State of the Art: Maintaining Process Instructions with XSteps – Available Functional Enhancements

Applies to:

SAP ERP >= ECC 5.00, SAP R/3 Enterprise 4.70 Extension Set 2.0.

Summary

Using XSteps and Standard XSteps for maintaining the content of your PI Sheets reduces your efforts and costs. Learn from this article how the existing limitations could be compensated by the new available functional enhancements.

.

Author:	Dr. Uwe Dittes
Company:	SAP AG
Created on:	18 April 2007

Author Bio

Dr. Uwe Dittes made his PhD in Chemistry at the University of Heidelberg. After working several years in the chemical and pharmaceutical industry (development & production), he joined the development organization of SAP in 2001. He has a strong expertise in SAP PP-PI (especially master recipe, production versions, electronic batch record [EBR], execution steps [XSteps]) and xMII content development.

Table of Contents

Introduction

Maintaining process instructions for process operators (browser-based PI Sheets) or process control systems could be a tedious job if you do it via characteristic-based process instructions:

- You need to now much about the syntax of the process instruction characteristic (this is quite not easy)

- Copying process instructions of one phase into other phases is impossible – you have to enter the complete data again

- You can't define templates for process instructions that you want to reuse

- You can't simulate the PI Sheet and check the correct layout (without creating test control recipes)

- Generation scopes assigned to control recipe destinations can't be used due to regulated restrictions

SAP therefore developed the XStep environment – the "next generation of process instructions". With the help of the XStep editor the maintenance of process instruction gets easier and simpler:

Figure 1: A look into the standard XStep repository (transaction CMXSV)

- You can drag & drop, copy and paste your XSteps

- You can define your templates for process instructions as standard XSteps (SXS) in a SXS repository. They can be referenced and reused

- The PI Sheets can be simulated in the master recipe, in the process order and in the SXS repository. You'll see the final layout of your PI Sheet just after pressing the F8 button.

- Various generation scopes exist; each of the XStep elements that belong to an XStep tree can have an own generation scope

- Etc., etc.

Execution Steps (XSteps)

Using XSteps you can describe the precise execution of production processes and can group and structure this information in the form of modules. XSteps can be structured in hierarchical form. These form XStep trees.

The production steps described in XSteps are grouped for each control recipe and sent to the production execution level. Production can then be executed either manually or automatically. In the case of manual production, the system creates interactive PI sheets, which are then processed by process operators on the production line. In the case of automatic production, the production steps are executed by a process control system.

Standard XSteps (SXS)

Standard XSteps are XSteps that you create as master data in the standard XStep repository (SXS repository). Standard XSteps can be reused by creating a reference to them in the SXS repository from an application (for example, master recipe).

Using the references to standard XSteps (SXS reference), you can considerably reduce the time required to maintain the data. If, for example, there are changes to the standard processes, you do not have to make these changes in each individual master recipe. Instead, you make the changes centrally in the SXS repository. All master recipes, which contain references to these standard XSteps, are changed automatically.

XStep Editor

The XSteps editor provides you with a maintenance environment for XSteps and standard XSteps and can be accessed from the following applications:

- The standard XStep repository
- The master recipe
- The process order

Depending on the application, you can access different functions. For example, in the process order you can only create control recipes.

Limitations so far – areas of improvement

Though the XStep environment offers a highly flexible way to maintain your process instructions for your browser-based PI Sheets or your process control systems there have been some limitations:

- Released SXS versions could be changed without control (critical for the regulated environment)
- There was no possibility to find out if an SXS was referenced in master recipes or process orders
- The possibility to export only the XStep tree and not complete SXS versions, SXS or XStep folders doesn't be useful to distribute your SXS master data

Enhancements for the standard XStep repository

The following chapters describe functional enhancements that are available for the SXS repository to compensate the mentioned limitations.

