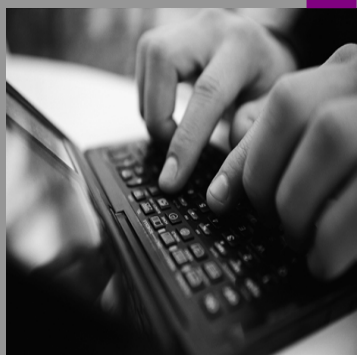


How-to Guide
SAP NetWeaver '04

How To... Transport EP 6.0 Content

Version 1.00 – April 2005

Applicable Releases:
SAP NetWeaver '04
Enterprise Portal 6.0 SP9 +

Table of Content

Scenario **2**

1 Introduction **2**

2 Related information **2**

3 Process Overview **3**

3.1 Process Definition 3
3.2 Configuration of transports 4
3.3 Transport Guidelines 6
3.3.1 Do one export and two imports! 6
3.3.2 Apply changes to originals only, not to copies! 8
3.4 Summary: Recommended Steps during content transport 9

4 Exporting Content **9**

4.1 Definition of Transport Packages 9
4.1.1 Customizing of Administration Roles 9
4.1.2 Design Rules for transport packages 10
4.2 Manual Export 15
4.3 Export Reporting 15
4.4 Summary: Exporting Portal Content 16

5 Import into Quality Assurance System **17**

5.1 Manual Import 17
5.2 Customized Import 19
5.2.1 SDM as central deployment tool 19
5.2.2 Importing content using SAP NWDS and NWDI 22
5.3 Import Reporting 30
5.3.1 Transport Log Files 30
5.3.2 Software Change Tracking component of SMD 30
5.4 Summary: Import into Quality Assurance System 31

6 Testing in Quality Assurance System **32**

7 Importing into Production System **33**

7.1 Import Procedure 33
7.2 Deletion of PCD objects in PRD system 33

Scenario

Customers need to transport EP 6.0 content between their 3-tier system landscape, which consists of a development system (DEV), quality assurance system (QA) and production system (PRD).

1 Introduction

The focus of this How To Guide is on designing a process to efficiently transport EP 6.0 content across system boundaries. It explains the use of the export and import functionality of EP 6.0 on Web AS 6.40.

Designing a transport process for EP 6.0 content generally can follow three different approaches:

- Transport content using standard export and import functionality
- Transport content using standard export and import functionality and enhancing it with custom developments
- Transport content using SAP NetWeaver Development Studio and SAP NetWeaver Development infrastructures

Disclaimer:

Transporting EP 6.0 on 6.40 content will be significantly improved with the next SAP NetWeaver releases, which will closely integrate content transports into the SAP Java Software logistics.

Proposals for transporting EP 6.0 content therefore can only provide temporary workarounds for the current release of SAP NetWeaver 04. It might be necessary to rework and redesign the transport process when upgrading or changing to further releases of SAP NetWeaver in the future.

The SAP NetWeaver Developer Studio (NWDS) and the SAP NetWeaver Development Infrastructure (NWDI) can be used to transport portal content from the SP Stack 11 shipment of SAP NetWeaver 04 onwards. The solution described in section 5.2.2, “Importing Content Using SAP NWDS and NWDI”, is a project solution and requires slight modification to the SAP NWDS settings. Due to its origin as a non-standard solution, there is no official support through SAP OSS.

2 Related Information

Additional information regarding transports of EP 6.0 content can be found at the following locations:

SAPNet:

http://help.sap.com > SAP NetWeaver > Portal > System Administration > Transporting Portal Content

SAP Service Marketplace:

http://service.sap.com/ep60howtoguides >
HowTo Use Business Packages in Enterprise Portal 6.0 (Kap. 5.3.5 Transport of Business Packages)

SDN:

Search for “Change Management and Transport in the Enterprise Portal”

https://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/com.sap.km.cm.docs/documents/a1-8-4/Change%20Management%20and%20Transport%20in%20the%20Enterprise%20Portal.evn

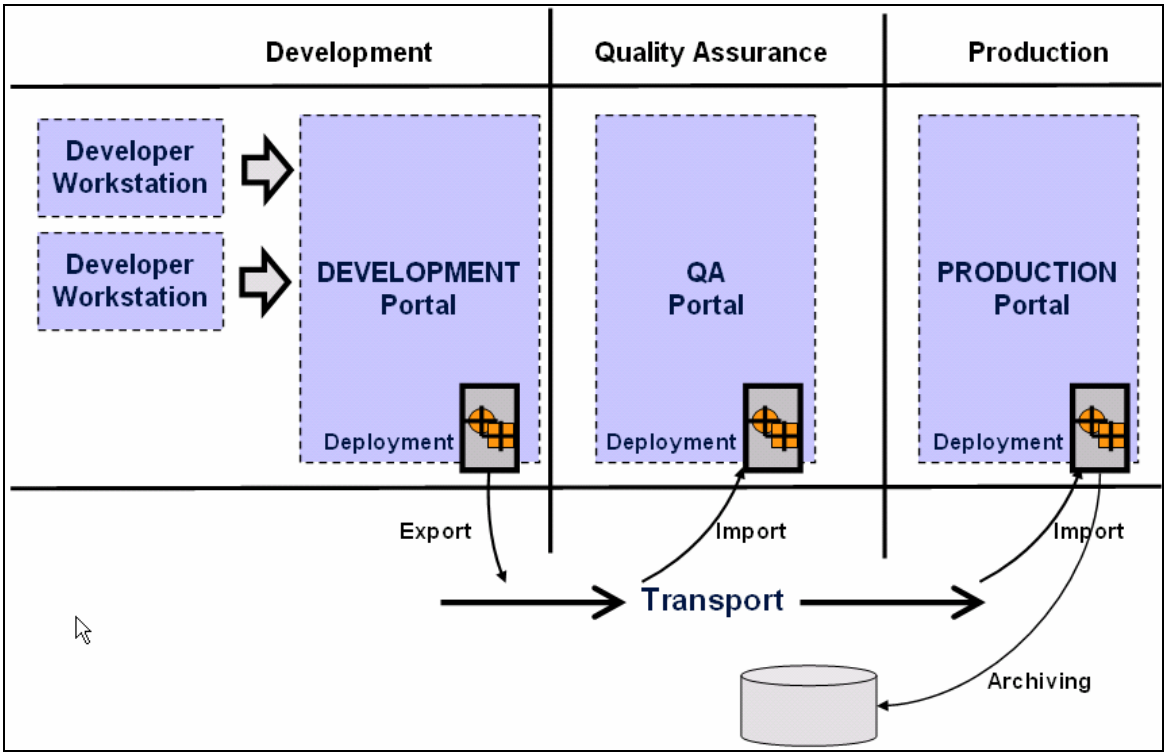
SDM Documentation included on installation CDs for Web AS 6.40:

By default, additional information about SDM can be found at C:\usr\sap\<sid>\<instance>\SDM\program\doc

- SDM_Documentation_630_EN.pdf
- SDMCommandLineDoc630_en_final.pdf

3 Process Overview

3.1 Process Definition



In general, content is initially developed and created in the development environment of SAP Enterprise Portal. Customer-specific development components (new iViews, layouts, etc.) must also be deployed here.

New content, such as roles, worksets, pages, iViews, layouts or other Portal Content Directory (PCD) objects, is also generated in the development environment. Adding Content can be added to the development environment in different ways:

- Content is newly created based on components (SAP standard or customer-specific components)

- Content, such as content of SAP Business Packages, is imported from the iViewStudio/SDN
  (Working with SAP business packages is described in detail in the How To Guide: "HowTo Use Business Packages in SAP Enterprise Portal 6.0" > see chapter 3) "Related Information").

To ensure functional correctness and compatibility of new software packages with the customer-specific environment, tests need to be carried out and planned carefully. After testing in the development environment, the relevant content for export is bundled in transport packages to be exported to the file system and thus prepared for import into the QA system.

The import of content into the QA environment needs to be organized by a procedure. The definition of system administrators for import, order of import packages, storage location of package files, and reporting about import success must be part of a process description. This guide discusses tools and methods to support such a process.

You should carefully document your testing procedure in the QA system to verify and approve that the import into the PRD environment can be started. After testing, EP 6.0 content must either be reworked and or deployed to the productive environment.

When importing content into the production system, the integrity and readiness of content must have been certified. Transport packages for the PRD system should include end-user relevant information only. Note that the current release of EP 6.0 on 6.40 does not offer preview or rollback functions for incorrect content imports. Any defective PCD object can be fixed by overwriting it with a corrected object version during a new import. Objects must be manually deleted in the PRD environment; this must be organized carefully.

3.2 Configuration of Transports

The configuration of the export and import is defined centrally in the PCDStartup.properties file. The table below lists default settings as described in the documentation at http://help.sap.com > SAP NetWeaver 04 > Portal > System Administration.

Pcd.TransportApplication. ExportRootDir	Storage directory for the export files	Directory is generated automatically based on the entries during portal installation.
Pcd.TransportApplication. ImportRootDir	Storage directory for the import files	Directory is generated automatically based on the entries during portal installation.
Pcd.TransportApplication. TempDir	Storage directory for the temporary files	Directory is generated automatically based on the entries during portal installation.
Pcd.TransportApplication. CleanTempDir	Causes a cleanup for temporary files	Default value is „true". **Do not change this value.**

Pcd.TransportApplication.Export.ExcludeSystemObjects	Defines whether or not the objects in the SAP namespace initially delivered by SAP are excluded from the export.	Default value is „true“. **Make sure that this value is not changed.**
Pcd.TransportApplication.Export.ExcludeObjectTypes	Here you can decide which object types are excluded from the export.	There is no default value. For example you can configure your transport scenario so that PAR files are excluded from the export. See also Packaging of Portal Transport Archives.
Pcd.TransportApplication.Export.ExcludeObjectTypes AfterRuleProcessing	Here too you can decide which object types are excluded from the export. You can also define that dependent objects should also be exported.	There is no default value. For example you can exclude PAR files from the export, but transport the corresponding resource bundles. See also Packaging of Portal Transport Archives.
Pcd.TransportApplication.ProtectedNamespaces	The protected namespaces of the objects initially shipped by SAP that are to be excluded from the export and import are entered here.	The following are entered as default values: com.sap.portal, com.sap.km, com.sap.netweaver. **Do not delete these entries.**
Pcd.TransportApplication.ProtectedUrlPatterns.Default	Here you find the object paths in the Portal Catalog defined by SAP that are excluded from the export.	The default value is the path to the portal themes: pcd:portal_content/themes/sap_*. This parameter is not intended for your own entries.
Pcd.TransportApplication.ProtectedUrlPatterns.Custom	Here you define the paths in the Portal Catalog that should be excluded from the export.	There is no default value. You enter the object paths to be excluded from the export.
Pcd.TransportApplication.StatusDetailLevel	Defines the degree of detail of the information in the log files. Possible values: ALL, Error and None.	The default value is ALL.
Pcd.TransportApplication.ProcessReportDir	Storage directory for the log files.	Directory is generated automatically based on the entries during portal installation.
Pcd.TransportApplication.ProcessReportCleanupInterval	Here you enter a number that defines the number of days in which the log files should be cleaned up.	The default value is 2.

