SAP NetWeaver
How-To Guide

Best Practices for Implementing CTS+

Applicable Releases:

SAP NetWeaver 7.0

IT Practice:

Unified Life-Cycle Management

IT Scenario:

Software Life-Cycle Management

Version 1.0

April 2008

THE BEST-RUN BUSINESSES RUN SAP™

© Copyright 2008 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Sun Microsystems, Inc.

JavaScript is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

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.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP’s Support Services and may not be modified or altered in any way.

Document History

Document Version	Description
1.00	First official release of this guide

Typographic Conventions

Type Style	Description
Example Text	Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Cross-references to other documentation
Example text	Emphasized words or phrases in body text, graphic titles, and table titles
`Example text`	File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.
`Example text`	User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation.
`<Example text>`	Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.
`EXAMPLE TEXT`	Keys on the keyboard, for example, F2 or ENTER.

Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

Table of Contents

1. Scenario 1

2. Prerequisites 1

3. Enhanced CTS Configuration 2

3.1 CTS+ Components - Overview 2
3.1.1 Setup CTS+ Server components: ABAP stack 4
3.1.2 Setup CTS+ Server components: JAVA stack 5
3.1.3 Setup CTS+ Server components: Separate stacks 5

3.2 TMS System Landscape Configuration 7
3.2.1 Configuration of Source Systems 8
3.2.2 Configuration of Target Systems 9
3.2.3 Configuration of Transport Routes 10

3.3 CTS+ Application Integration (Close Coupling) 10
3.3.1 Example – Use Case EP 11
3.3.2 Example – Use Case NWDI 12
3.3.3 Matrix 13

4. Landscape Scenarios 16

4.1 Central Transport Management 16

4.2 Landscape Patterns 18
4.2.1 Non-ABAP Landscape 18
4.2.2 Dual Stack Landscape 19
4.2.3 Enhanced ABAP Landscape 20
4.2.3.1 CTS+ Configuration on dedicated ABAP and Java System 21
4.2.3.2 Distributed CTS+ Configuration 23

4.3 Combined Landscape Scenarios 26
4.3.1 SLD and PI Landscape 26
4.3.1.1 Single productive SLD System 26
4.3.1.2 Separate SLD Landscape 27
4.3.1.3 Combined PI and SLD System Landscape 28
4.3.1.4 Separate PI and SLD System Landscape – One Transport 28
4.3.1.5 Combined PI and SLD System Landscape – separate Productive Systems 29
4.3.1.6 Combined PI and SLD System Landscape – no QA SLD System 30
4.3.1.7 Combined PI and SLD System Landscape – no DEV SLD System 32
4.3.2 NWDI and SAP Portal Landscape 34
4.3.2.1 Combined SAP Portal and NWDI/Java landscape 34
4.3.2.2 Separate Landscape for SAP Portal and NWDI/Java landscape 35
4.3.3 Enhanced CTS Configuration in a Complex Landscape 36

1. Scenario

This guide explains the architecture and main components of the enhanced Change and Transport System (short: CTS+). Chapter 3 gives an overview of the relevant CTS+ components and configuration steps in general whereas chapter 4 discusses different system landscapes and gives examples on how to configure simple and complex system landscapes supporting different use cases.

2. Prerequisites

The SAP NetWeaver Release of the system where the CTS+ relevant components are configured should be SAP NetWeaver 7.0 SP Stack 14 to follow this guide. Nevertheless this document should be a generic guideline in understanding the setup, configuration and usage of CTS+.

Further information:

- See central SAP Note: 1003674
- Standard documentation in the SAP Library: http://help.sap.com/saphelp_nw70/helpdata/en/45/ec25370fdc3481e10000000a1553f6/frameset.htm
- Required/recommended expertise or prior knowledge:
  - Basic knowledge of ABAP Change and Transport System
  - Basic knowledge of Enhanced Change and Transport system

3. Enhanced CTS Configuration

To take advantage of the enhanced Change and Transport System several components and configuration steps are needed. Chapter 3.1 gives an overview of the relevant CTS+ parts that need to be setup, installed, activated or configured on the ABAP and JAVA stack of a CTS+ enabled system.

For the usage of CTS+ at least one ABAP and one JAVA stack is needed. The recommendation is to use the SAP Solution Manager system, which is a dual stack system. CTS+ does not need any Solution Manager functionality, it is a pure SAP NetWeaver technology. But the Solution Manager system can act as central transport management system for the whole landscape (see chapter 4 for more details).

This chapter describes the configuration of the CTS+ relevant components for one central system (simplest case: a dual-stack system). In Chapter 3.1.3 the configuration steps are described for a landscape where the ABAP stack and the JAVA stack are installed as different systems (SIDs) and run probably on different hosts. Depending on the use case and system landscape, the CTS+ components will be distributed. This is discussed and explained in chapter 4.

After CTS+ is enabled, which means the CTS+ relevant server components are available and configured, the different system landscapes need to be defined for the specific use cases (such as Portal track or PI track). The definition of the systems for each landscape with its transport route is done as TMS configuration which is described in 3.2.

Last but not least CTS+ is tightly integrated in the applications – like Enterprise Portal, SLD, PI – so that the developer or content creator does not need to leave the application in order to be able to attach the non-ABAP objects to a transport request. Chapter 3.3 will give an overview of that configuration part.

3.1 CTS+ Components - Overview

This chapter explains the additional components and configuration steps that are needed to enable the enhancements of a CTS system landscape to be able to transport non-ABAP objects. The main concepts of the Change and Transport System (such as TMS systems, transport layers, transport routes) are still valid and required for a CTS+ landscape. The main parts of the enhanced CTS are shown in Figure 1 and explained below. The decision where to configure these components depends on the use case (such as PI, EP, NWDI) and system landscape especially when a complex system landscape needs to be enabled for CTS+ functionality. Figure 1 is an example for a central dual-stack system that acts as domain controller for a non-ABAP landscape. Here all server components are configured within the same system. See chapter 3.1.1 and 3.1.2 for more details. Chapter 3.1.3 explains the configuration in case the ABAP and Java stacks do not run in the same system.