Approval procedure for SXS versions (including digital signatures)

Current status management for SXS versions

SXS versions are connected to the SAP status management. These system statuses were previously defined for the SXS version (transaction BS23):

System status	Status	Short text
I8501	XR01	In Process
I8502	XR02	Rejected
I8503	XR03	Release for Test
I8504	XR04	Released
I8505	XR05	Obsolete
I8506	XR06	Flagged for Archiving
I8507	XR07	Archived

Figure 2: Overview about current system status for SXS versions

Changes of the system status for a SXS version will be executed if the corresponding business transactions are executed. Business transactions for the object type 'XSV' [SXS Repository – Version] are defined in transaction BS32 [Maintain processes]. The following business transactions for the status change of SXS versions were defined and shipped by SAP until now:

- XR02 (Reject), XR03 (Released for Test), XR04 (Release), XR05 (Declare as Obsolete), XR06 (Mark for Archiving) and XR18 (Delete Version)

The current system status of a SXS version allows business transactions that could be executed. They can be selected via the context menu of a version node. After execution of a selected business transaction the corresponding new status of the version is set. This new status allows the execution of other business transactions that are different from those assigned to the previous status. The standard scheme presents an overview about the interaction between system status and business transactions for SXS versions:
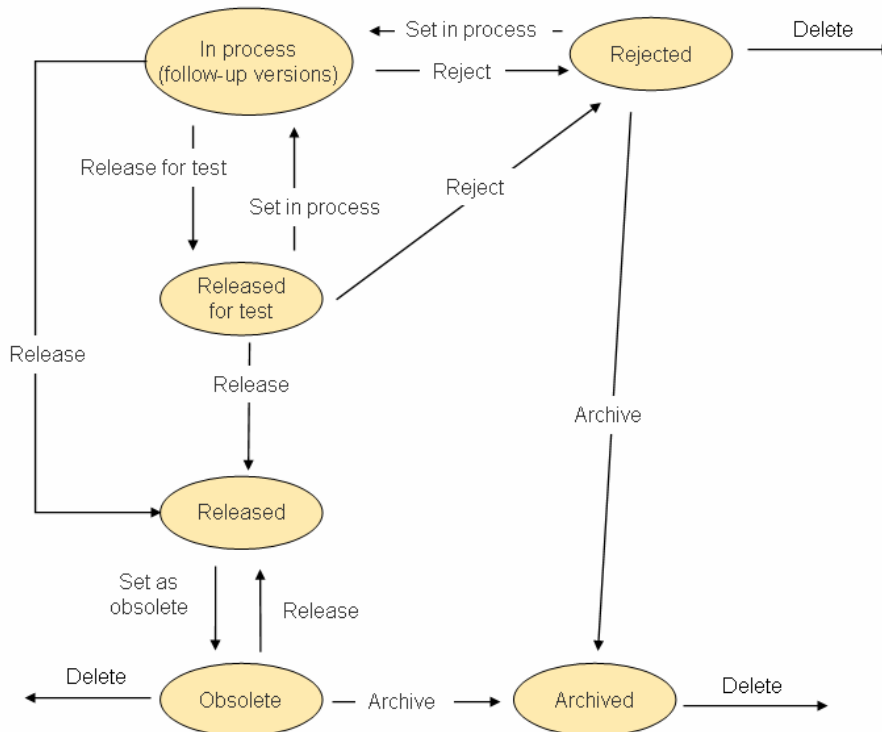
Standard scheme for an SXS version

Figure 3: Standard status scheme for SXS versions (default)

Limitations of the current status management for the regulated environment

If data of a SXS version is applied for the creation of control recipes for PI sheets or external systems the version must have the system status 'released'. The key date for the control recipe creation must lie inside the validity range of the released SXS version. Released SXS versions that fulfill these criteria are called 'valid' SXS versions.

A major requirement for a usage of SXS versions in the regulated environment is traceability of changes. The current system status and business transactions don't fulfill the requirements for the regulated environment:

- The XStep tree assigned to the SXS version could be changed, e. g: by adding new XSteps, changing or deleting assigned parameter etc..

- Obsolete versions could be released again and declared as obsolete many times.

- It is possible to change released versions restrictively: It is not allowed to change the category of an XStep parameter. But the description of the parameter could be changed.