The current values can be viewed in the portal by navigating to System Administration > Support > Portal Content Directory > PCD Configuration:

Parameter für Transport Application	
✔ Pcd.TransportApplication.ExportRootDir	C:\usr\sap\J2E\SYS\global\pcd\Export
✔ Pcd.TransportApplication.ImportRootDir	C:\usr\sap\J2E\SYS\global\pcd\Import
✔ Pcd.TransportApplication.TempDir	C:\usr\sap\J2E\JC03\j2ee\temp\pcd\transport
✔ Pcd.TransportApplication.CleanTempDir	true
✔ Pcd.TransportApplication.Export.ExcludeSystemObjects	true
✔ Pcd.TransportApplication.Export.ExcludeObjectTypes	<empty>
✔ Pcd.TransportApplication.Export.ExcludeObjectTypesAfterRuleProcessing	<empty>
✔ Pcd.TransportApplication.ProtectedNamespaces	com.sap.portal, com.sap.km, com.sap.netweaver.bc
✔ Pcd.TransportApplication.ProtectedUrlPatterns.Default	pcd:portal_content/themes/sap_*
✔ Pcd.TransportApplication.ProtectedUrlPatterns.Custom	<empty>
✔ Pcd.TransportApplication.StatusDetailLevel	All
✔ Pcd.TransportApplication.ProcessReportDir	C:\usr\sap\J2E\JC03\j2ee\temp\pcd\transport\reports
✔ Pcd.TransportApplication.ProcessReportCleanupInterval	2

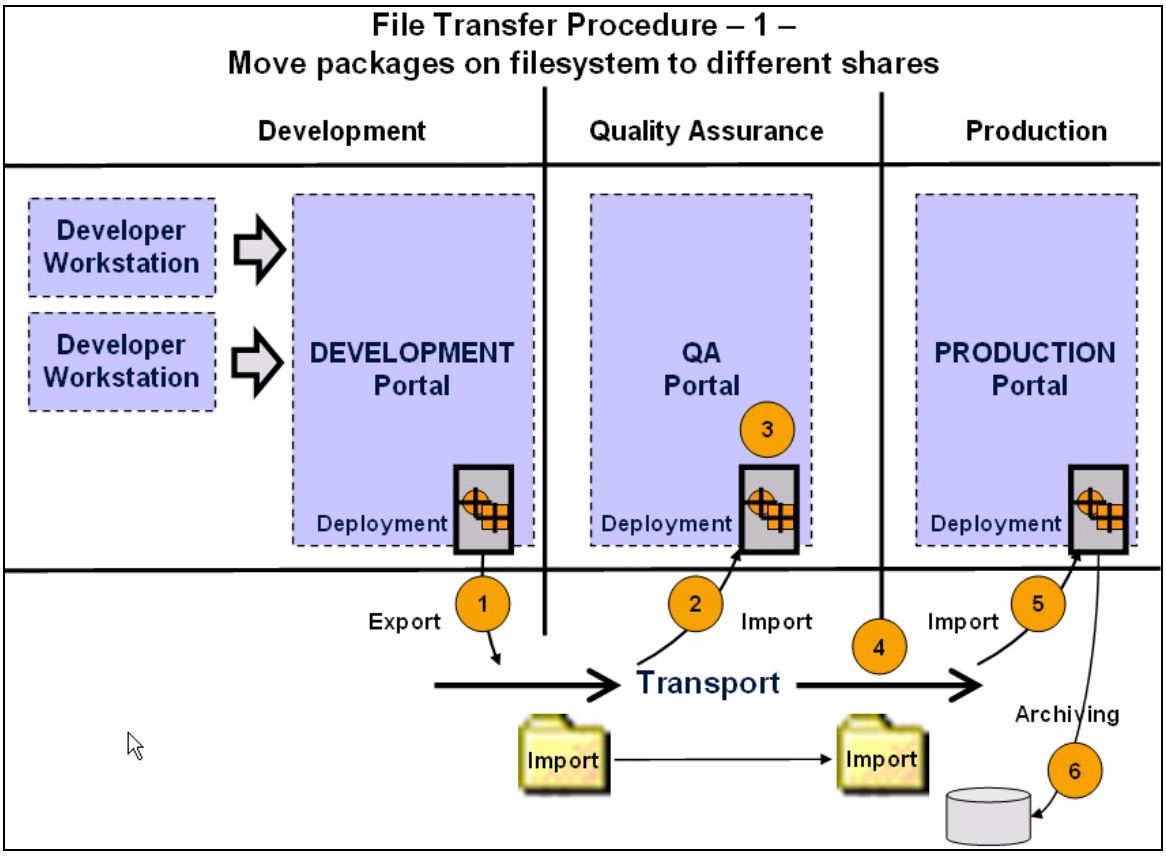
3.3 Transport Guidelines

Simple rules help you to design an efficient transport process across system boundaries:

- Perform one export and two imports.
- Apply changes to originals only, not to copies.

3.3.1 Perform one export and two imports

Transport packages must be exported and imported to various file system locations from a predefined path.



1. After creation, export the transport package to a predefined share on the file system. The PCD property Pcd.TransportApplication.ExportRootDir = “path on file system” can be configured.

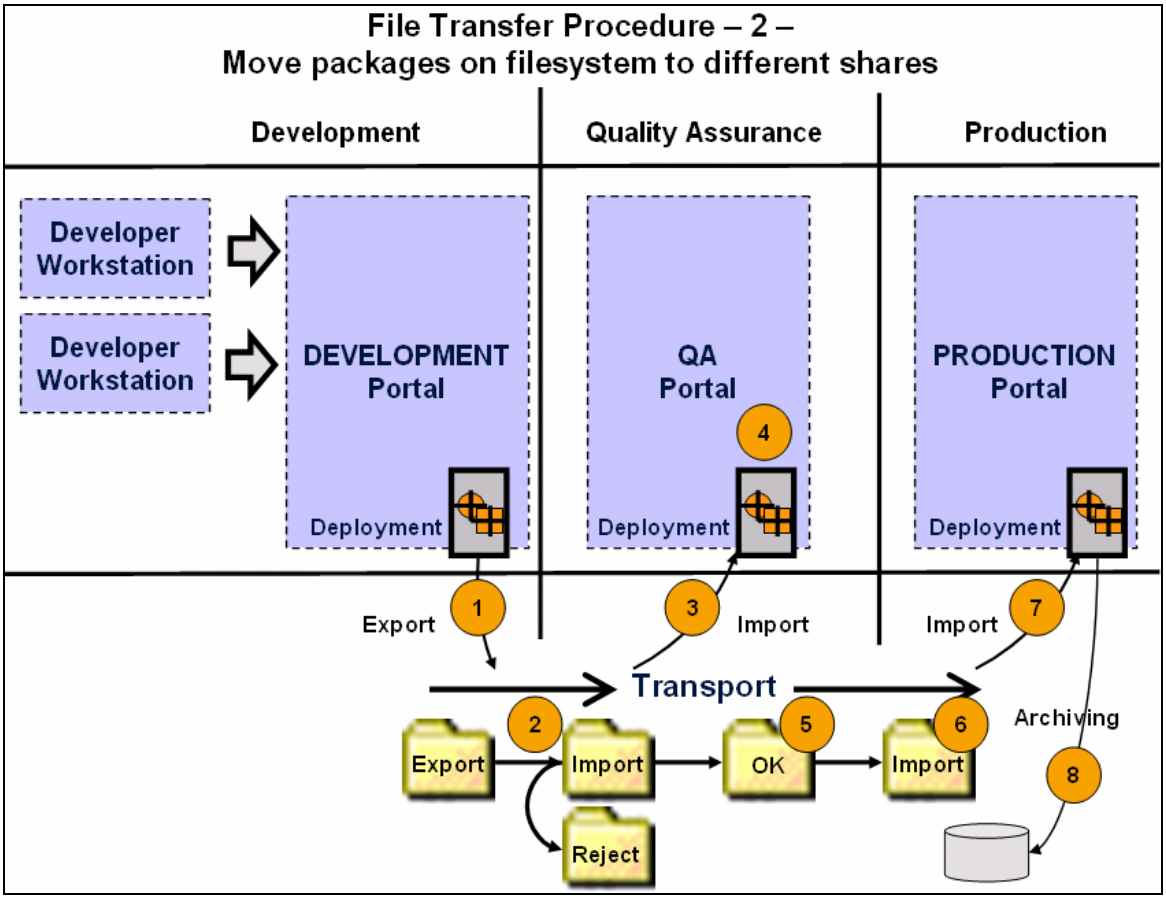
2. To import packages from a share, mount the import directory on the export share, making all export packages available for import. Alternatively, the PCD property Pcd.TransportApplication.ImportRootDir = “path in file system" can point to the specified share for Pcd.TransportApplication.ExportRootDir.

3. After an import, test the imported content.

4. Once the import has been finished, move the imported package from the shared directory of the QA system to another predefined share. Instead of moving the files, you can also mount the PRD import share to point to the QA import share.

5. From this predefined share, start the import into the production environment.

6. After successfully importing content into the production system, move the package files from the production import share to an archiving directory.



1. After creation, export the transport package to a predefined and configurable share on the file system (see above).

2. To import packages, move the packages from the export share to a predefined import share of the QA environment (see above: customizing PCD property Pcd.TransportApplication.ImportRootDir = “path in file system").

3. Perform the import into the QA environment from the QA import share.

4. After an import, test the imported content.

5. Once the import has finished successfully, move the imported files from the QA import share to another predefined share for packages, waiting for approval.

6. If a package test has been approved, move the import packages to the PRD import directory.

7. From the PRD import share, import the packages into the production system.

8. After a successful import, archive the packages.

3.3.2 Apply changes to originals only, not to copies

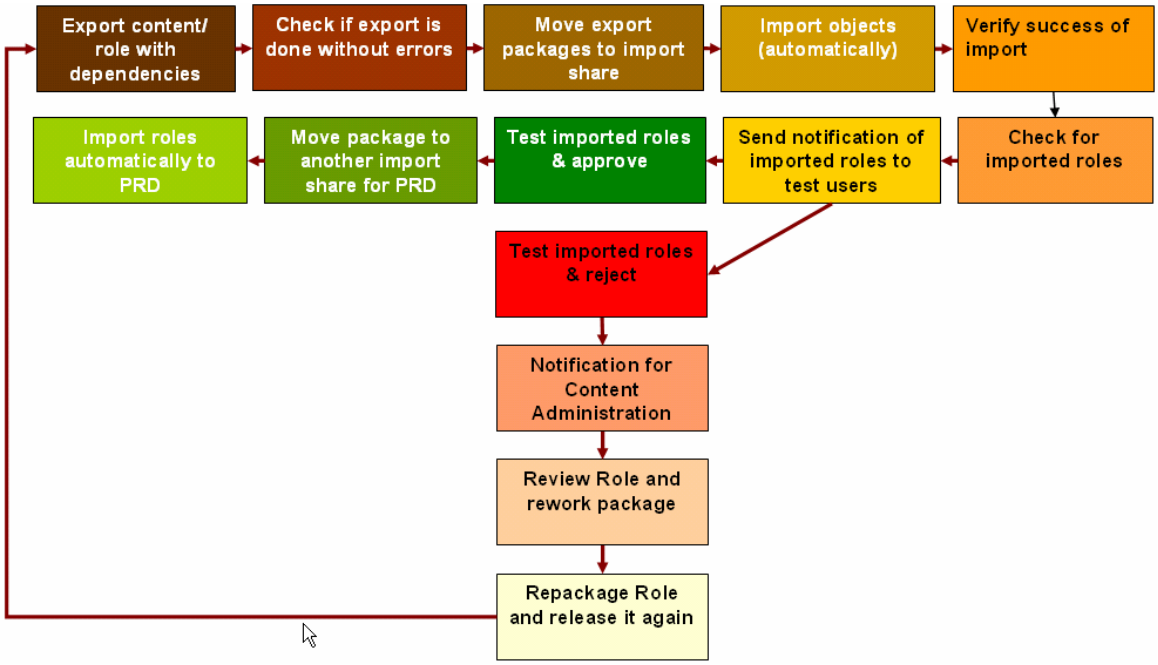
If errors were found during export or import, the author must rework the faulty transport package and export it again. Remove the package from the export share to prevent its transfer to another import share.

The author of a transport package can be identified either by the metadata “author” in the corresponding log file of each package or by namespaces that are applied to each transport package during creation.

After package export, check the log file (see chapter 4.3 Export Reporting) to verify that the transport package was properly created. If no errors were reported in the export log file for the package, the epa-file can be moved to another import share.

If a faulty package is found during the import, you should exclude the package from further transfers to the next transport share in order to prevent another import. The author of the damaged file must rework and repair the package. For error analysis during import, see section 5.3 Import Reporting.

3.4 Summary: Recommended Steps During Content Transport



4 Exporting Content

A package is a transportable archive containing the portal objects that can be exported or imported. All objects of the Portal Content Directory (PCD) can be exported and imported (for example roles, worksets, pages, iViews).

4.1 Definition of Transport Packages

4.1.1 Customizing of Administration Roles

By default, creation of a transport package is part of the system administration role. The creation and maintenance of a transport package, however is often regarded as part of the content administration role. A common business scenario is that content developers are responsible for deriving EP content based on custom-specific development. To implement this kind of scenario, administration roles in EP 6.0 can be customized according to the business needs.