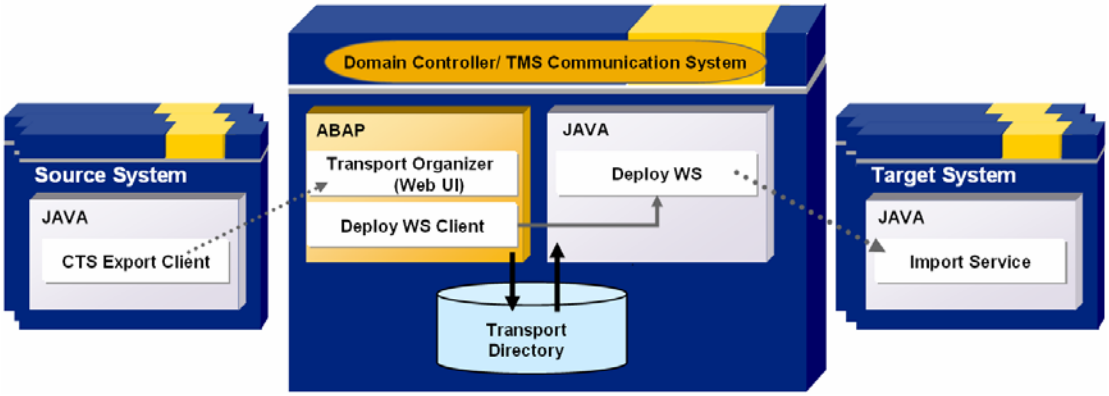


Figure 1: CTS+ components – dual stack configuration

Enhanced CTS relevant components and configuration parts in more detail:

- **Domain Controller (ABAP)**: The domain controller is the system, where the TMS transport systems (ABAP and non-ABAP) with their transport layer and transport route are configured.
- **TMS Communication System (ABAP)**: The communication system is the ABAP system, where the transport program *tp* is triggered to perform the transport steps. For source systems the communication system covers the *Transport Organizer Web UI*, for target systems the communication system triggers *tp* together with the *Deploy Web Service Client*. By default the Domain Controller is used as Communication System. But that is not mandatory as any other system can act as Communication System.

  Important

  When the Communication System for a non-ABAP system has to be changed after creating the system, the Transport Tool parameter *COMMUNICATION_SYSTEM* and the *Communication System* in tab Communication has be changed identically.

- **Deploy Web Service Client (ABAP)**: The Deploy Web Service Client is needed on the ABAP side to enable the transport control program *tp* on SAP NetWeaver AS ABAP to communicate with the CTS Deploy Web service on SAP NetWeaver AS Java. It consists of two configuration parts (for more details see Figure 2: Deploy Web Service Client Configuration):
  - HTTP Connection
  - Logical Port
- **Deploy Web Service (JAVA)**: This is a Java Web service responsible for communicating with the deployment/import services in the non-ABAP target systems. The transport control program `tp` communicates with the CTS Deploy Web service when importing/deploying the non-ABAP objects.
- **Transport Directory (ABAP)**: directory to store the data and log files between transports. The ABAP stack writes the data files in that directory and the Deploy Web Service in the Java stack reads to data files for imports to the target system. In addition to the standard role of the transport directory for ABAP transports the transport directory gets a new role for CTS+ non-ABAP transports. Here the data sub directory of the transport directory is needed to transport the non-ABAP files between the ABAP stack (tp/R3trans) and the Java stack (Deploy Web Service).

  Note

  R3trans creates a subfolder in the <DIR_TRANS>/data directory with the name of the transport request. All attached non-ABAP files of that transport request are copied into that folder. During Import the Deploy Web Service reads these non-ABAP files and transfers them to the Import Service (e.g. SDM of the target system).

- **Transport Organizer Web UI (ABAP)**: Web UI implemented as Web Dynpro ABAP application with basic SE09 functions to create and maintain transport requests for non-ABAP systems and Java stacks of dual-stack systems, attach non-ABAP objects and release transport requests.
- **CTS Export Client**: tight integration of CTS+ in the application of the source system where the non-ABAP objects are created (known as close coupling – available since SP Stack 14 for EP, NWDI, SLD, PI, KM); to enable this tight integration an RFC destination to the TMS communication system needs to be configured in the Visual Administrator of the source system (except for NWDI use case – here the NWDI server needs to be configured for tight integration, not the non-ABAP source system). The Export Client communicates with the enhanced Transport Organizer to fetch a proper transport request and to attach the non-ABAP objects directly within the application (e.g. EP Content Studio – Export Wizard).
- **Import Service**: import mechanism running on the target system (SDM, XI Import Tool, SLD Import, KM Import, etc.). During import of non-ABAP objects the transport control program *tp* calls the Deploy Web Service on the Java stack and the Deploy Web Service triggers the CTS+ Import Service on the target system (e.g. SDM) to deploy the transported objects.