To avoid the loss of traceability and process control in a regulated environment the following requirements shall be fulfilled:

- Establish an approval procedure to release SXS versions for a productive use. Digital signatures during the approval procedure document that only authorized users are able to execute the status change (release the SXS version).

- Released and approved SXS versions shall not be changed anymore. It shall be allowed to copy the released and approved version and to discard the SXS version. Discarding released and approved SXS versions requires the successful execution of digital signatures also.

- If a SXS version is discarded it shall not be possible to use it for production. This means that it shall not be possible to release and approve it again.

- If the content of a released and approved SXS version shall be changed, the user shall use this version as a template and copy it to a new version. To make the changes effective, the new SXS version shall be released and approved. The system checks that only one valid SXS version exist at a key date.

The interaction of business transaction and system statuses for SXS versions requiring an approval procedure is displayed in the graphic below:

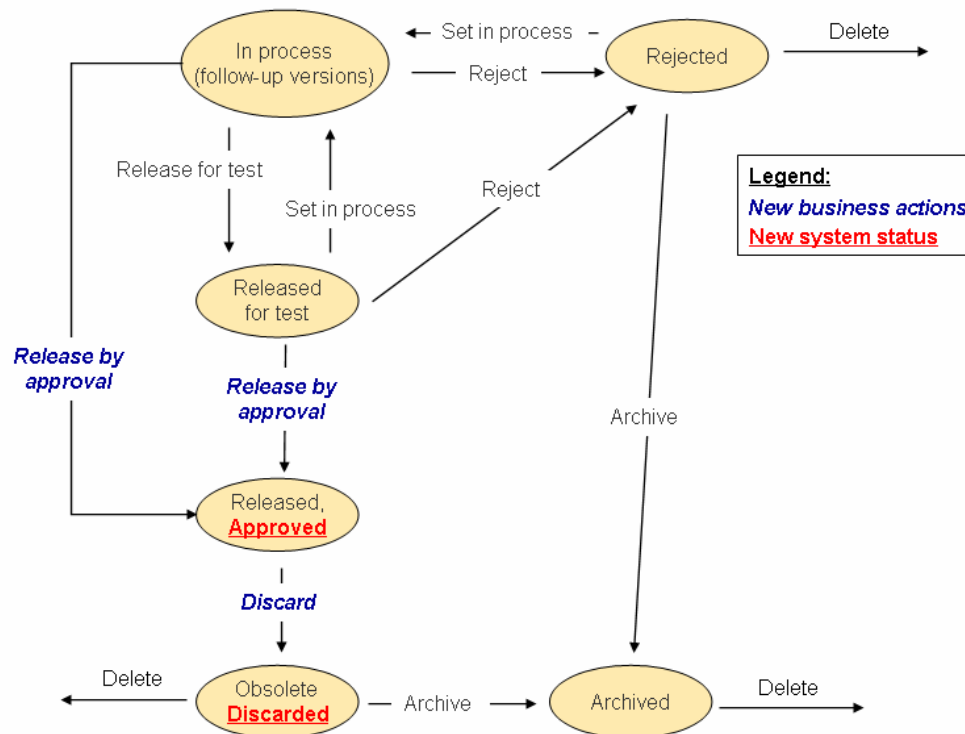## Standard scheme for an SXS version with approval (pharma)

Figure 4: System status scheme for SXS versions that require approval

To support the approval status scheme new business transactions for the approval procedure were created:

- 'Approve':
  - Replaces the business transaction 'Release' (context menu entry for SXS versions with system statuses 'In Process' or 'Released for test').
  - Sets the new system status 'Approved' and the (old) system status 'Released'
- 'Discard':
  - Replaces the business transaction 'Set as obsolete' (context menu entry for SXS versions with system statuses 'Released' & 'Approved').
  - Sets the new system status 'Discarded' and the (old) system status 'Set as obsolete'

New system statuses for the approval procedure were also created:

- 'Approved':
  - Allows only business transactions that don't change a SXS version or the assigned XStep tree. Allows executing the new business action 'discard.
- 'Discarded':
  - Forbids action 'release'.