The following setup can be used to enable content administrators to export own content:

- Assign users to the content administrator role
- Create a new role and assign the system administrator role (ID = pcd:portal_content/administrator/system_admin/system_admin_role) as delta link
- Customize the new role to contain nothing but the export iView
- Adjust the security zone settings according to the role design: Add the new role or its assigned users to the targeted security zone to enable end-users to access this iView.
    - ACLs on security zones define access to components.
    - Note that ACLs for an iView need to correspond to the ACL of the security zone from which the iView was derived. If the export-related iViews will be accessed by a request user, that user must also be added to the corresponding security zones.
    - For detailed information about permission and security zone settings, see the SAP Service Marketplace at alias /nw-howtoguides > Portal, KM and Collaboration > “How to use Security Zones in NW04 SPS09” and “Configuring Permissions for Initial Content in SAP EP 6.0”.
- Adjust the ACLs of the Portal Catalog to suit the content administrator and security requirements:
    - Content administrators only have access to dedicated content areas. This is implemented by ACLs on certain parts of the Portal Catalog tree.
    - To create a new package, you need read/write access for the location where the object will be located.
    - To include an object in a transport package, you need at least read access.

4.1.2 Design Rules for Transport Packages

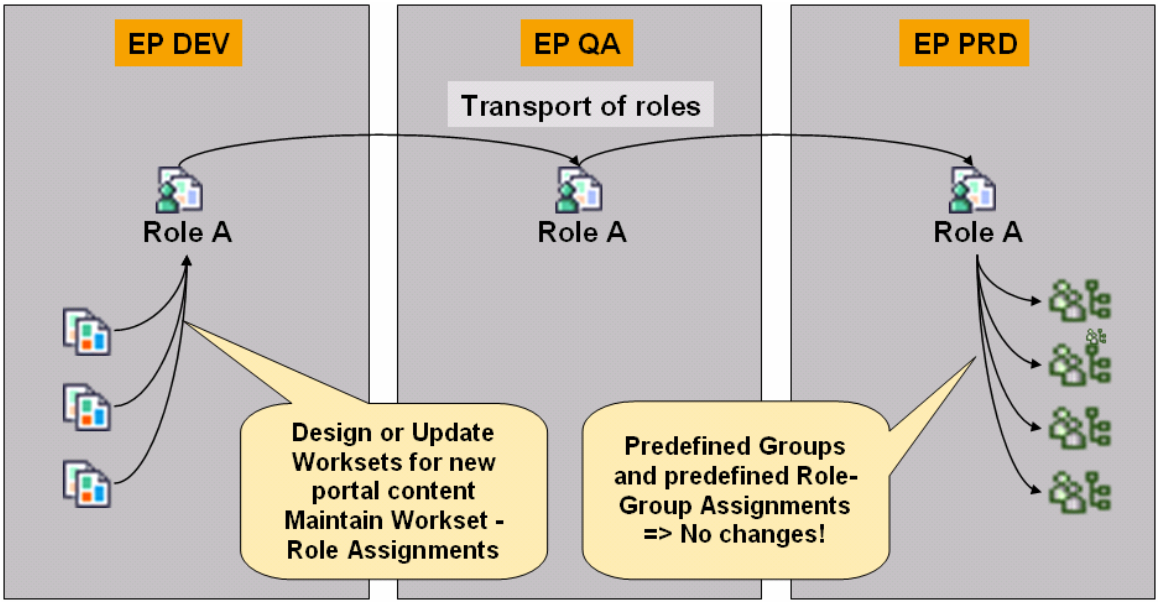
When designing transport packages, you should keep the following aspects in mind:

- Only transport end-user relevant information
- Create small and clearly-defined packages
- Define naming conventions for transport packages

4.1.2.1 Transport End-user Relevant Information

Information is primarily provided to end users with roles. To transport end user relevant information between DEV – QA – PROD, the design of a transport package must be aligned with the role maintenance process.

An example of a process for role design and publishing is:



Role Creation

Roles are initially created in the development system (see 3.1 Process: “Content is added to the development environment in different ways...”). The role is designed to be a container for various “information clusters”, bundled by assigned worksets or objects, such as pages and iViews. Roles can either be transported into the QA and PRD system together with their dependent objects or separately from their linked content.

Transporting content means exporting objects from one system and importing them into another target system: By importing PCD objects you

- Add new objects to a system and /or …
- Replace and overwrite existing objects
- Update existing relationships between an overwritten object and other objects (e.g. overwriting an iView that is linked to a page means updating and refreshing the related page information as well).
- Create the PCD structure of the source system analogously in the target system.

If an object is transported without its links, this can result in missing or broken delta links in the target system. You can therefore transport a new role together with its dependent objects to ensure consistency of the role object in the target system.

As ACLs and role user assignments usually vary in the different systems according to another user persistent store of the DEV, QA and PRD systems, role user assignments and ACLs need to be maintained once in each import target system:

- Roles can be assigned to predefined test user accounts in the QA environment
- Roles can be assigned to the respective end users in the PRD system

Updating roles or dependent objects overwrites existing PCD objects, but does not affect related role-user assignments or ACLs. If role-user assignments were created in the target system, they are neither deleted nor updated by content transports.

Role Updates

Changes to existing roles should be applied by updating the linked objects, e.g. deleting or updating a linked workset, page or iView. Two transport scenarios are possible:

1. Only changed objects are packaged into a transport package and exported.

2. The role object and all dependent objects are bundled in a transport package and exported.

1. Transporting only changed objects means:
   - The transport content is less than for a role with all its dependent objects. Thus, the duration and load caused by an import is significantly reduced. (See previous chapter: Duration of import and export grows according to the number of objects included in a transport package due to the import net time of objects and updates of relationships to other content objects.)
   - In case of troubleshooting, errors can be isolated and faulty packages identified more easily.

2. Transporting the role with all its dependent content means:
   - The number of objects in the transport package might be large.
   - You are therefore recommended not to include more than one role in the transport package in order to limit the overall number of objects to be transported.
   - The duration of the import grows according to the number of objects: All content is deployed, which means that dependent delta links are also updated.

Note: Transport of System Objects

System objects are created in the development environment and usually correspond to R/3 development systems. An alias that can be referenced by any iView, such as a transaction iView, is maintained for all system objects.

If a DEV system object and its dependent objects is exported, the system alias is not transported: A system object must be reworked in the QA environment to establish a connection to its corresponding QA R/3 system instead of still pointing to the DEV environment. You must recreate the alias definition for the iView.

IViews such as transaction iViews reference the alias definition of a system object. The alias reference is part of the iView definition (property system alias) and is therefore transported. In order to ensure execution of a transaction iView, the referenced alias must exist in the target system. When maintaining a system object definition, you must recreate identical aliases of the source system.

Rollout of New Templates

Special attention should be given to template objects that can be commonly used by administrators of different areas. To rollout global templates, create separate transport packages that only transport this “shared content” once to the target system. To avoid broken delta links in target system, you need to ensure that common templates are rolled out prior to content that references new templates. To protect a template from being used by unauthorized users, use ACLs. For example, after a successful import into the target system, you can for example release a global template by copying it to the global share to which other administrators have read access.

4.1.2.2 Create Small and Clearly-Defined Packages

Design goals for transport packages are to:

- Limit the number of objects included in a transport package (see previous paragraph)
- Limit potential content overlaps among transport packages

Using the Include Filter

The include option can be used

- When new content, such as roles, worksets or pages, is to be released
- When updates are applied to several components, and not only to a single one
- When the author of a transport package cannot be sure if dependent objects have been changed
- When inheritance and delta link chains in the target system might be inconsistent, e.g. if a corrupted object was imported.

A reasonable approach to transporting complete units of content like roles is to include dependent objects. But if the number of transport objects increases to more than 250-300 objects, the duration of the import and the system load might affect end user performance.

You should carefully plan the import of transport packages to prevent overlaps between the transport packages. The same object might be included in various packages, but with different versions. The import order of the packages is important to prevent older content from overwriting younger content. Enforcing imports by time stamp order will be explained in Section 5.2 Customized Import.

Using the Namespace Filter

The namespace filter of the export iView can be used to control the overall number of objects. The namespace filter only includes those objects that belong to the given namespace. When using the include option during transport package creation, authors should use the namespace of objects that belong to their area of responsibility.

Example:

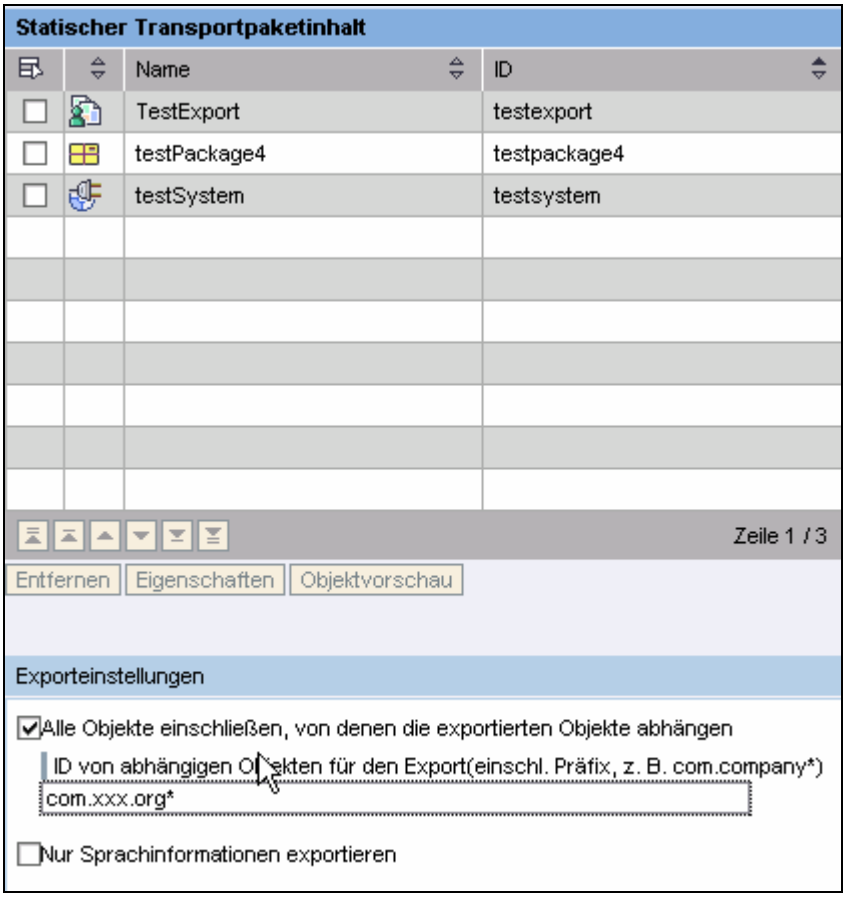
A content administrator structures his Portal Catalog branch by folders, and sorts all objects by predefined naming conventions. The naming conventions must define clear areas of responsibility for portal content.

- First level: <global company content> | <public content>
  e.g. **com.sap.**
- Second level: <content of a certain department, e.g. HR>
  e.g. **com.sap.hr.**
- Third level: <content of an administrator or project, e.g. user>
  e.g. **com.sap.hr.<user>**

- Fourth level: <content sorted by object types; e.g. roles, pages, iViews>
  e.g. **com.sap.hr.<user>.roles**

Objects should “inherit” the namespaces of their parent folders so that they can be easily located in the Portal Catalog tree and their origin indicated. To filter transport package objects during the export, content administrators should use the right namespace to include only those objects that they created or changed.

Example:
An administrator created a role, and defines a transport package, including the role and all its dependent objects. He includes only objects with namespace **com.sap.hr.<user>*.** Content sources like common templates belonging to namespace **com.sap.hr*** are not included and therefore must be imported into the target system in advance.



Note:

- In general, dependent content can only be included in a business package if the request user has at least read access.

- The transport property Pcd.TransportApplication.Export.ExcludeSystemObjects defines whether or not the objects in the SAP namespace initially delivered by SAP are excluded from the export. Default value is „true“. **Make sure that this value is not changed.** PCD objects in the SAP namespace are also skipped during an import due to a namespace filter for com.sap.netweaver, com.sap.km and com.sap.portal. To avoid error messages during the import, set the export filter to “true”.