3.1.1 Setup CTS+ Server components: ABAP stack

- **TMS Domain Controller**: the ABAP system is configured as TMS Domain Controller (transaction *STMS*).
- **TMS Communication System**: for non-ABAP systems a specific ABAP system acts as TMS Communication System (no special setup needed)
  - Transport Tool Parameter for non-ABAP systems in TMS: *COMMUNICATION_SYSTEM*
- **Transport Organizer Web UI**: the ABAP Web Dynpro Application *CTS_BROWSER* is activated (Transaction **`SICF`**) for a specific client (e.g. 001)
  - Transport Tool Parameter for source systems: *NON_ABAP_WBO_CLIENT*; client for which the application is activated and should run (e.g. 001). This parameter defines the system for which transport requests are created. (see 3.2.1).
- **Deploy Web Service Client**: (see Figure 2: Deploy Web Service Client Configuration)
  - HTTP Connection: the HTTP Connection (recommended name: **`CTSDEPLOY`**) for the connection to the Java stack is created (transaction `SM59` – client independent)
  - Logical Port: the Logical Port (recommended name: **`CTSDEPLOY`**) to the Java Web Service *DeployProxy* is created (transaction **`lpconfig`** – client 000)
- **Transport Directory**: see transaction **`AL11`** – value of directory parameter *DIR_TRANS*

For a better understanding on how things work together, the following picture “Figure 2: Deploy Web Service Client Configuration” shows an example of a HTTP Connection and Logical Port configuration.

For more details on how to configure the CTS+ components on the ABAP stack see the standard documentation.

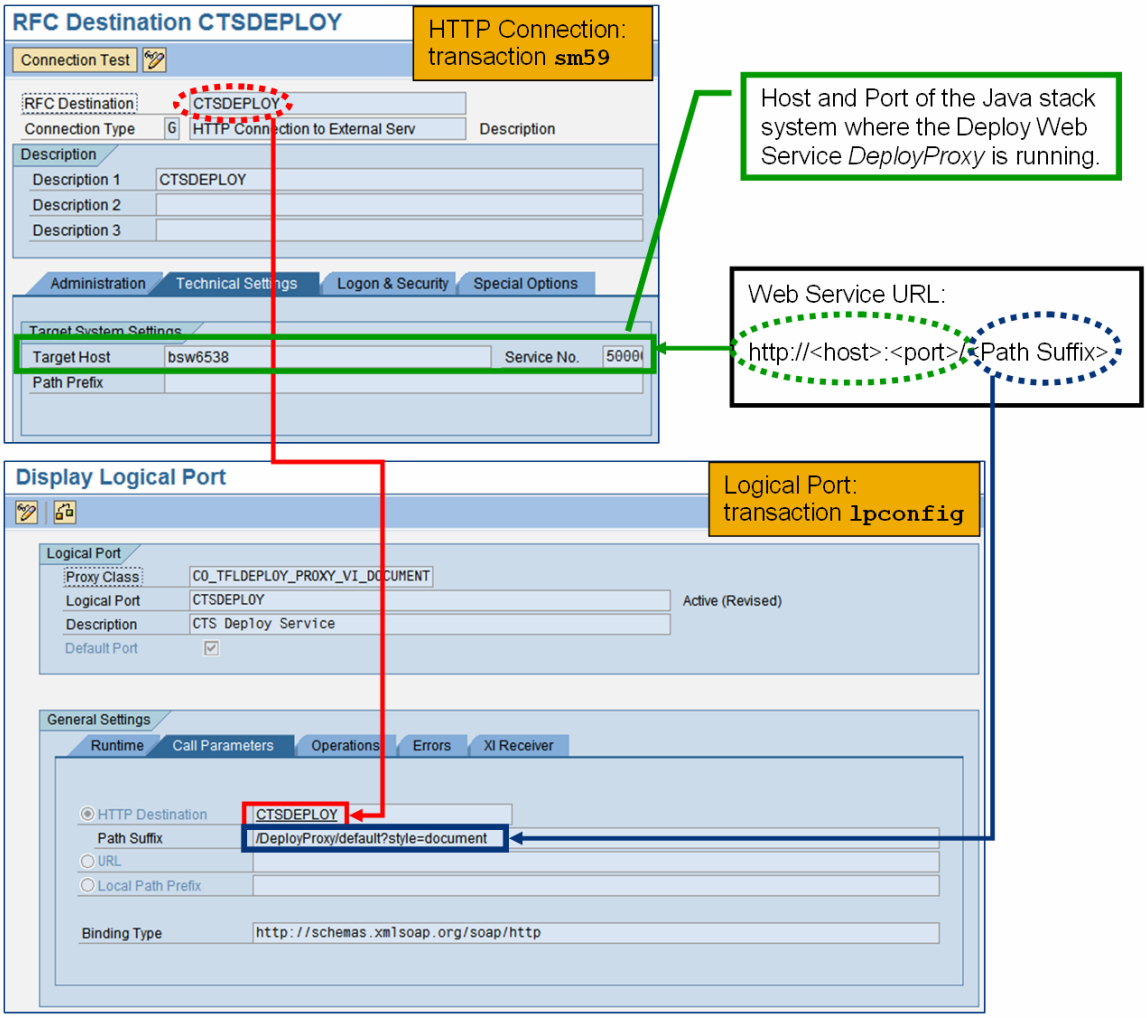


Figure 2: Deploy Web Service Client Configuration

> **Important**
>
> In the standard documentation the configuration is described in the way that the Path Prefix of the Web Service URL is entered in the RFC Destination instead of the Path Suffix in the Logical Port. Both configurations are possible but the future releases will recommend and describe the configuration as shown in Figure 2: Deploy Web Service Client Configuration.

3.1.2 Setup CTS+ Server components: JAVA stack

- **Deploy Web Service**: the Web Service *DeployProxy* is available (go to *Web Service Navigator*)

  > **Note:** See note 1076189 if the Web Service does not exist on the Java stack (the Web Service is part of the SCA DI_CMS which needs to be deployed on the Java stack of a CTS+ system)

3.1.3 Setup CTS+ Server components: Separate stacks

If the ABAP and Java stacks are implemented on separate systems that run on different hosts, some additional configuration aspects need to be considered. The ABAP and Java stack configuration is exactly the same as described before. Make sure, that the HTTP connection (Target Host and Service No.) points to the Java system and that the logon is configured correctly.

