How To Make Consistent Changes to Software Component Versions in SAP NetWeaver XI

Version 1.00 – January 2006

Applicable Releases:
SAP NetWeaver '04 Support Package Stack 13
These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies (“SAP Group”) for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

SAP NetWeaver “How-to” Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.
## Content

1. Scenario ........................................................................................................................................ 2
2. Introduction ................................................................................................................................. 3
3. Use Cases and Step-by-Step Solutions ....................................................................................... 4
   3.1 Use Case: ‘Changing Non-Key Elements’ ............................................................................. 4
      3.1.1 Adapt the SWCV (SLD) ................................................................................................. 4
      3.1.2 Import the SWCV (Integration Repository) ................................................................. 4
      3.1.3 Import the SWCV (CMS) ............................................................................................ 5
   3.2 Use Case: ‘Changing the SLD Key’ ....................................................................................... 6
      3.2.1 Cleanup (CMS) ............................................................................................................ 6
      3.2.2 Adapt the SWCV (SLD) ............................................................................................... 7
      3.2.3 Import the SWCV (Integration Repository) ................................................................. 13
      3.2.4 Import the SWCV and Prepare for Transport (CMS) ................................................ 13
   3.3 Use Case: ‘Change the XI Key’ ............................................................................................ 15
4. Alternatives ................................................................................................................................... 16
5. Appendix ....................................................................................................................................... 17
1 Scenario

The Change Management Service (CMS) is part of SAP NetWeaver Development Infrastructure (NWDI). This service enables you to configure your transport landscape and manage the transport of objects between different systems. From the point of view of transport, the System Landscape Directory (SLD) is used as the central storage location for Software Component Versions (SWCVs). These SWCVs are used in Integration Repository and CMS to partition and transport Process Integration content (XI content).

1.1 Figure 1: Communication Between SLD, CMS, and Integration Repository (IR)

This How-To Guide describes the steps that are necessary to change or replace an SWCV.

Since the SWCV information is used in the SLD, CMS, and in SAP NetWeaver XI, the consistency of the SWCVs within the landscape is essential for successful transport within the SAP NetWeaver XI landscape.

Note:
The SAP NetWeaver XI Integration Builder is the tool for storing and accessing information both in the Integration Repository (at design time) and the Integration Directory (at configuration time). Since there are no SWCVs in the Integration Directory, this How-To Guide always refers to the Integration Builder (Design) when accessing the Integration Repository, even if it is not mentioned explicitly.
2 Introduction

The SWCVs, which are maintained in the SLD, are also used in SAP NetWeaver XI to partition XI content, and within CMS to define transport tracks.

Therefore, the SWCV information is replicated to CMS and SAP NetWeaver XI by means of an SWCV import from the SLD.

This How-To Guide describes where, how, and in which order an SWCV must be changed or replaced to ensure consistency. This is especially important when a CMS track has already been established for this SWCV.

Inconsistent SWCV information (in particular, inconsistent GUIDs) is very likely to lead to the following problems:

- Duplicate entries for the same SWCV in the Integration Builder.
- Transport problems:
  - XI content, which is transported by means of a file transport, is not imported to the correct SWCV. Instead, a new node of the same SWCV appears in the Integration Builder navigation tree.
  - Old CMS transport requests can no longer be completed.
  - New CMS transport requests cannot be created despite the SWCV having been added to a CMS track.
  - CMS transports fail due to import or export exceptions.

Chapter 3 describes the main use cases and provides a step-by-step solution for each one. The solution for the second use case is more complex than that for the first use case. Therefore, chapter 4 describes an easier alternative with less manual steps which you should use in preference to the second solution.
3 Use Cases and Step-by-Step Solutions

Each SWCV can be identified by its two keys:

- SLD key (name, vendor, version, elementTypeId)
- XI key (SWCV GUID).

These keys must be consistent in the Integration Repository, SLD, and CMS.

3.1 Use Case: ‘Changing Non-Key Elements’

If only non-key elements of the SWCV, such as the caption, have to be changed, the following steps are sufficient:

3.1.1 Adapt the SWCV (SLD)

Adapt the non-key attributes in the SLD (Home → Content Maintenance → Subset: Component Information → Class: Software Component Version → <Select your Component>). Save your SWCV.

![SAP NetWeaver System Landscape Directory]

3.1.2 Import the SWCV (Integration Repository)

Import the SWCV from the SLD to the Integration Repository (Integration Builder (Design) → Tools → Transfer from System Landscape Directory → Import Software Component Versions)
The content of the SWCV in the Integration Repository will not be affected.
This step can be omitted if the SWCV has not already been imported to the Integration Repository. In some cases, it may be necessary to close and reopen the Integration Builder to refresh the navigation tree.

3.1.3 Import the SWCV (CMS)

Import the SWCV from the SLD to CMS (CMS → Landscape Configurator → Domain Data → Update CMS)

3.1.3.1 This step can be omitted if no CMS is connected.
3.2 Use Case: ‘Changing the SLD Key’

If the SLD key of an SWCV has to be changed then it is not sufficient to just perform the steps in the previous section.

Note that once the SLD key has been changed, the SWCV is treated as a completely new SWCV by the SLD and CMS. You should only change the SLD key without changing the SWCV GUID if you want to hide the change from the Integration Repository. This can be useful if a release transfer between an old and a new SWCV must be avoided. But since a mistake in the following manual steps can easily lead to inconsistencies, you are strongly advised to follow the alternative procedure, which is described in chapter 4.

You must execute the following main steps:

- Cleanup (CMS)
- Adapt the SWCV (SLD)
- Import the SWCV (Integration Repository)
- Import the SWCV and prepare for transport (CMS)

Those steps will be described in more detail in the following sections.