- You can exclude certain object-types from transports by customizing the property Pcd.TransportApplication.Export.ExcludeObjectTypes. There is no default value. For example, you can configure your transport scenario so that PAR files are excluded from the export. See also Packaging of Portal Transport Archives. Use cases of this example are discussed in 5.2.1 SDM as Central Deployment Tool.

4.1.2.3 Define Naming Conventions for Transport Packages

Transport packages are a specific PCD content type and must follow common naming conventions. A package ID is automatically generated when the package is exported. The package ID consists of

- <package name>
- <date in format yyyy mm dd>
- <time stamp in format hh mm ss>

Proposal for a naming convention for transport packages:

- <domain> e.g. **com.**
- <corporation> e.g. **com.sap.**
- <orgunit> e.g. **com.sap.hr.**
- <user/ project> e.g. **com.sap.hr.userID**
- <object type> e.g. **com.sap.hr.userID.pkg**
- <description> e.g. **com.sap.hr.userID.pkg.EmployeeRole**

Note: The length of a PCD object name is limited to 100 characters.

4.2 Manual Export

Exporting content means defining semantically correct packages. Export procedures therefore can hardly be automated.

As mentioned in 4.1.1 Customizing of Administration Roles, content administrators and developers are usually responsible for exporting complete and correct packages.

4.3 Export Reporting

Since EP 6.0 on 6.40 SPS9, log files are automatically generated when an export package is created. The location of these protocol files can be customized in the PCDstartup.properties: The property Pcd.TransportApplication.ProcessReportDir defines the storage directory for the log files. This directory is generated automatically based on the entries during portal installation. The default location for export log files is C:\usr\sap\<sid>\<instance_##>\j2ee\temp\pcd\transport\reports.

The protocol files are generated in XML format and can for example be used to

- Verify the functional correctness of the export package (e.g. any kind or error during export)

- Report about available transport packages (at a certain point of time), transport package authors, exported content, etc.

List of parameters provided in export protocol files:

<Meta-Data>	<Value>
Transport Report Version	1.0
Process ID	e.g. EXPORT-0208_101906_122_f176997758847b5c
type	EXPORT
state	FINISHED | CANCELED | ERROR
user	IUser (Request User/ Package Author)
startTime | endTIME	<e.g. 2005-02-08T10:19:06.563+01:00>
TotalObjects	<# of transported objects in epa-file>
processedObjects	<# of processed objects>
errorObjects	<# of objects exported with errors>
packageUrl	<relative location in Portal Catalog>
file	location in file system
Object url	<PCD URL>
type	com.sapportals.portal.transport.TransportPackage
State (of single object)	OK | ERROR | NO_PERMISSION | NOT_FOUND | NOT_TRANSPORTABLE
StartTime | endTime for single object	<e.g. 2005-02-08T10:19:06.563+01:00>
ObjectURL for any single object	e.g.<pcd:portal_content/administrator/super_admin/super_admin_role>

4.4 Summary: Exporting Portal Content

To sum up the guidelines for portal transport, keep in mind the following rules:

- Perform one export and two imports (see above)
  - Transport end user information, structured and aggregated in roles
  - Transport roles with dependent objects by using the include option and namespace filter.
  - Transport single objects for shared objects like generic templates!
  - Transport single objects for system objects => these transports require reworking in the target system by changing or reapplying the system alias.

- Apply changes to originals only, not to copies (see above)
  - Apply corrections to the original transport package and export the package again.

- Use unique object IDs when creating a package. This eases identification of package ownership.

- Analyze and store export log files of an export package to be verified before import if errors were reported.

5 Import into Quality Assurance System

Guidelines help you to design efficient import procedures for QA and PRD systems:

- Import packages by time stamp
- Import packages by ownership
- Schedule imports

5.1 Manual Import

Importing PCD objects (see chapter 3.3.1 Perform one export and two imports) means to:

- Add new objects to a system and /or …
- Replace and overwrite existing objects
- Update existing relationships between an overwritten object and other objects (e.g. overwriting an iView that is linked to a page means updating and refreshing the related page information as well).
- Create the PCD structure of the source system analogously in the target system.

By default, the import of transport packages is a manual process. A transport package can be imported from an end-user client or a predefined share (see chapter 3.3.1 Perform one export and two imports). PCD property Pcd.TransportApplication.ImportRootDir defines a central import share, which is by default C:\usr\sap\<sid>\SYS\global\pcd\Import. You can import packages from a client PC. This is mainly done for test reasons: After exporting a package, the administrator can download the package from the central file share and test the import into a target system.

Content can be imported by the system administrator role and requires certain permission settings. The request user who performs the import needs write access to all folders into which objects are imported.

There are two scenarios for planning imports:

- Content is imported asynchronously by many system administrators
- Content is imported synchronously by few administrators

Asynchronous Imports

A typical scenario for asynchronously importing content can be shown in a QA environment. Several content administrators of the DEV system are also assigned to the system administrator role in the QA environment, and regularly perform imports to test their own content in the test environment.

Another use is that several system administrators constantly check for new transport packages to import content.

When importing content, you should consider:

- If transport packages contain different versions of the same PCD object, the object is overwritten by any new import if the “overwrite option” is used.
  By default, the overwrite option is active. It can be switched on or off, and always affects the complete transport package content. Thus, the overwrite option must be used for content updates and it applies to the whole package.

- If transport packages contain an object and its dependent objects, the system administrator needs to have write access to all PCD target folders into which objects are imported. If the request user does not have sufficient permission to import an object, the affected iView will not be imported.

When defining import procedures for the QA system, keep in mind:

- Administrators assigned to the system administrator role who have access to dedicated parts of the PCD tree are responsible for content imports. To prevent permission clashes, imports can be performed:
  - By a few system administrators with write permission for certain parts of the PCD tree
  - By many system administrators with limited access to dedicated PCD folders.

- Packages are imported from a central file share only, and not from a client.

- Responsibility for content imports is defined by the prefix (see chapter 4.1.2.3 Define naming conventions for transport packages) of the import package.

- Packages are imported strictly by time stamps to guarantee that older content is overwritten by younger content.

- Several administrators can import packages at the same time: ACLs of the target system can be used to handle content overlaps of packages and thus prevent content from being overwritten unintentionally. Content is only imported to PCD folders to which the import user has write access.

Synchronous Imports

Permission clashes and unintended overwriting can be prevented during content imports by defining strict import procedures:

- All imports are carried out by a dedicated administrator with write access to the whole PCD tree.

- Packages are imported from a central file share only, and not from a client.

- Packages are imported strictly by time stamps to guarantee that older content is overwritten by younger content.

- Packages are imported within a scheduled maintenance window.

To support such guidelines, customer-specific development can be done as a project solution. Examples of customer-specific implementations are explained in the following paragraph 5.2 Customized Import.

5.2 Customized Import

The requirements for customizing the transport process are basically:

- Automatic import: Import is triggered by event or another tool.
- Scheduled import: Import is performed during a predefined timeframe.

5.2.1 SDM as Central Deployment Tool

The Software Deployment Manager (SDM) is the central deployment tool of Web AS 6.40 Java. When upgrading or patching a system, SAPInst uses the SDM to hand over deployable units, packaged as SDAs (Software Deployment Archive) or SCAs (Software Component Archive). The SDM offers a UI for (remote) access and a command line interface to trigger deployment.

Software packages like EPA-files as well as portal archives (PAR files) can be deployed by SDM like any other Java software package (e.g. Web Dynpro components). EPA files must be converted into an SDA format.

Scenario: Exclude PAR files from a transport package during export

A transport package for EP 6.0 content can include all dependent objects (see chapter 4.1.2 Design Rules for Transport Packages). This usually includes the related coding that is deployed as a portal archive. Only dependent objects within the SAP namespace (see chapter 4.1.2 Design Rules for Transport Packages > Export Filter) are not integrated into a transport package during the export. No additional deployment step for the Java code is therefore necessary when transporting a package with all dependent PAR-files: During import, PAR-files are *automatically* deployed to the target system.

To better separate coding from content, the default EP 6.0 transport behavior can be changed. PAR-files can be excluded from the export and deployed separately by SDM. By changing the following PCD properties you can exclude PAR-files from the transport:

- *Pcd.TransportApplication.Export.ExcludeObjectTypes* = Here you can configure the transport scenario to exclude PAR files from the export.
- *Pcd.TransportApplication.Export.ExcludeObjectTypesAfterRuleProcessing* = Here too you can decide which object types are excluded from the export. You can also define that dependent objects should be exported. There is no default value. For example, you can exclude PAR files from the export, but transport the corresponding resource bundles. To exclude PAR files, insert the line: **com.sapportals.portal.application.applicationrepository.Archive** as value for *Pcd.TransportApplication.Export.ExcludeObjectTypesAfterRuleProcessing*.

PAR files can be deployed separately from portal content by using either a portal application or the Software Deployment Manager (SDM):

- “Hot reload” using EP component http://<host>:<port>irj/servlet/prt/portal/prtroot/com.sap.portal.runtime.system.console.ClusterAdminConsole
- SDM: Requires conversion into another format.

An additional Make utility can be downloaded from the Service Market Place to support the conversion of PAR files into SDA files. As a prerequisite, you need the SAP NetWeaver Developer Studio.

For more details, refer to SAP Notes **696084** and **725797** and http://help.sap.com > SAP NetWeaver 04 > People Collaboration > Portal > Administration Manual > System Administrator > Transport, Upload, Content Mirroring > Transport of Portal Objects > Transport Scenarios > Packaging Portal Transport Archives.

Scenario: Deploying EPA files with SDM GUI

The SDM is a central deployment tool that can be used to define a synchronized and centralized transport process for EP content. After content has been exported to the file system, EPA files need to be converted to an SDA format to be deployable using the SDM (see previous paragraph). The new file can be stored on the file system and selected by an administrator with (remote) access to the SDM – user name and password are required to log onto the SDM server from the remote GUI application.

In addition to the import protocol files, which are still written to the file system (see chapter 5.3 Import Reporting), the SDM also protocols the deployment of a transport package:

- any deployment of SDM is kept in the SDM file system
- metadata is stored in the file system for any deployment
- log output about SDM deployments is directed to /usr/sap/<sid>/<instance>/SDM program/log. Log files can for be viewed from the Log Viewer tab of the SDM Remote GUI client or alternatively with the Web AS Java Log Viewer Application (see documentation on help.sap.com for further information about the Web AS 6.40 Java Log Viewer).
- redeployment of the same file is recognized by SDM

Documentation about the SDM can be found in the directory path of the SDM, running on each central instance of a Web AS 6.40 Java system: /usr/sap/<sid>/instance_##/SDM/program/doc.

Scenario: Using SDM command line interface for EPA file deployment

The SDM offers a command line interface, which can be used to deploy files. Customers can use this interface to design a transport process according to their needs, e.g. imports can be scheduled using customer-specific scripts, triggering multi-package deployment.

Basically, the following prerequisites must be met to deploy portal transport packages from the SDM:

1. Existing EPA files of the export directory must be converted to SDA files. This process is supported through ANT-based tools (“Make Utility”), which are available through SAP Note **696084.**
2. The converted files are in a new share, which is used for deployments and imports, e.g. a predefined deployment share for the SDM.

3. To enable deployment, the SDM on the central Java Instance must be stopped and set to “standalone” mode.

    a. An easy way to stop the SDM is to use the StopServer.bat /StopServer.sh file, which is located in the program directory of the SDM.
    b. To set the SDM to standalone mode, use the following command: sdm jstartup "mode=standalone"

4. The SDM deployment can then be started from its command line interface. The syntax to initiate any deployment is documented in “Command Line Interface Documentation for SDM 6.30/6.40” (see previous chapter).

    a. The SDM can deploy single files with the command: sdm deploy "file=<path>, e.g. "C:\usr\sap\J2E\SYS\global\pcd\Export\<file>". To deploy multiple files, the command should be executed multiple times, each time selecting a new file. The return code for each deployment indicates that the deployment was completed.
    b. Alternatively, the SDM can deploy files listed in a text-file. If a list of files is passed to SDM for deployment, the files are sorted by SDM – it is not possible to enforce a deployment order.