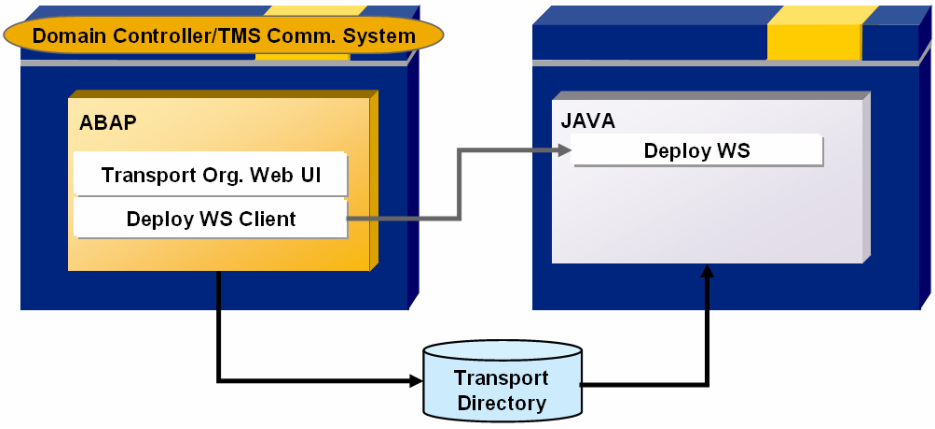


Figure 3: CTS+ components – separate stack configuration

As you see above in “Figure 3: CTS+ components – separate stack configuration” the Java stack needs read access to the Transport Directory. Therefore a share or mount has to be created, so that the transport directory is accessible from both stacks. In addition a Transport Tool parameter for all non-ABAP target systems is needed with the name *DEPLOY_DATA_SHARE*. The value for this parameter is the actual path under which the Java stack can access the transport directory.

Note

The value for the *DEPLOY_DATA_SHARE* parameter can differ to the value for the profile parameter *DIR_TRANS* in the ABAP stack, in case it is a share or mount point.

3.2 TMS System Landscape Configuration

The following chapter explains the steps to setup CTS+ enabled systems and system landscapes in TMS. After creating the transport systems with their transport routes, the tight integration of CTS+ in the applications of the source system has to be configured as well. See chapter 3.3 for more details.

With the enhanced CTS a new type of TMS/CTS System is introduced – the non-ABAP System. Figure 4 shows the menu entry on how to create such non-ABAP Systems: Transaction *stms → SAP System → Create → Non-ABAP System.* This system type is needed for all pure AS Java landscapes for example EP, JAVA or SLD.

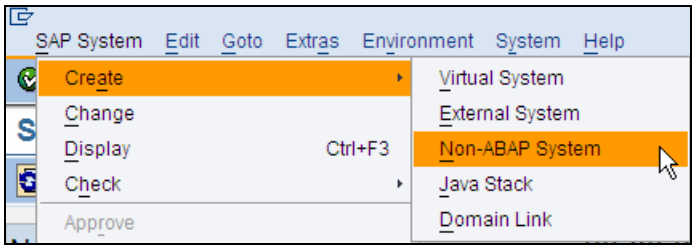


Figure 4: Transaction stms → SAP System → Create → Non-ABAP System

Furthermore new Transport Tool parameters are needed for non-ABAP systems to enable the transport of non-ABAP objects. These parameters are automatically added when using the wizard – see Figure 6 and Figure 7 and detailed description in the corresponding chapter. Transport Tool parameters (short: tp parameters) are defined in transaction *stms → System Overview → Change TMS Configuration: System XYZ → tab Transport Tool.*

Existing systems in the transport landscape that are dual-Stack systems (e.g. use case PI) can now be enhanced with the new Transport Tool parameters for the Java stack to be able to transport non-ABAP objects. Typically the ABAP stacks are already configured and included in the transport domain. To add the CTS+ functionality for transports to the Java stack, the existing configuration for the ABAP systems needs to be extended with the relevant CTS+ parameters. Since SAP NetWeaver 7.0 SP Stack 14 a wizard can be used to add the relevant parameters to existing ABAP system configurations (see Figure 5: Transaction STMS → Create →Java Stack).

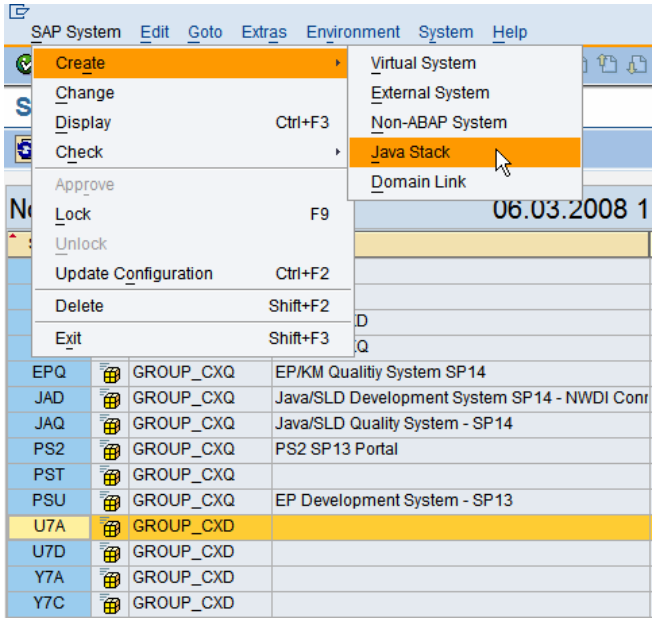


Figure 5: Transaction STMS → Create →Java Stack

3.2.1 Configuration of Source Systems

To create a non-ABAP source system, the wizard shown in Figure 6 can be used. Enter the System ID and description and choose the Source System Settings. Typically a system is either a source system or a target system – in rare cases it is both (e.g. repair scenario for QA systems). With the Source System Settings the SAP System ID (SAPSID) of the Communication System and the client on which the Transport Organizer Web UI should run is defined. With the activation of the Transport Organizer for this system, transport requests for that system can be created. The transport requests will have names starting with the SAPSID of this system.