Usage of digital signatures for approving or discarding SXS versions

During the approval the users shall successfully execute digital signatures to trigger the release or the discarding of a SXS version.

Before SXS versions are released and approved or discarded a digital signature must be executed successfully. Signature strategies are preferred to be used to allow cross-checks by more than one user. Asynchronous signature processes were implemented to enable that several other functions can be carried out in between the individual signatures.

Digital signatures during the approval of a SXS version ensure:

- Only authorized users could approve SXS versions and release them for the productive usage or discard them.

- It is reproducible who approved or discarded the SXS version at which point of time (via the log function of the digital signature tool [transaction DSAL]).

- A cross-check of the SXS version during approval is executed if a signature strategy containing more than one signer is chosen.

- The statuses 'Approved' or 'Discarded' are only set after passing the signing successfully.

Where-used list for referenced SXS

Standard XSteps (SXS) could be used in various master recipes, process orders or inside XStep trees of other SXS. Changes on the content of the SXS therefore could have an impact on many master data object. It is essential that a where-used list for referenced SXS is available for the standard XStep repository.

With the help of the enhancements it is quite easy to search for SXS references: Just mark the SXS and call the context menu (right mouse key):
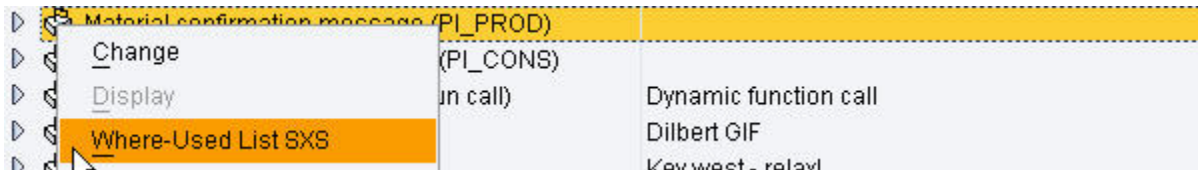

Figure 5: Execute the where-used list search for a marked SXS

If applications have XStep trees that contain SXS reference nodes pointing to the marked SXS, the where-used list will show up and display the results:

Figure 6: Where-used list results

You can create you own layout, hide fields and define sorting. In addition you can add user-specific fields without the need to modify the coding: The result list is based on the class CL_ALV_GRID_XT. This class offers the BAdI ALV_GRID_XT as a customer-exit. For more details please have a look at the class documentation for CL_ALV_GRID_XT (available via transaction SE24).

The where-used list allows you to directly jump into the referencing application. If the SXS is referenced within an XStep tree of another SXS, you can navigate to the SXS reference node by double-clicking on the corresponding where-used list entry.

The where-used list also detects SXS references that were referenced in other SXS references (explosion level > 1). A change of the referenced SXS will also have an impact on the referencing SXS and the applications that carry SXS reference node pointing to the referencing SXS.

Download/Upload of XStep folders, SXS or SXS versions via XML files

An easy way of distributing your master data of the SXS repository is now available: You can down- and upload the complete data of XStep folders, SXS or SXS versions via XML files. Just mark the SXS repository object you want to download and call the context menu:

Figure 7: Download/export an XStep folder

You can store the XML file on you local hard disc or another safe place.

If you want to upload the file into another SXS repository, just mark the SXS object, set it into change mode (via context menu) and call the context menu entry for importing the SXS object:
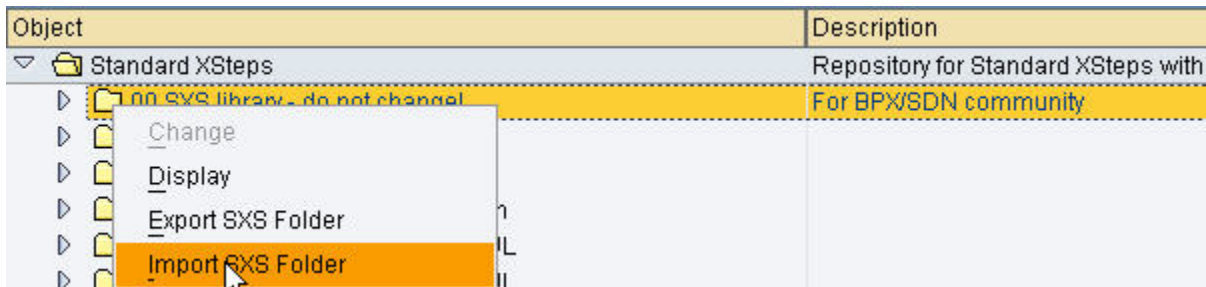
Figure 8: Import/Upload an XStep folder XML file