5. Return codes indicate whether or not the deployment was successful. The list of return codes for the commands *deploy, undeploy, shutdown* can be found in “Command Line Interface Documentation for SDM 6.30/6.40” (see previous chapter).

6. After the deployment has finished, you must switch the SDM back to “integrated” mode and restart it.

    a. To set the SDM to integrated mode, use the following command: sdm jstartup "mode=integrated".
    b. An easy way to start the SDM is to use the StartServer.bat /StartServer.sh file, which is located in the program directory of the SDM.

7. The SDM keeps information about deployed files in its own repository.

    a. The same transport package is not redeployed unless the option to apply a lower version is active.
    b. It is not possible to undeploy a transport package from the SDM since the SDM cannot access PCD tables to remove deployed content.
    c. Import protocol files are written to the file system as for a standard (manual) import.

8. Troubleshooting deployments

    a. SDM deployment errors are written to /usr/sap/<sid>/<instance>/SDM program/log
    b. Errors during (portal) import are written to file system/server.log.

More specific information on the SDM can be found in the following SAP Notes:

- SAP Note 756084: Commonly met SDM v6.30/6.40 problems
- SAP Note 795142: Portal file deployment failed - Troubleshooting

5.2.2 Importing Content Using SAP NWDS and NWDI

It is also possible to transport EPA files through the different system stages DEV – QA – PRD using the SAP NetWeaver Development Studio (NWDS) and NetWeaver Development Infrastructure (NWDI).

The NetWeaver Development Infrastructure is the strategic environment for enabling change management for the entire NetWeaver platform. A tight integration of the portal content development tools with the NWDI is planned for a future NetWeaver release. This document describes how portal content can be transported with the NWDI in the NetWeaver ’04 release. As an intermediate solution, you can connect to the NWDI using the NetWeaver Developer Studio.

Common use cases for setting up the transport process using NWDS and NWDI are:

- The customer is using the NWDI for his own Java Development.
- Creating and updating portal content is a typical developer task.
- A developer creates new content and releases it as a development component.

Additional configuration is necessary to set up a transport process using NWDS and NWDI. The main transport steps for this scenario are:

1. Transport packages are created in the portal: The portal content objects (iViews, pages, worksets, roles, etc) are created and maintained in the development system with the browser-based portal editors. A transport package is created and exported to transport the content.

2. The transport package is exported to the file system as an EPA file. The resulting Enterprise Portal Archive (EPA) file can be downloaded to the local file system.

3. Use the NetWeaver Developer Studio to work with the NWDI. The connection to the NWDI backend servers is established by importing the corresponding development configuration. All development objects are organized by development components (DCs) having different types. For portal content, the development component type is “Content/Business Package”. Configuration of the NetWeaver Developer Studio to enable the creation of DCs of this type is described below.

4. Copy the EPA file to the local NWDS file system create a new development component for the EPA file. The relationship of a portal content transport package to a DC is 1:1. We recommend that transport packages and DCs be either completely created by the system administrator or that the user apply the guidelines for content packaging and component structure.

5. Once the DC has been created, check the exported content archive (EPA file) into the Design Time Repository (DTR).

    - If it is a new file, it is checked out for “Add”, if it is an update of a package that was already transported, it is checked out for “Edit”.
      Changes to versioned files in the DTR are organized in activities. These objects allow you to bundle several changes and to specify a name and description for the change.

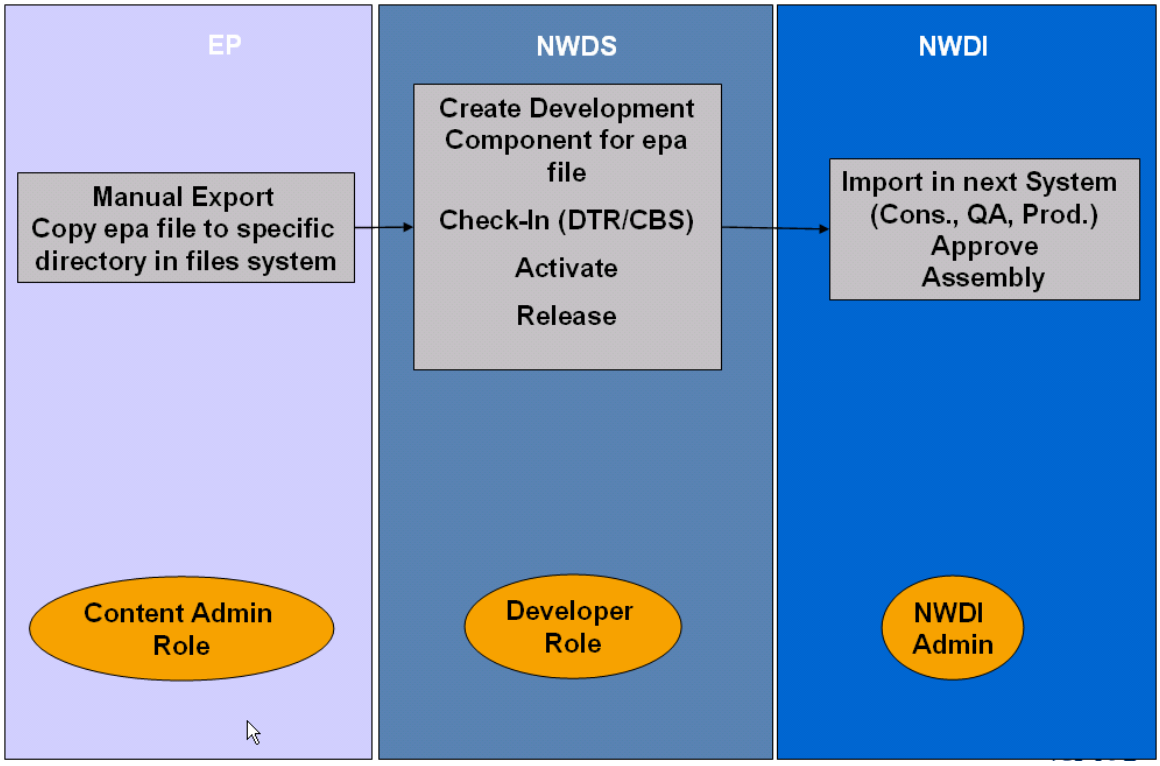
- Check the activity into the DTR, resulting in a new version of the EPA file.

- After check-in, the activation is completed. This means that a central build is triggered in the Component Build Service (CBS). Normally, this comprises a compilation or generation step. For portal content DCs, the build only consists of a packaging step that creates a deployable Software Component Archive (SDA) for the content.

6. The developer then creates a transport request and releases the changes. A transport request can comprise several activities (it would also be possible to bundle changes in Java code or WebDynpro components with changes in the corresponding portal content objects).

7. All released transport requests are displayed in the Change Management System (CMS) import queue. The system administrator triggers the import of waiting transport requests, which causes integration of all changes in the corresponding consolidation DTR workspace, a build on the CBS, and automatic deployment in the consolidation runtime system.

8. The content developer or a quality engineer can verify the changes in the consolidation system.
   Finally, the system administrator assembles all verified changes to a new software component version and approves the release to the production system.



Benefits:

- Standard procedures of release management for Java development components can be used for EP 6.0 content.

- Content transport is supported through the NWDI workflow for the complete system landscape (import of the same development component into QA and PRD).

Configuration of SAP NetWeaver Developer Studio

To create portal content development components, modify file

`"C:\Program Files\SAP\JDT\eclipse\plugins\com.sap.ide.eclipse.component.provider\componentTypes.xml"` as follows:

Add the section

```
<sub-type name="Business Package" caption="Enterprise Portal Content
Archive" icon="type/content_business_package.gif" selectable="true">
    <wizard pluginid="com.sap.ide.eclipse.umepermissions"
    class="com.sap.ide.eclipse.content.NewContentWizard" />
    <build name="tc/bi/bp/content" vendor="sap.com" scalias=""
    ppref="content" />
</sub-type>
```

under the tag

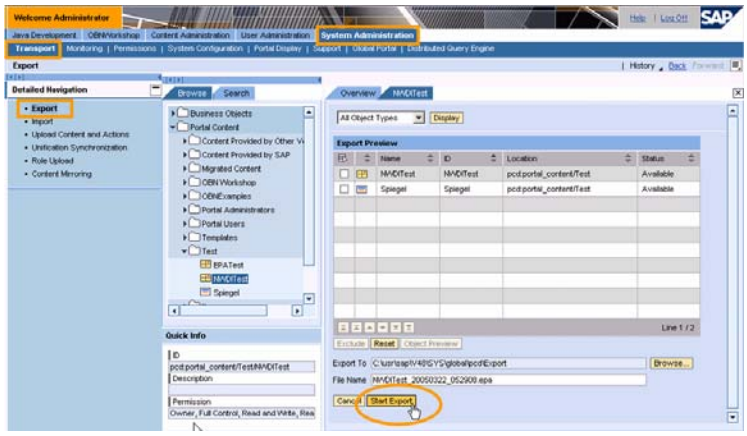
```
<type name="Content" icon="type/content.gif" selectable="false">
```

After restarting the NWDS, the new development component type should be available in the DC creation wizard.

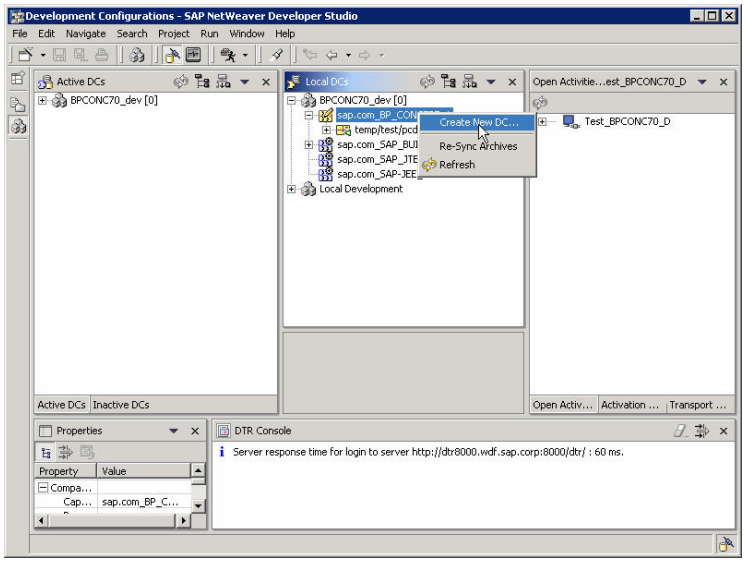
A further prerequisite is to set up the NWDI. Read the documentation on help.sap.com > NetWeaver 04 > SAP NetWeaver > Solution Lifecycle Management > Software Change Management.

Screenflow: How to use NWDS & NWDI to transport EPA files

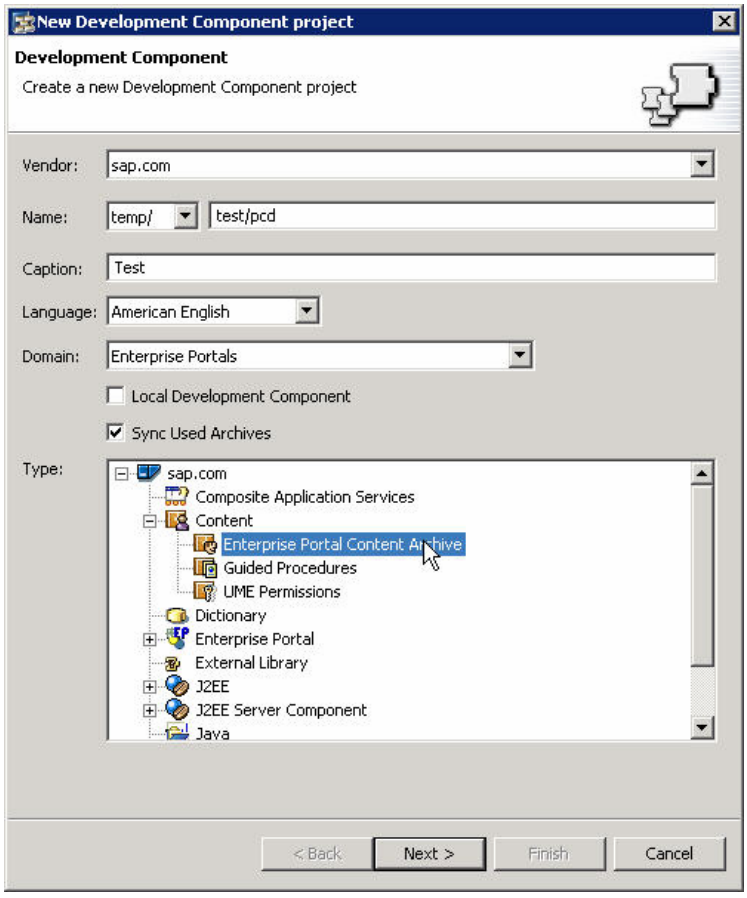
1. In the Portal, create a new transport package.