3.2.1 Cleanup (CMS)

If the SWCV is connected to a CMS transport then the following preparations are necessary:

1. Remove the entries that belong to your SWCV from the CMS import queues. If you omit this step then it will not be possible to remove the entries later on. The queues can be found in CMS \ Transport Studio. The relevant queues are Check-In, Development, Consolidation, Assembly, and Approval. In some older CMS versions, elements can only be removed from the queues by a successful import or assembly. Newer versions support the deletion of any entry, however.

   Note that no CMS transports for the given SWCV must be triggered until all the following steps have been executed. Otherwise pending transport requests, which can no longer be removed, will remain in the CMS queues.
2. Remove the SWCV from all CMS tracks (CMS → Landscape Configurator → Track Data → Delete Table Entry). Save the track.

3.2.1.1 Result:
The SWCV is no longer included in any CMS transports. The transports lists for this SWCV have been removed from all transport queues.

3.2.2 Adapt the SWCV (SLD)

This is the most important step. If the SWCV is not consistent within the SLD then it cannot be consistent within the other systems in the SAP NetWeaver XI landscape.

The following steps describe how to replace the old SWCV with a new SWCV, which has the same GUID as the old SWCV.

3. Open the SWCV editor for the old SWCV (SLD → Administration → Content Maintenance → Subset: Component Information → Class: Software Components Versions → click on your old SWCV) and copy the attribute GUID (for example, to Notepad) for later use.
4. Navigate to the associated instances and from there to the associated GUID instance.
5. Delete this GUID instance.
6. Navigate to the old SWCV (SLD → Administration → Content Maintenance → Subset: Component Information → Class: Software Components Versions → click on your old SWCV) and delete it.
7. Create the new SWCV (SLD → Software Catalog → Software Components → New Software Component). Do not use the content maintenance function (SLD → Administration → Content Maintenance) to do this. Otherwise the GUID will not be created automatically.
8. Navigate to the new SWCV (SLD ➔ Administration ➔ Content Maintenance ➔ Subset: Component Information ➔ Class: Software Components Versions ➔ click on your new SWCV). Replace the value of the attribute GUID with the one you copied in step 3 from the old SWCV. Save the SWCV.

9. Navigate to the associated instances and from there to the associated GUID instance.
10. Copy the caption to somewhere else (for example, to Notepad) for later use and delete the GUID instance afterwards.
11. Create a new GUID instance (SLD → Administration → Content Maintenance → Subset: Component Information → Class: Global Unique Identifier → New Global Unique Identifier). Enter the value of the GUID you copied in step 3 as the name, and the caption you copied in step 10 as the caption. Save the GUID instance.

12. Navigate to the associations of the GUID instance (SLD → Administration → Content Maintenance → Component Information → Global Unique Identifier → <select your new SWCV> → Associations).

13. Create a new ‘Software Component GUID’ association from this GUID instance to your SWCV (New Association → Association Type ‘Software Component GUID’ → Select your SWCV and then choose Associate)
3.2.2.1 Result:
Now the old SWCV has been removed from the SLD and a new SWCV with the same GUID has been created. The attribute GUID and associated GUID of the new SWCV are consistent.
3.2.3 Import the SWCV (Integration Repository)

The following steps are only necessary if the SWCV has already been imported into the Integration Repository.

14. Import the SWCV from the SLD to the Integration Repository again (Integration Builder → Tools → Transfer from System Landscape Directory → Import Software Component Versions).

Afterwards, only one entry for the SWCV should be displayed in the Integration Builder navigation tree. If multiple entries are displayed, then the GUID of the old and new version of the SWCV are different and therefore the new SWCV cannot be mapped to the old one. In this case, you must resolve the inconsistency in the SLD before reattempting the import to the Integration Repository.

3.2.3.1 Result:
The old SWCV definition is overwritten by the new one in the Integration Repository without any content being lost.

3.2.4 Import the SWCV and Prepare for Transport (CMS)

The following steps are only necessary if the SWCV is to be connected to a CMS transport.

15. Import the SWCV from the SLD to the CMS (CMS → Landscape Configurator → Domain Data → Select: Update CMS)
16. Add the SWCV to a CMS transport track (CMS → Landscape Configurator → Track Data → Select: Add SC).

17. Activate the CMS transport on the SAP NetWeaver XI administration page (XI Start Page → Administration → Repository → CMS Adjustment → select per Changelist Transport). In this way, you can create CMS transport requests for changelists.
3.3 Result:
CMS transports are enabled for the new SWCV.

3.4 Use Case: ‘Change the XI Key’

If both the XI key and SLD key are changed, then this is the same as creating a completely new SWCV, which is independent of the old SWCV for the SLD, CMS, and Integration Repository.

Therefore, no additional restrictions have to be considered.

It is not possible to change the XI key without simultaneously changing the SLD key.
4 Alternatives

In some cases, it is much easier to create a new SWCV, which is completely independent of the old SWCV, and to transfer the content of the old SWCV to the new one by release transfer (*Integration Builder → Tools → Release Transfer*).

You can then remove the old SWCV from the Integration Builder. Note that before you delete the SWCV, you must also remove the SWCV from the CMS tracks as described in section 3.2.1.

If necessary, you can create a CMS transport for the new SWCV.

Using release transfer reduces the number of manual steps and the likelihood of inconsistencies due to human errors.
5 Appendix

For more information, see:

- SAP Note 780297 - Troubleshooting using CMS for SAP NetWeaver XI
- CMS online documentation on the SAP Help Portal at help.sap.com/nw04 → Netweaver → Application Platform → Java Technology in SAP Web Application Server → Administration Manual → Administration of SAP NetWeaver Java Development Infrastructure
www.sdn.sap.com/irj/sdn/howtoguides