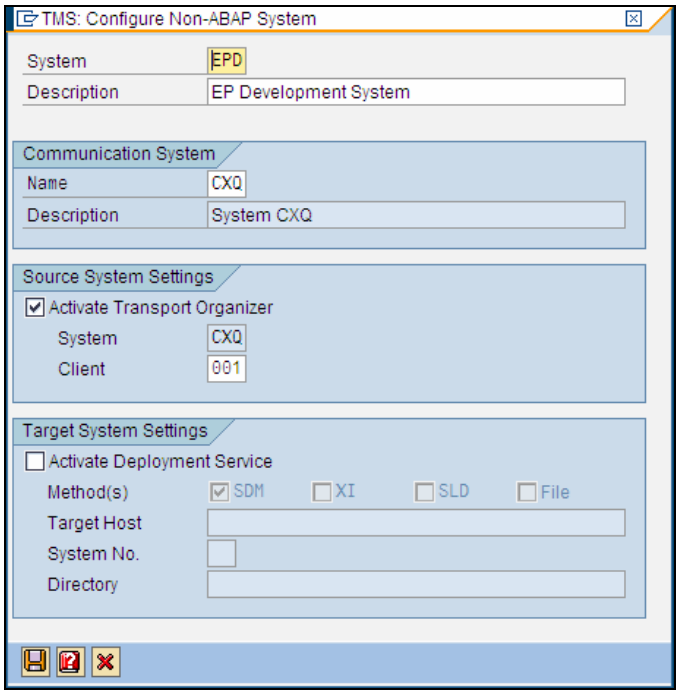


Figure 6: Create non-ABAP Source System

The following parameters are mandatory and automatically added when using the wizard for the configuration of a pure non-ABAP source system:

- *COMMUNICATION_SYSTEM*: SAPSID of the ABAP communication system (e.g. the domain controller)
- *NON_ABAP_SYSTEM*: 1
- *NON_ABAP_WBO_CLIENT*: Client of the ABAP stack on which the Transport Organizer Web UI (CTS_BROWSER) is activated and will run.

For dual-stack source systems, the following Java parameters are added using the wizard shown in Figure 5: Transaction STMS → Create →Java Stack. The difference of the two wizards is, that for adding Java stack parameters an existing system is selected and no new system is created as in Figure 6: Create non-ABAP Source System. Furthermore no *COMMUNICATION_SYSTEM* parameter is needed for dual-stack systems, as every ABAP system has its own tp that manages transports.

- *NON_ABAP_WBO_CLIENT*: Client of the ABAP stack on which the Transport Organizer Web UI (CTS_BROWSER) is activated and will run.

> **Note**
>
> All clients set as value for parameter *NON_ABAP_WBO_CLIENT* need the system change option (transaction **`se03`** → *Administration* → *Set System Change Option*) set to “Automatic recording of changes”.
>
> In chapter 3.3 additional strategy parameters (*WBO_GET_REQ_STRATEGY* and *WBO_REL_REQ_STRATEGY*) for source systems are introduced that are important for the tight application integration.

3.2.2 Configuration of Target Systems

To create a non-ABAP target system, the wizard shown in Figure 7: Create non-ABAP Target System can be used. Enter the System ID and description and choose the Target System Settings. Typically a system is either a source system or a target system – in rare cases it is both (e.g. repair scenario for QA systems). The Communication System for a non-ABAP target system has the task to communicate with the transport control program *tp* running on that system together with the *Deploy Web Service Client* to trigger the import on the target runtime system. With the Target System Settings the types of import are specified.

The enhancements for CTS support the following import types for non-ABAP objects:

- Imports via SDM/Deploy Controller (Use Cases: EP, KM, NWDI SCAs, SDAs, EARs):
  - tp Parameter: *DEPLOY_URL*
- Imports for PI/XI objects:
  - tp Parameter: *DEPLOY_XI_URL*
- Imports of SLD objects:
  - tp Parameter: *DEPLOY_SLD_URL*
- File transfer:
  - tp Parameter: *DEPLOY_OUTBOX*
  -

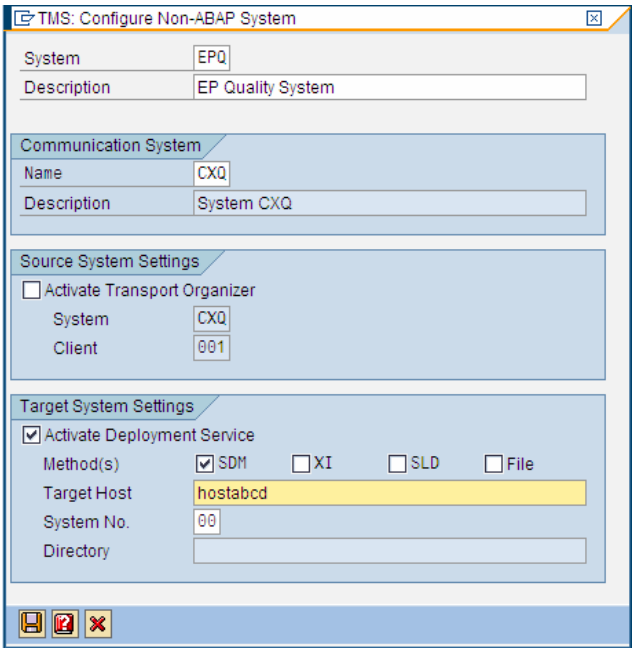


Figure 7: Create non-ABAP Target System

The following parameters are mandatory for the configuration of pure non-ABAP target system:

- *COMMUNICATION_SYSTEM*: SAPSID of the ABAP communication system
- *DEPLOY_DATA_SHARE: (shared)* transport directory (e.g. <DIR_TRANS>\data)
- *DEPLOY_URL* and/or *DEPLOY_SLD_URL* and/or *DEPLOY_XI_URL* and or *DEPLOY_OUTBOX*
- *DEPLOY_WEB_SERVICE*: **`CTSDEPLOY`** (Name of the Logical Port) that is created to connect to the Java Web Service *DeployProxy*
- NON_ABAP_SYSTEM: **`1`**

  Note

  The tp parameter *CTS_SYSTEM (*value=**`Java`** or value=**`others`** for pure file system deployment) is only needed when using CTS+ with Solution Manager Change Request Manager (ChaRM).