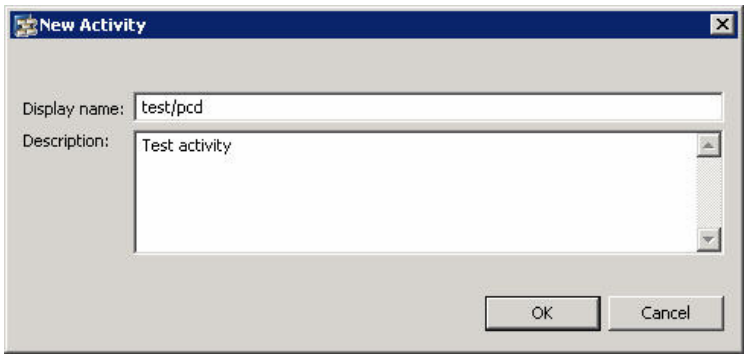
2. In SAP NWDS, enter the development configuration perspective and create a new development component.



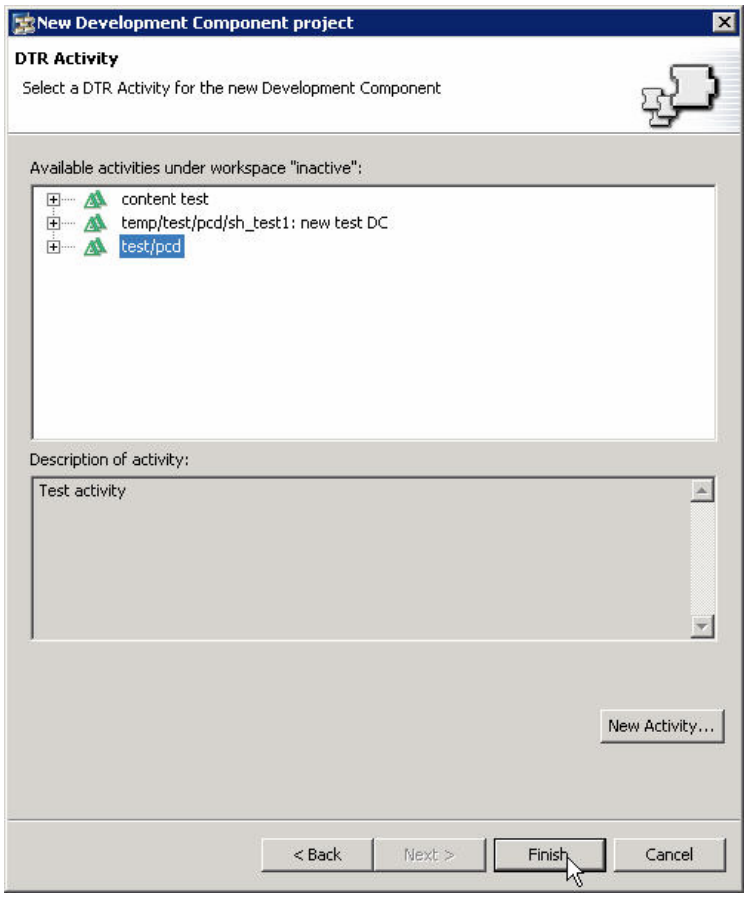
3. For the development component, select the type “Enterprise Portal Content Archive” below the sap.com > Content.



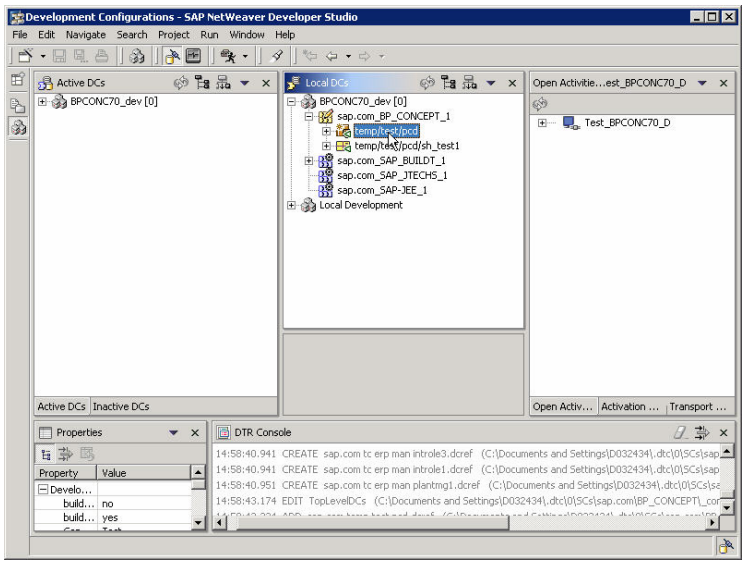
4. You are guided to a screen where you can define a new DTR activity.



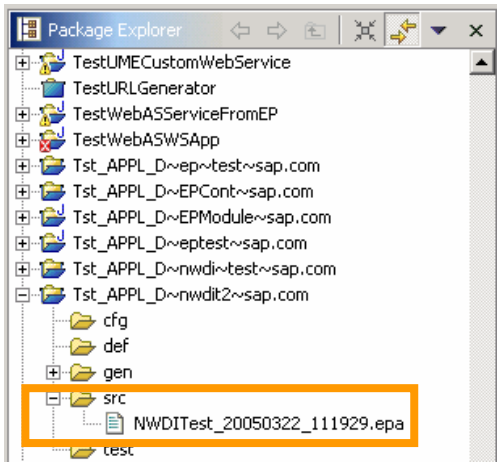
5. The DTR activity appears; you have to select it to continue.



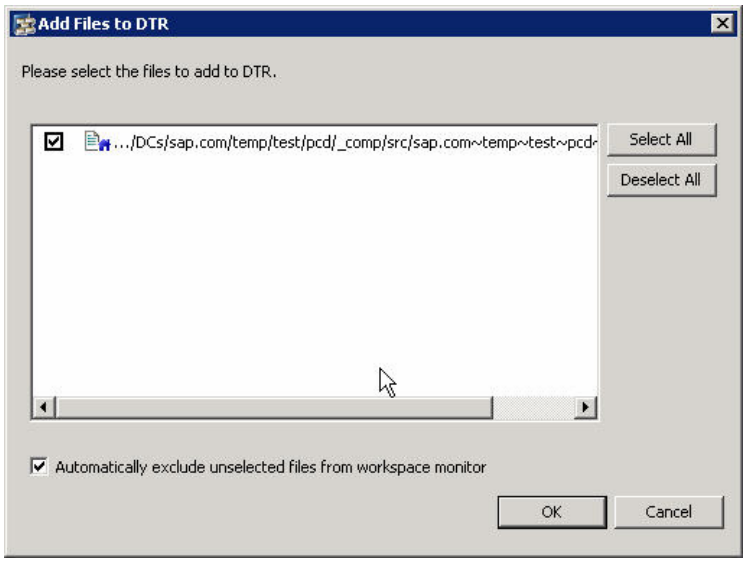
6. After finishing the wizard steps, this screen appears.



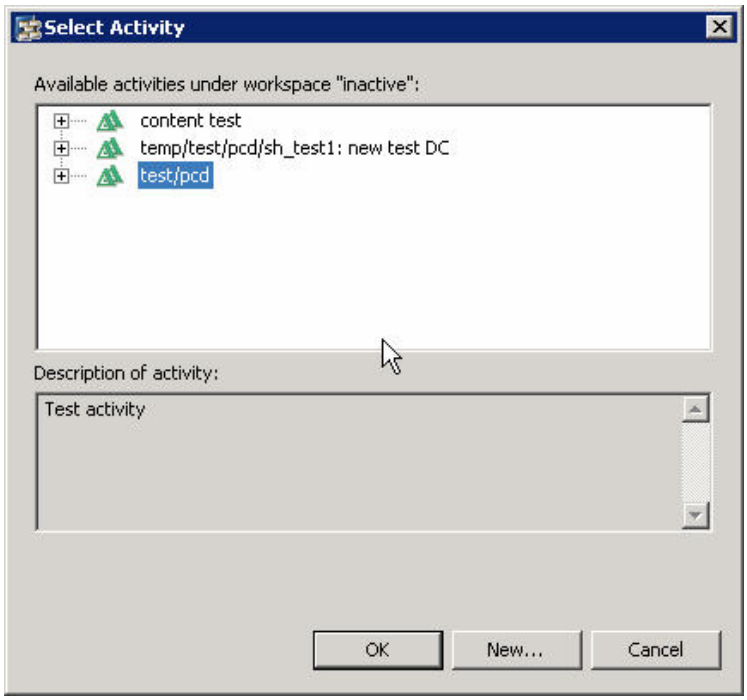
7. Copy and paste the EPA file into the src-directory of your DC project.



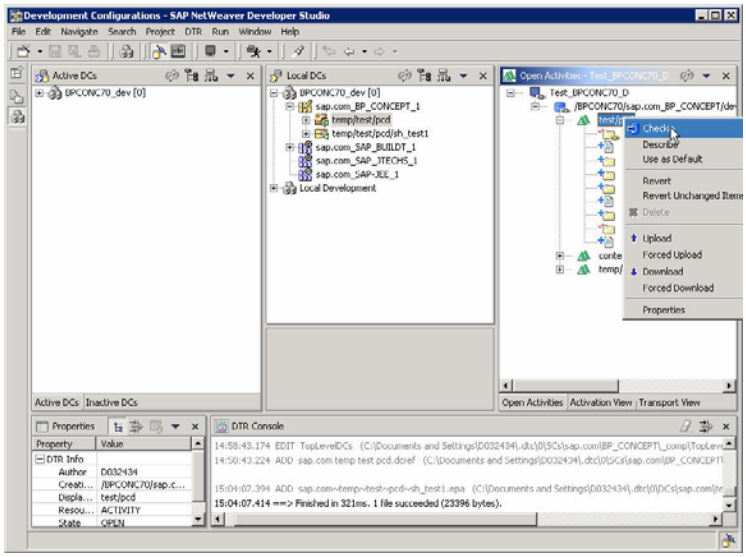
8. This screen appears; select the files to be added to the DTR. Choose "OK".



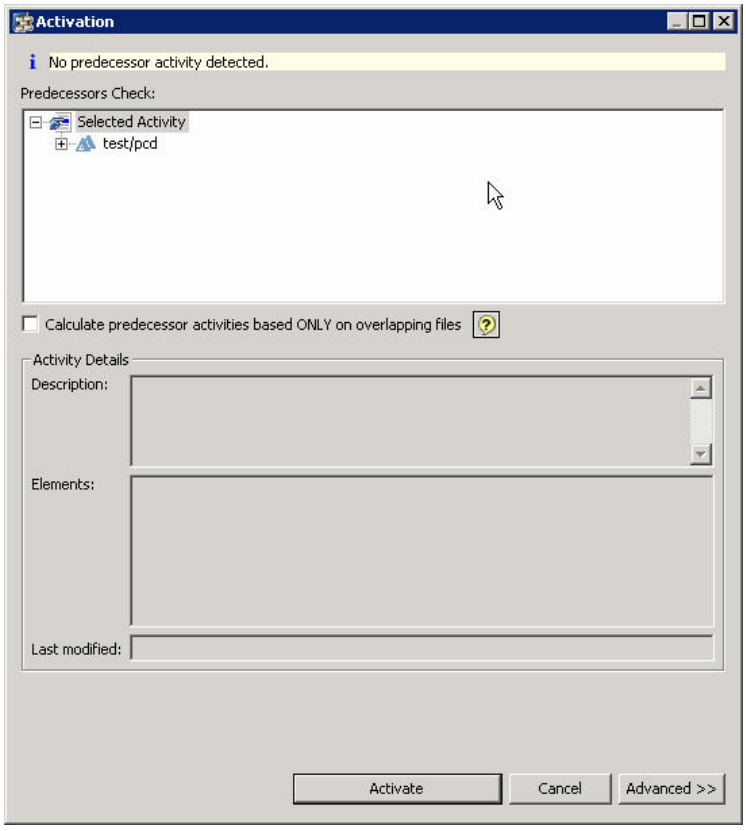
9. Select the DTR activity that was created in step 4.