If you activated the checksum check in the customizing setting before the system would check during the upload if the XML data file was manipulated after download. Depending on your customizing setting the system would:

- Raise an information message and continue with the upload if the XML data file was changed

- Raise a warning message and continue with the upload if the XML data file was changed

- Raise an error message and abort the upload if the XML data file was manipulated

- Upload the data file without raising any message
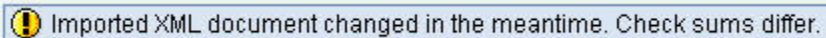
Figure 9: Warning status message caused by detected XML file manipulation

During upload the system also checks for completeness of the data included in the XML file. An upload will be terminated if not all required information is included in the XML file. This could occur if XStep trees of exported SXS repository objects have SXS reference nodes, but the data of the referenced SXS wasn't downloaded together with the other object data. Then an upload would result in inconsistent data. If so the system will also abort the upload and raise an error message.

Automatic numbering for SXS versions

Activating this enhancement leads to an automatic numbering of the SXS versions. The first version gets the number '0001', the second version you create will be labeled as '0002' and so on.

Automatic calculation of the valid-to date for released SXS versions

Standard XSteps could have more than one version that is valid within different time intervals. To avoid confusion and allow tracking and tracing of changes, the system prevents that two released versions can be used productively at the same key date. The automatic calculation of the valid-to date during SXS version release provides an automatic adjustment of the validity ranges of the released SXS versions:

Figure 10: Different SXS version for a standard XStep

During release of an SXS version the system detects if an overlap of the validity ranges of released SXS versions would occur. If necessary it calculates the new valid-to dates of the affected SXS versions:
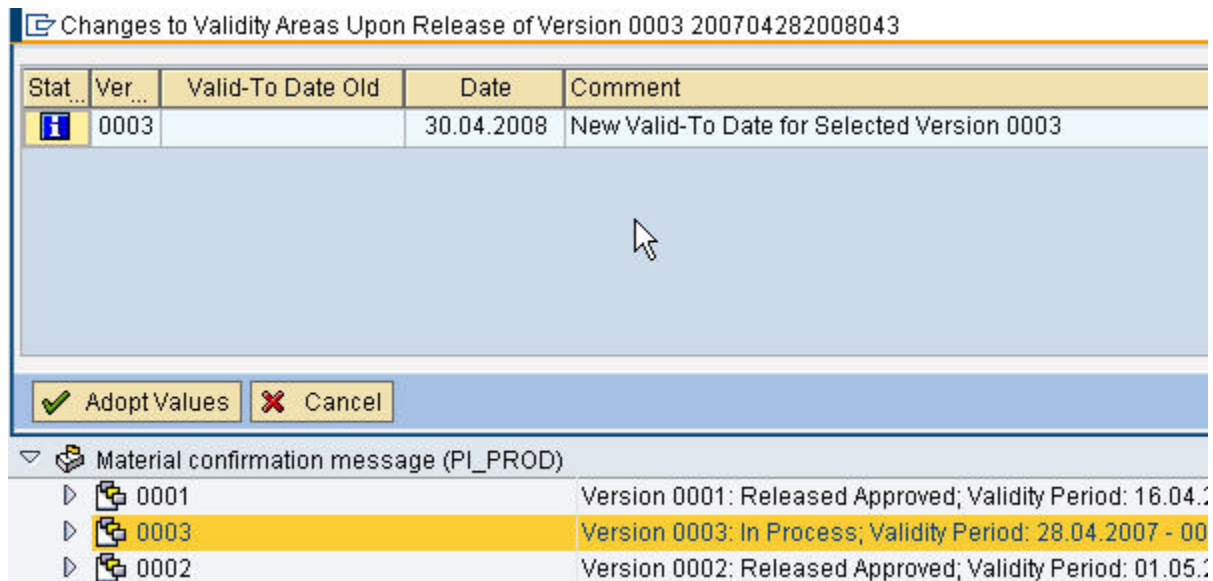
Figure 11: Adjustment of valid-to date for a released SXS version