For dual-stack target systems, the following Java parameters are added using the wizard shown in Figure 5: Transaction STMS → Create →Java Stack:

- *DEPLOY_DATA_SHARE: (shared)* transport directory (e.g. <DIR_TRANS>\data)
- *DEPLOY_URL* and/or *DEPLOY_SLD_URL* and/or *DEPLOY_XI_URL* and or *DEPLOY_OUTBOX*
- *DEPLOY_WEB_SERVICE*: **`CTSDEPLOY`** (Name of the Logical Port) that is created to connect to the Java Web Service *DeployProxy)*

3.2.3 Configuration of Transport Routes

Now the transport routes for the non-ABAP system landscape can be configured as described in the standard documentation. No additional or CTS+ specific configuration is needed.

3.3 CTS+ Application Integration (Close Coupling)

The take advantage of all CTS+ features the recommended scenario is to enable the tight integration of CTS+ into the development systems or design time application which is also called “Close Coupling”. Without Close/Tight Coupling CTS+ can be used in a Loose Coupling scenario, where the Transport Organizer Web UI is used standalone. Here all steps that are transport steps handled by developers or content creators are performed manually: creating transport requests, attaching non-ABAP objects and releasing transport requests. After a transport request is released, it is the task of a transport administrator to trigger the imports in TMS to the corresponding target systems in the transport route.

With the tight integration the CTS+ developer tasks can be performed within the design time application of the development/source system, without changing the tools. The transport process consists of getting/creating a transport request for the corresponding source system, attaching the non-ABAP object files and releasing the transport request. The CTS+ integration supports developers in the following way:

- **Creation of transport requests**: with close coupling the strategy parameter *WBO_GET_REQ_STRATEGY* can be set to **`smart`** so that a transport request is automatically created (if no standard request is already available)

- **Attachment of non-ABAP objects**: with close coupling the non-ABAP object files are automatically attached to the transport request. Since SP Stack 14 the standard file transfer method is using file streaming. Parameter *WBO_FILE_TRANSFER* can be set to use mount/share instead.
- **Release of transport requests**: with close coupling the strategy parameter *WBO_REL_REQ_STRATEGY* can be set to auto so that the transport request is automatically released after export and attach.

To enable the tight integration a configuration step in the Visual Administrator of the source runtime system is needed. In all cases except NWDI the tight integration is done in the development system (with is the TMS source system), where the content is created (e.g. EP or PI DEV system). For NWDI the NWDI server itself is the system that is tightly integrated with CTS+. Here the assembled software component is attached to a transport request. Please have a look at the standard documentation to see how to create a new RFC destination called **`sap.com/com.sap.tc.di.CTSserver`** in the Destination Service of the Visual Administrator.

Example

See chapter 4.3.2 for an system landscape example.

3.3.1 Example – Use Case EP

You have a three system portal landscape on NW 7.0 SP14 or higher: EPD, EPQ and EPP. For the tight integration of the development portal (source system) to CTS+ (close coupling), you need to configure that Portal source system: here EPD.

Go to the Visual Administrator of that system and add the RFC destination as explained in the standard documentation (System, System Number and System ID of the CTS+ system). Here the use case EP is explained, but the configuration is generic for all other use cases except of NWDI (see 3.3.2).

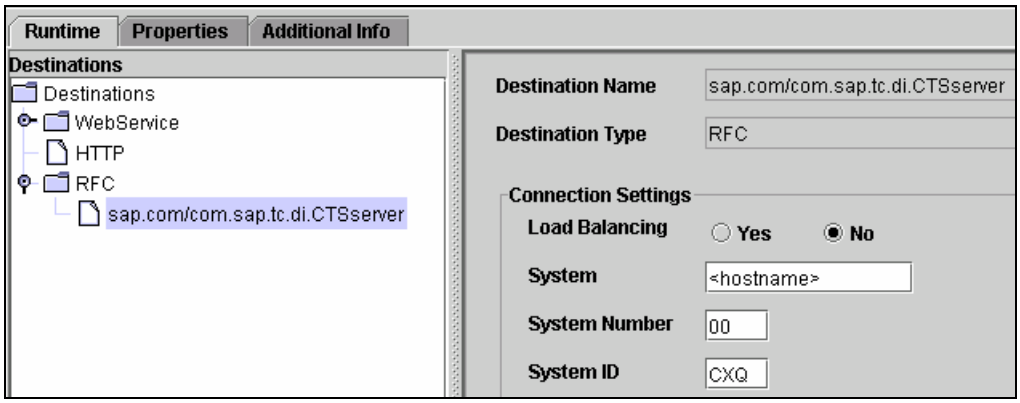


Figure 8: Visual Administrator → Destinations → RFC

Important

The standard configuration of the tight integration for use case EP, PI, SLD, etc. is implemented in that way, that a non-ABAP source system has to be configured in TMS with the real SAPSID as System ID (here: EPD).

3.3.2 Example – Use Case NWDI

You have a three system Java landscape: JAD, JAQ and JAP. The development infrastructure for the Java based developments for these systems is NWDI (SID: DII) on NW 7.0 SP14 or higher. The typical landscape design looks like this. The development system JAD is added as runtime system in the NWDI track. Furthermore a non-ABAP source system JAD is configured in TMS with a consolidation route to the non-ABAP target system JAQ and a delivery route to the non-ABAP target system JAP.