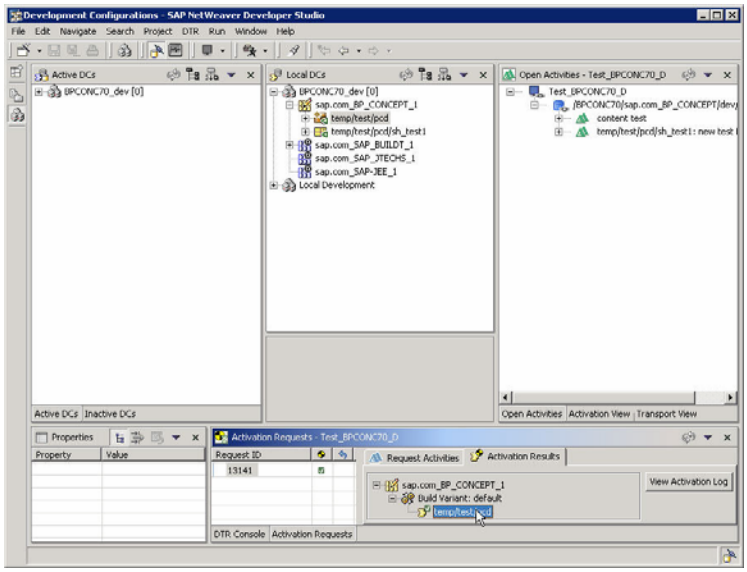
10. In the “Open Activities” view of the development configuration perspective, right-click the activity and choose “CheckIn”.



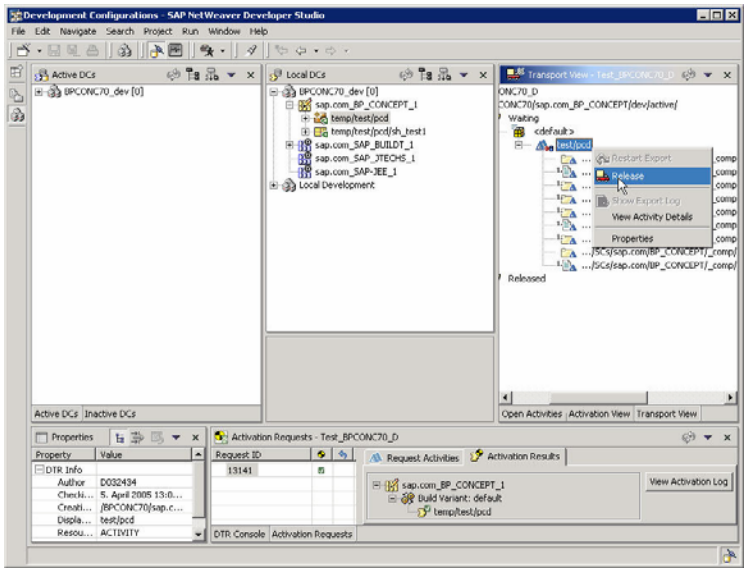
11. A wizard guides you to this screen, where you can activate your DC.



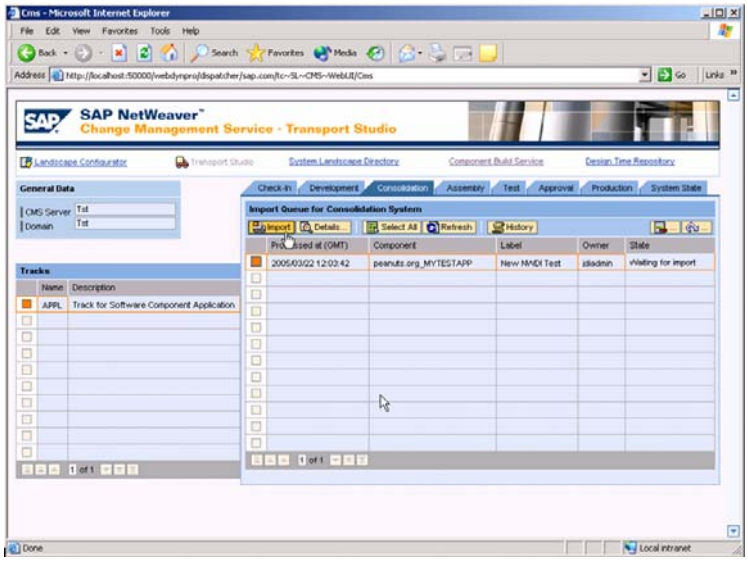
12. Open the “Activation Requests” view to monitor the activation process (wait until the green arrow appears).



13. Switch to the “Transport” view to release your DC to the QA system.



14. The DC component is transported to other systems according to standard NWDI procedures.



5.3 Import Reporting

5.3.1 Transport Log Files

Since EP 6.0 on 6.40 SPS9, log files are automatically generated when a transport package is imported. The location of these files can be customized in the PCDstartup.properties: Property Pcd.TransportApplication.ProcessReportDir defines the storage directory for the log files. This directory is generated automatically based on the entries during portal installation. The default location for export log files is C:\usr\sap\<sid>\<instance_##>\j2ee\temp\pcd\transport\reports (see chapter 4.3 Export Reporting).

The log files are generated in XML format and can for example be used to

- Verify the correctness of the import (e.g. any kind or error during import)
- Report about the content of transport packages (at a certain point of time), transport package authors, objects included in the transport package, etc.

List of parameters provided in import log files:

<Metadata>	<Value>
Transport Report Version	1.0
Process ID	e.g. IMPORT-0208_101906_122_f176997758847b5c
type	IMPORT
state	FINISHED | CANCELED | ERROR
user	IUser (Request User)
startTime | endTIME	<e.g. 2005-02-08T10:19:06.563+01:00>
TotalObjects	<# of transported objects in EPA file>
processedObjects	<# of processed objects>
errorObjects	<# of objects exported with errors>
packageUrl	<relative location in Portal Catalog>
file	Location in file system
Object url	<PCD URL>
type	com.sapportals.portal.transport.TransportPackage
State (for single object)	OK | ERROR | EXISTS_ALREADY| NO_PERMISSION
StartTime | endTime for single object	<e.g. 2005-02-08T10:19:06.563+01:00>
ObjectURL for any single object	e.g.<pcd:portal_content/administrator/super_admin/super_admin_role>

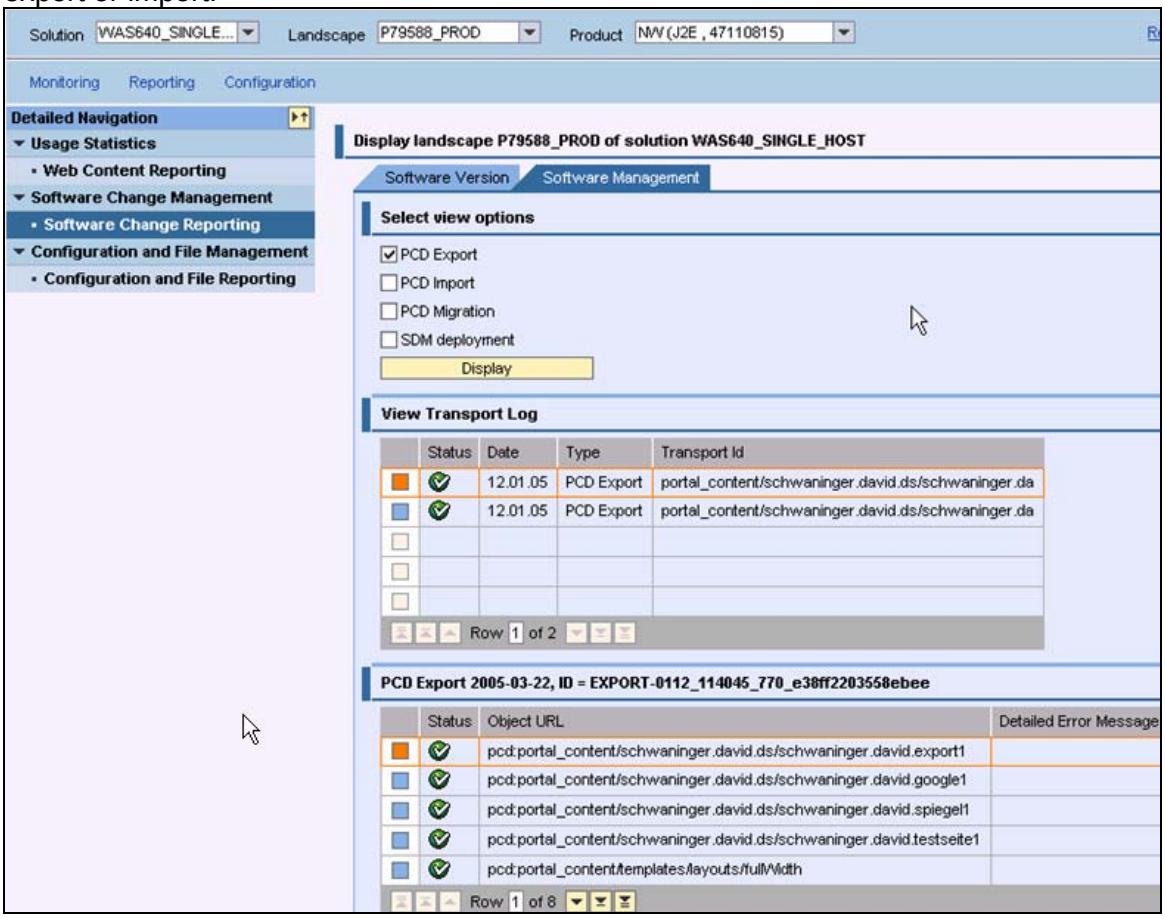
5.3.2 Software Change Tracking Component of SMD

To efficiently support EP 6.0 systems, the SAP support organization requires that you set up some tools to monitor, maintain and troubleshoot a portal installation. They are delivered as a software package called Solution Manager Diagnostics.

For detailed information, see http://service.sap.com/diagnostics.

One tool of the Solution Manager Diagnostics is the Software Change Tracking component. Software changes on a dedicated host on which a SAP Enterprise Portal is installed can be tracked and reported. The date and content of a transport package are displayed. Any errors that occurred during the export or import are reported.

Note: The Software Change Tracking component is for reporting only, not for initiating an export or import.



5.4 Summary: Import into Quality Assurance System

There are different approaches for setting up an appropriate infrastructure for importing transport packages into the QA system:

- Manual Import
  - Asynchronously – by many administrators
  - Synchronously – by one administrator

- Customized Import
  - Custom development: Centralized deployment using SDM
  - Import using NWDI and NWDS

Procedures also have to be defined for the following actions:

- Adapt role-user assignments in the target system

- Adapt ACLs of the Portal Catalog: Additional tools that support permission changes are described in SAP Note 696874 (also see the How To Guide “Initial Permission Creation for EP 6.0 SPS9 +).