How does the system determine the valid-to dates of released versions?

The automatic calculation of the valid to date during the release of SXS versions is activated via the new SXS customizing by setting the corresponding flag. The system reacts as follows:

- While creating new SXS versions it is only possible to enter the valid from date of a version. The valid to date field is inactive and grayed out. During release/release for test the system doesn't accept a valid from date for the current version in the past (earlier than the current date).

- The field to maintain the valid from date of the current version (to be released/released for test) is set inactive after getting the status 'released for test', 'released' or 'obsolete'. To change the valid from date of this version it is necessary to set the status 'in process', if the active system status of the version allow it to do so.

- If a version is released or released for test (date from date should be equal or greater than the current key date) the system checks if other versions exist that were released within the validity interval. The system then proceeds as follows:

  If the old released version begins before the new version (to be released) begins, the end date to of the old version is set to one day before the new version is released:

Date from (New version [NV]) > Date from (Old version [OV])

Date to (NV) = Date to (OV) =>
Date to (OV) = Date from (NV) - 1

Date to (NV) < Date to (OV) =>
Date to (OV) = Date from (NV) - 1

Date to (NV) > Date to (OV) and
Date to (NV) > Date from (OV)=>
No changes necessary

Date to (NV) > Date to (OV) =>
Date to (OV) = Date from (NV) - 1
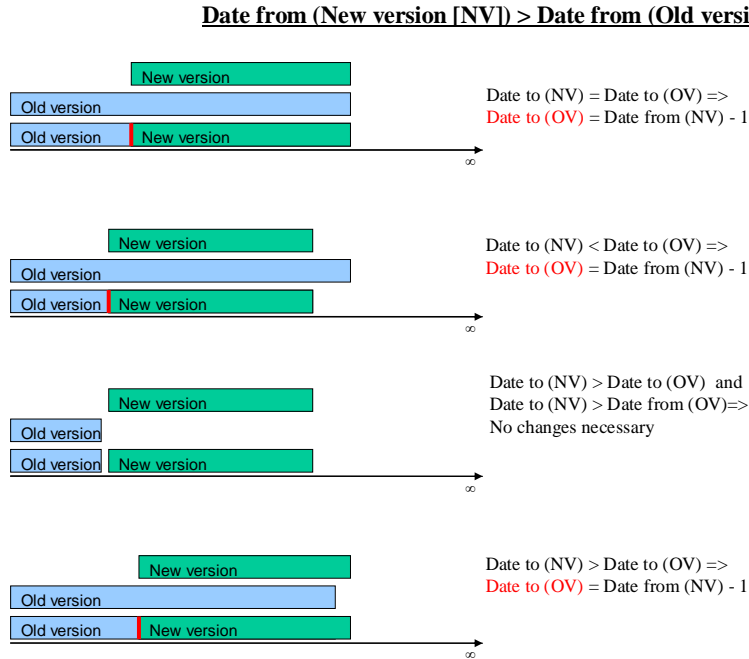
Figure 12: Calculation of new end date for released versions

If the new version (to be released) begins before the old released version begins, the end date of the new version is set to one day before the old version was previously released:

Date from (New version [NV]) < Date from (Old version [OV])

Date to (NV) = Date to (OV) =>
Date to (NV) = Date from (OV) - 1

Date to (NV) < Date to (OV) =>
Date to (NV) = Date from (OV) - 1

Date to (NV) < Date to (OV) and
Date to (NV) < Date from (OV)=>
No changes necessary