To configure the tight integration of NWDI to CTS+ you need to go to the Visual Administrator of the NWDI server and add the RFC destination as explained in the standard documentation (same as explained in 3.3.1). There is no need for configuration in the Java development system JAD. Within the NWDI track you need to configure that track for CTS usage and add the CTS upload system JAD. Go to NWDI – *Change Management Service* → *Landscape Configurator* → *Track Data*. Check the checkbox "*Connected to CTS*" and enter the SID of the non-ABAP TMS source system as shown in Figure 9: NWDI track configuration (here: EPD). Every track can be connected to a different CTS upload system and therefore to a different transport route in TMS.

Note

Please see notes for changes between SPs:

- 1145268: Changes from SP12 to SP13
- 1146170: Changes from SP13 to SP14

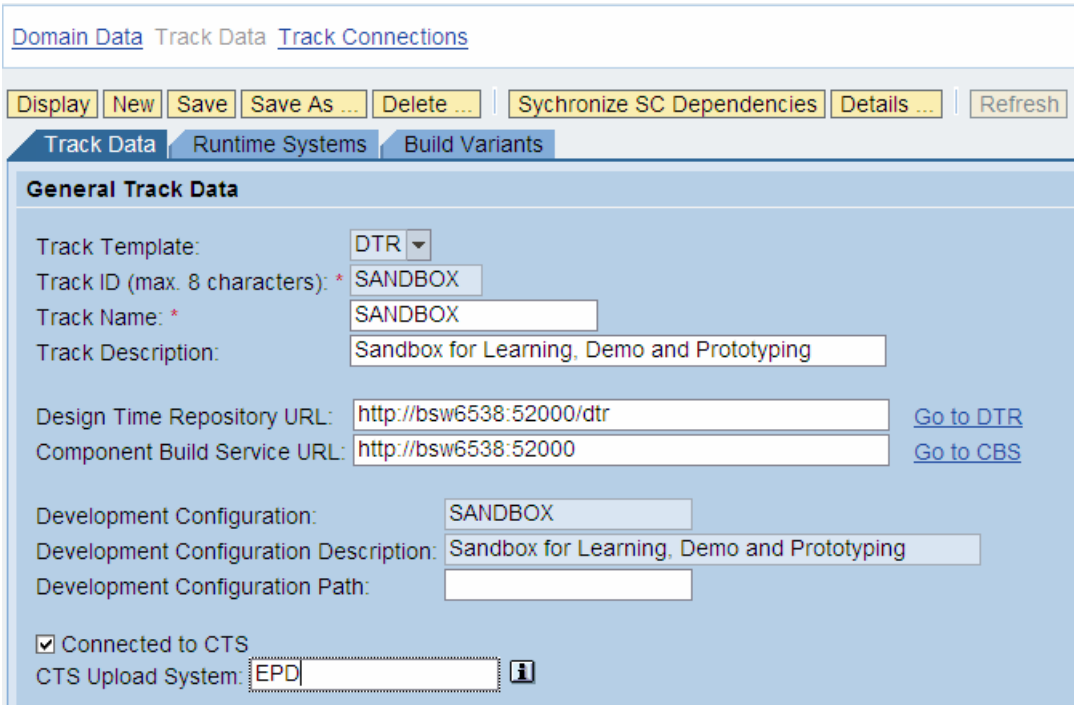


Figure 9: NWDI track configuration

3.3.3 Matrix

The tight application or development environment integration in CTS+ can only be used when the system requirements for the CTS+ system and the development system (source system) is met.

The following matrix should give you an overview which SAP NetWeaver SP level is needed for close coupling, how to configure it and what limits exist when using a combination of SP levels (e.g. CTS+ on SAP NetWeaver SP 14 with EP on SAP NetWeaver SP13).

As described in chapter 3.1 the components for the tight integration are the Export Client on the source system side and the Transport Organizer server application on the ABAP CTS+ system side. Therefore the best combination is always to have both systems on the same SP level. If that is not possible and not the case in your landscape, some side affects ore limitations need to be considered. These are listed in the tables below per use case.

Use Case EP:

CTS+ / EP	NW 7.0 SP12	NW 7.0 SP13	>= NW 7.0 SP14
NW 6.40 >= SP9 NW 7.0 all	Loose Coupling Transport Org. Web UI: SODIS_CORE_WBO	Loose Coupling Transport Org. Web UI: CTS_BROWSER	Loose Coupling Transport Org. Web UI: CTS_BROWSER
NW 7.0 SP13	Loose Coupling Transport Org. Web UI: SODIS_CORE_WBO	Close Coupling Configuration: Configuration Adapter Transport Org. Web UI: CTS_BROWSER	Close Coupling Configuration: Configuration Adapter Transport Org. Web UI: CTS_BROWSER 2
>= NW 7.0 SP14	Loose Coupling Transport Org. Web UI: SODIS_CORE_WBO	Close Coupling Configuration: Destination Service Transport Org. Web UI: CTS_BROWSER 1	Close Coupling Configuration:Destination Service Transport Org. Web UI: CTS_BROWSER

1. CTS+ NW 7.0 SP13 & EP NW 7.0 SP14
   - Definition of Share between EP and CTS+ Server needed → parameter tp: N*ON_ABAP_WBO_INBOX*
   - Links to Transport Organizer visible in the UI, but not supported. To release a transport request open the browser with the URL of the Transport Organizer Web UI for your development system or integrate it as an iView.
2. CTS+ NW 7.0 SP14 & EP NW 7.0 SP13
   - Definition of Share between EP and CTS+ Server needed needed → parameter tp: N*ON_ABAP_WBO_INBOX*
   - Links to Transport Organizer not available in the UI. To release a transport request open the browser with the URL of the Transport Organizer Web UI for your development system or integrate it as an iView.