- Adapt aliases and system definition of imported system objects

6 Testing in Quality Assurance Systems

Test procedures need to be defined and documented separately for components. Follow these guidelines when setting up test scenarios:

- Business users are responsible for testing new content
- Tests must verify if new content is functionally and semantically correct
- Test approvals are required before further transport actions: Import will only be performed if no errors were found in the QA environment
- Tests must be scheduled
- Tests must be documented

Test procedures can for example be structured using roles (see chapter 4.1.2.1 Transport End-user Relevant Information):

- A role was created in the DEV system and is transported into the QA system as a unit with its dependent content.
- If the imported role is a new object, a test user account for the role has to be created and assigned to the role.
- After initial creation, the role-user assignment remains permanently in the QA environment. Whenever the role is updated, the role-user assignments are automatically retained.
- The test user logs onto the QA portal with his test account. Ideally, imports and tests are scheduled for a predefined timeslot: e.g. scheduled imports during times of low end-user activity, such as 2:00 – 3:00 am; tests scheduled for next day between 8:00 a.m. – 5:00 p.m.
- The test user tests functions and features of the role:
    - Is the navigation structure correct?
    - Is the UI of the content correct?
    - Are pages and iViews displayed correctly?
    - Can information from backend systems be accessed?
    - Does the role provide information appropriate for its target user group? (Ideally, the test user is responsible for a special target group such as: managers, all employees, employees of a certain business unit)
- Test users document their results in a central document (e.g. Excel file). The documentation file needs to include information that is provided by the export and import log files (see chapters 4.3 Export Reporting and 5.3 Import Reporting):
    - Transport package ID must be specified
    - Transport type: IMPORT
    - State: FINISHED | ERROR | CANCELED
    - ID of included objects
    - ID of included role
    - Object paths (PCD location)
    - Test user account: ID
    - Test description: Short description about the test case

    - Test status: OK | ERROR

- An approval workflow for content imports should be set up to ensure that content is tested carefully before being imported to the PRD environment.
    - If the test indicates that the transported content is correct, the transport package file can be prepared for further imports, e.g. moved to the import share.
    - If the test indicates errors in the transport package, the transport package needs to be prepared for changes, e.g. moved to a correction share. Transport package authors can be found from either the corresponding export protocol (see chapter 4.3 Export Reporting) or by the package namespace.
    - After applying the required changes to the transport package in the DEV environment, the package is exported again to generate a new transport package file in the file system.

7 Importing into Production System

7.1 Import Procedure

This section refers to chapter 5 Import into Quality Assurance System. Import procedures for the QA and PRD environment are set up in a similar manner. The same guidelines apply to both systems:

- Import packages by time stamp
- Import packages by ownership
- Schedule imports

Procedures also have to be defined for the following actions:

- Adapt role-user assignment in the target system, if necessary
- Adapt ACLs of the Portal Catalog: Additional tools to support permission changes are described in SAP Note 696874 (also see the How To Guide “Initial Permission Creation for EP 6.0 SPS9 +).
- Adapt aliases and system definition of newly imported system objects

7.2 Deletion of PCD Objects in the PRD System

Objects in the PRD system can be overwritten, but not automatically deleted by transports.

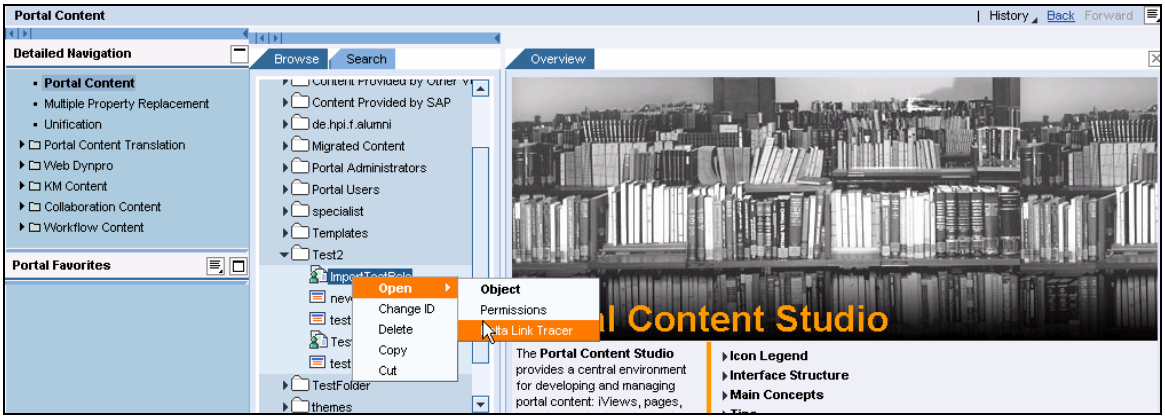
Example:

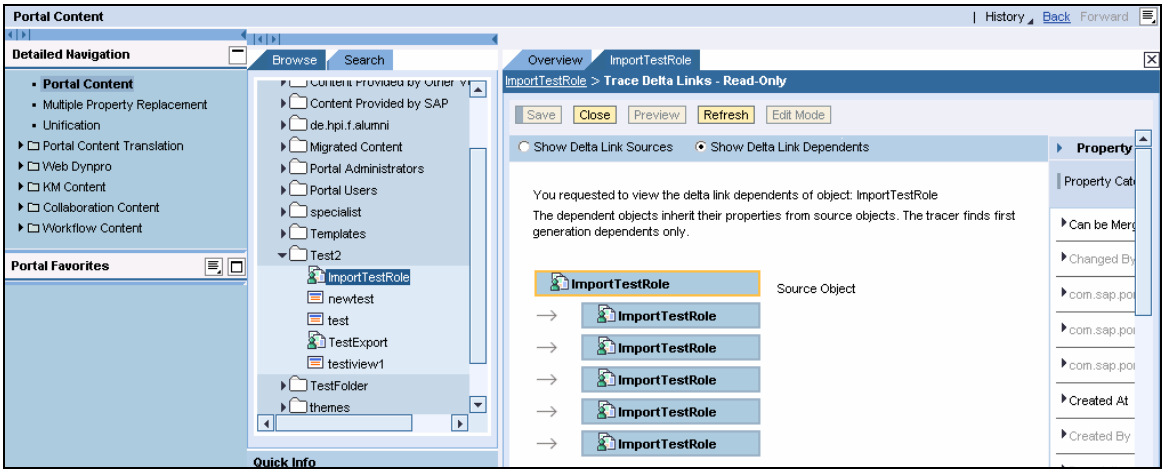
A role was released in the target system and is to be updated. The content administrator of the role reworked the role object by deleting workset W1 and adding a new one – W2. Importing the role means: The role object is overwritten, and as a result it displays the content of W2. The delta links of the roles to W1 no longer exist.

Nevertheless, W1 is not deleted in the target system, as it might also be referenced by other objects.

Deletions in the portal thus need to be planned carefully. The first step is to identify objects that need to be deleted; this is mainly a manual step (a future service pack release will include tool enhancements). Before you delete an object, you should make sure that it does not reference an object that is currently in use.

The delta link tracer tool helps to identify an object's backward and forward dependencies.

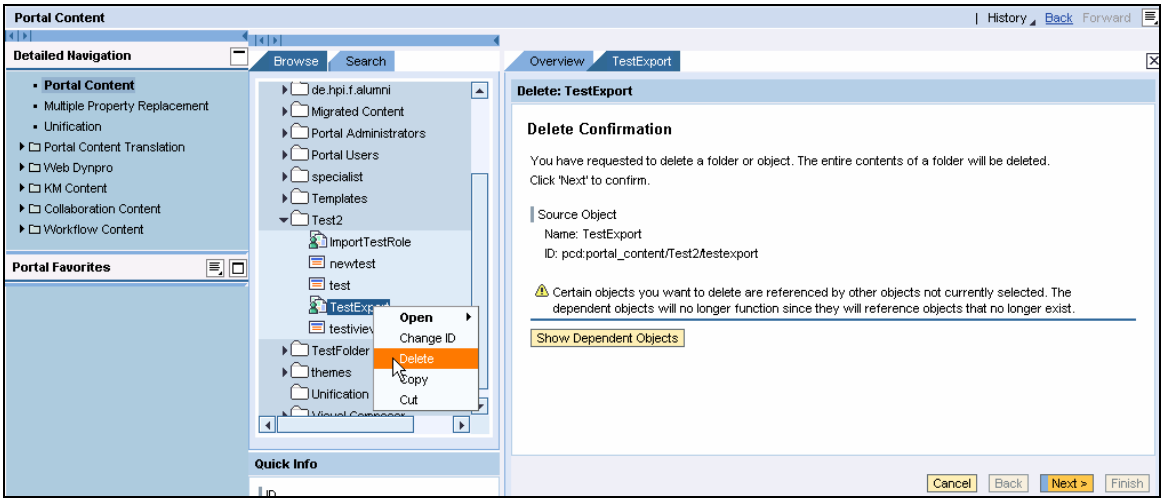


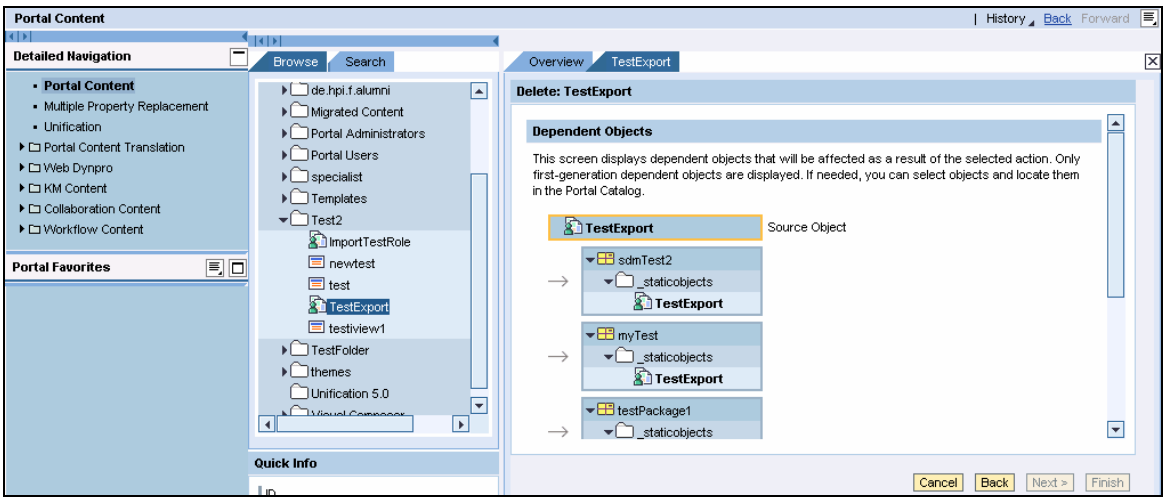


The portal provides the following tool for deleting objects in the target system

Delete option available in Portal Catalog

- To delete an object, you need full control access to the object
- A wizard that offers dependency checks (since SPS11) supports the deletion of objects
- Dependent objects can be located in the Portal Catalog (since SPS11)


Portal Content
History Back Forward
Detailed Navigation
Portal Content
Multiple Property Replacement
Unification
Portal Content Translation
Web Dynpro
KM Content
Collaboration Content
Workflow Content
Portal Favorites
Browse
Search
de.hpi.f.alumni
Migrated Content
Portal Administrators
Portal Users
specialist
Templates
Test2
ImportTestRole
newtest
test
TestExport
testiview1
TestFolder
themes
Unification
Open
Change ID
Delete
Copy
Cut
Quick Info
Overview
TestExport
Delete: TestExport
Delete Confirmation
You have requested to delete a folder or object. The entire contents of a folder will be deleted.
Click 'Next' to confirm.
Source Object
Name: TestExport
ID: pcd:portal_content/Test2/testexport
Certain objects you want to delete are referenced by other objects not currently selected. The dependent objects will no longer function since they will reference objects that no longer exist.
Show Dependent Objects
Cancel
Back
Next >
Finish


Portal Content
History Back Forward
Detailed Navigation
Portal Content
Multiple Property Replacement
Unification
Portal Content Translation
Web Dynpro
KM Content
Collaboration Content
Workflow Content
Portal Favorites
Browse
Search
de.hpi.f.alumni
Migrated Content
Portal Administrators
Portal Users
specialist
Templates
Test2
ImportTestRole
newtest
test
TestExport
testiview1
TestFolder
themes
Unification 5.0
Quick Info
Overview
TestExport
Delete: TestExport
Dependent Objects
This screen displays dependent objects that will be affected as a result of the selected action. Only first-generation dependent objects are displayed. If needed, you can select objects and locate them in the Portal Catalog.
TestExport
Source Object
sdmTest2
_staticobjects
TestExport
myTest
_staticobjects
TestExport
testPackage1
_staticobjects
Cancel
Back
Next >
Finish

www.sap.com/netweaver