Date to (NV) > Date to (OV)
Date to (NV) = Date from (OV) - 1
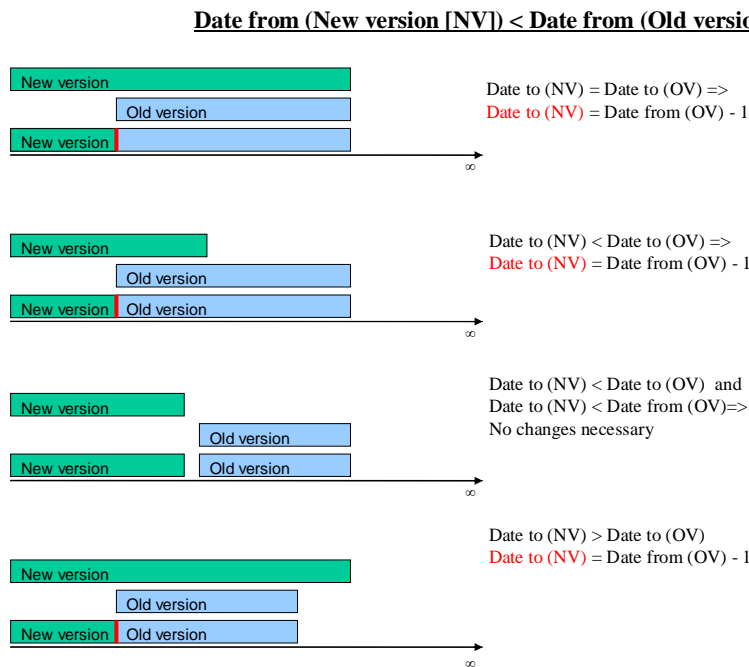
Figure 13: Calculation of a new end date for the current version

If both versions have identical start dates no automatic calculation of the validity intervals is executed. An error message is raised:

Date from (New version [NV]) = Date from (Old version [OV])

New version

Old version

No changes on valid from dates => error

∞
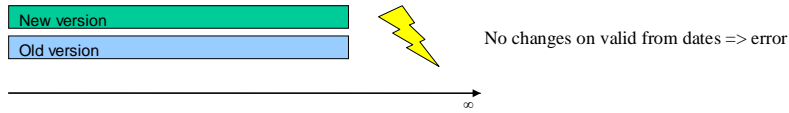
Figure 14: No calculation for versions with identical start dates

The calculation logic allows it to close gaps between old versions that were release before:

Intersection of new versions

New version

Old version

Old version

Changed: Valid to date old version

Changed: Valid to date new version

Old ver   New version   Old version

∞

∞
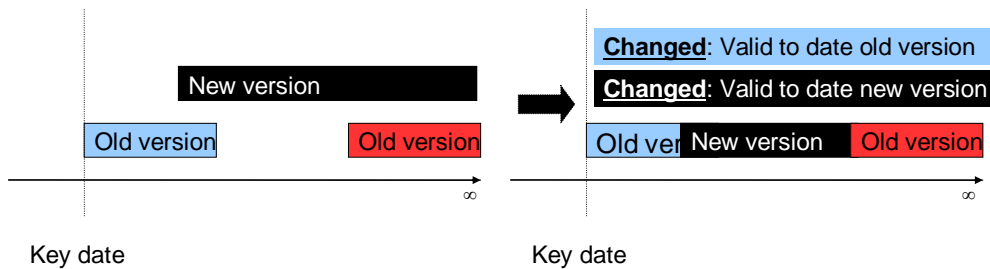
Key date

Key date

Figure 15: Example for the automatic calculation of validity dates for a multiple SXS version scenario

To avoid traceability conflicts it shall not be possible to release new versions with a key date earlier than the current date. The system shall also stop calculating the validity intervals during release if the new and an old released SXS version have identical valid from dates:

