large eSOA implementations.

Related Content in SDN

- Folders: https://weblogs.sdn.sap.com/pub/wlg/5356

- PI scenarios implementation: https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/7010

- SAP NetWeaver PI 7.1 Overview: https://www.sdn.sap.com/irj/sdn/weblogs?blog=/pub/wlg/6204

Copyright