Restrictions

No calculation if valid from date of the new version < current date

New version

Old version

No calculation if both versions have identical valid from dates

New version

Old version

∞

∞

Key date

Key date
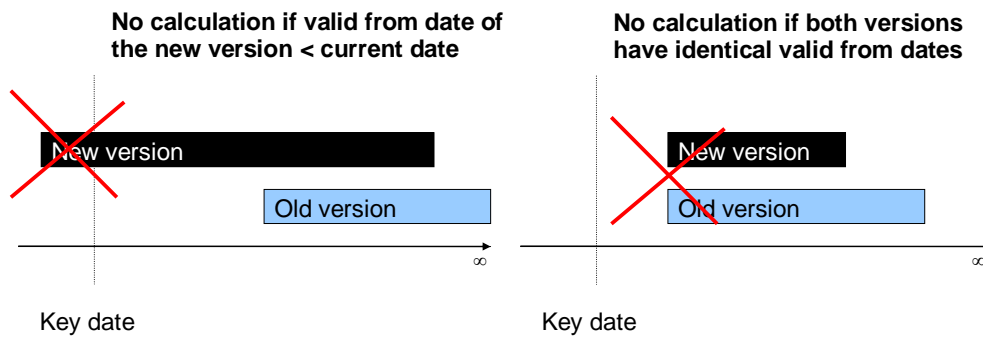
Figure 16: Restrictions for the automatic calculation of validity dates for SXS versions

Customizing settings

The new customizing transaction CMX21 allows you to 'fine-tune' your SXS repositories on a plant level:
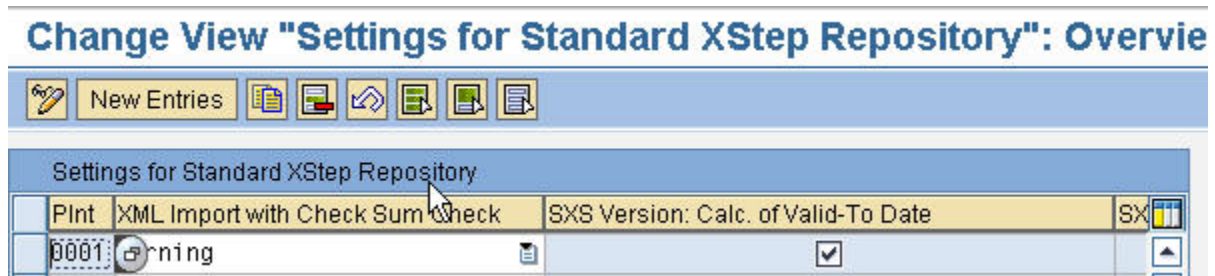
Figure 17: Customizing settings for the enhancements

You can activate:

- the message type for the message that appears if a manipulated XML file is uploaded

- the automatic calculation of the valid-to date for SXS versions

- the automatic numbering of SXS versions

- the approval status scheme and assign signature strategies for approving or discarding SXS versions

Realization of the enhancements

SAP R/3 Enterprise 4.70 Extension Set 2.00

A customer-specific solution (subject to fee) exists that covers most of the above mentioned functionality.

SAP ERP ECC 5.00 (ERP 2004) & ERP ECC 6.00 (ERP 2005)

A modification of the SAP standard is available. It is activated after implementation of the pilot note 948513. As a prerequisite the support package level of the ERP system must at least be higher than SAPKGPAC13 (ECC 5.00, ERP 2004) or SAPKGPAD05 (ECC 6.00, ERP 2005). It is recommended to implement the actual available support package as a basis for the ERP system.

SAP ERP ECC 6.00 Enhancement Package 2 (ERP 2005 EhP 2)

The enhancements become SAP standard. They will be activated after implementation of the enhancement package if the corresponding business function is switched on.

Related Content

Here you'll find additional information about similar topics.

- [Improvements of SAP ERP Applications for Process Industry (PP-PI*) Delivered with ERP 2004 and ERP 2005](#)

- [Improvements of SAP ERP Batch Functionality (LO-BM*) Delivered with ERP 2004 and ERP 2005](#)

- [Digital signatures for SAP applications using the Digital Signature Tool – Developer Guideline and best practices](#)

Copyright