Use Case PI:

CTS+ / XI	NW 7.0 SP12	NW 7.0 SP13	NW 7.0 SP14
NW 6.40 >= SP14 NW 7.0 all	Loose Coupling Transport Organizer: SODIS_CORE_WBO		
NW 7.0 SP13		Close Coupling Configuration: Configuration Adapter Transport Organizer: CTS_BROWSER	
NW 7.0 SP14			Close Coupling Configuration: Destination Service Transport Organizer: CTS_BROWSER

As every PI system is a system with CTS+ functionality, there are no combinations of CTS+ system and PI system. For central administration of transports to PI systems, the Solution Manager system can be used. Even so the configuration and setup of a PI landscape is done in the domain controller of the PI landscape (which can be the Solution Manager System as well - see chapter 4.1 for more details).

Use Case NWDI:

CTS+ / NWDI	NW 7.0 SP12	NW 7.0 SP13	NW 7.0 SP14
NW 6.40 all NW 7.0 all	Loose Coupling Transport Organizer: SODIS_CORE_WBO	Loose Coupling Transport Organizer: CTS_BROWSER	Loose Coupling Transport Organizer: CTS_BROWSER
NW 7.0 SP12 NW 7.0 SP13	Close Coupling Configuration: Configuration Adapter Transport Organizer: SI_CORE_WBO_API	Close Coupling Configuration: Configuration Adapter Transport Organizer: SI_CORE_WBO_API Loose Coupling (1)	Close Coupling Configuration: Configuration Adapter Transport Organizer: SI_CORE_WBO_API Loose Coupling (2)
NW 7.0 SP14	Loose Coupling Transport Organizer: SODIS_CORE_WBO	Close Coupling Configuration: Destination Service Transport Organizer: CTS_BROWSER* (3)	Recommended: Close Coupling Configuration: Destination Service Transport Organizer: CTS_BROWSER

1. CTS+ NW 7.0 SP13 & NWDI NW SP12/SP13
   - Close Coupling NWDI uses the SI_CORE_WBO_API Web UI instead of the new generic CTS_BROWSER.

- Close Coupling uses only the Transport Requests of the Domain Controller (CTS+ behavior of SP12 version)
- Definition of Share between NWDI and CTS+ Server needed → parameter tp: N*ON_ABAP_WBO_INBOX*
- Recommendation for NWDI SP12/SP13: Loose Coupling

2. CTS+ NW 7.0 SP14 & NWDI NW SP12/SP13:
   - Not recommended scenario. Might not work as the old transport organizer application SI_CORE_WBO_API might not exist in CTS+ SP14 and higher.
   - Recommendation for NWDI SP12/SP13: Loose Coupling
   - Recommendation to update NWDI system to NW 7.0 SP14 or higher.
3. CTS+ NW 7.0 SP13 & NWDI NW SP14:
   - Links to Transport Organizer Web UI are included in the UI, but does not work. To release a transport request open the browser with the URL of the Transport Organizer Web UI for your development system.
   - Definition of Share between NWDI and CTS+ Server needed → parameter tp: N*ON_ABAP_WBO_INBOX*

4. Landscape Scenarios

This chapter explains how to configure the components for using CTS+ for different landscape scenarios. The general recommendation is to use the Solution Manager system as central CTS+ domain controller which corresponds to the scenarios described in 4.2.1, 4.3.2 and 4.3.3. A reasonable alternative is to setup the CTS+ components in existing systems, which is the recommended way within a PI landscape (see 4.2.2 and 4.3.1) and whenever the synchronization aspect of ABAP and non-ABAP transports is important (see 4.2.3.2). In case a Solution Manager system is not available, a separate system (with an ABAP and JAVA stack) that provides the CTS+ functionality can be used instead.

Before the landscape patterns in general and the combined landscape examples are introduced in chapter 4.2 and 4.3, the role of the SAP Solution Manager system as central transport management tool is explained.

4.1 Central Transport Management

The configuration and usage of CTS+ functionality is irrespective of the domain design. You can have a landscape with one domain as shown in Figure 10: Single Domain" or you can have one domain per landscape (see Figure 11: Multiple Domains").

Within each domain a dedicated system plays the role of a domain controller (DC). The domain controller is the system in which the system setup (e.g. creation of non-ABAP systems) is done and the transport routes are configured. This is done locally in each domain. Therefore it is important for the CTS+ usage, to have a domain controller that is CTS+ enabled (means: has the required release/SP stack level) in case that domain needs non-ABAP transports.

The Solution Manager system plays a central and crucial role within the change control management to coordinate the changes that are introduced to a system landscape. With the Solution Manager components Maintenance Optimizer and Change Request Management (ChaRM) for example the maintenance and change management can be controlled centrally for the complete system landscape across domains. Therefore it is obvious that the Solution Manager System can act as central transport management system (with or without ChaRM) from which all transports can be triggered within the landscape. To trigger transports across domains, the domains need to be linked via domain links as shown in Figure 11: Multiple Domains with Domain Links. This is described in the standard documentation.

> **Tip**
>
> Configuring Links between Transport Domains: http://help.sap.com/saphelp_nw70/helpdata/EN/a3/139838280c4f18e10000009b38f8cf/frameset.htm).

The Solution Manager system can be used in all scenarios that are described in the following chapters as central transport management system. That means that it is possible to trigger and control the imports centrally. The actual imports run on the target systems or on the systems that act as communication system for the non-ABAP target systems. That's not necessarily the domain controller or Solution Manger system.

The configuration of the CTS+ relevant parts (Transport Organizer Web UI, Deploy Web Service and Deploy Web Service Client) and the configuration of the source and target systems with the CTS+ relevant transport parameters depend on the system landscape and are explained in the next chapter.

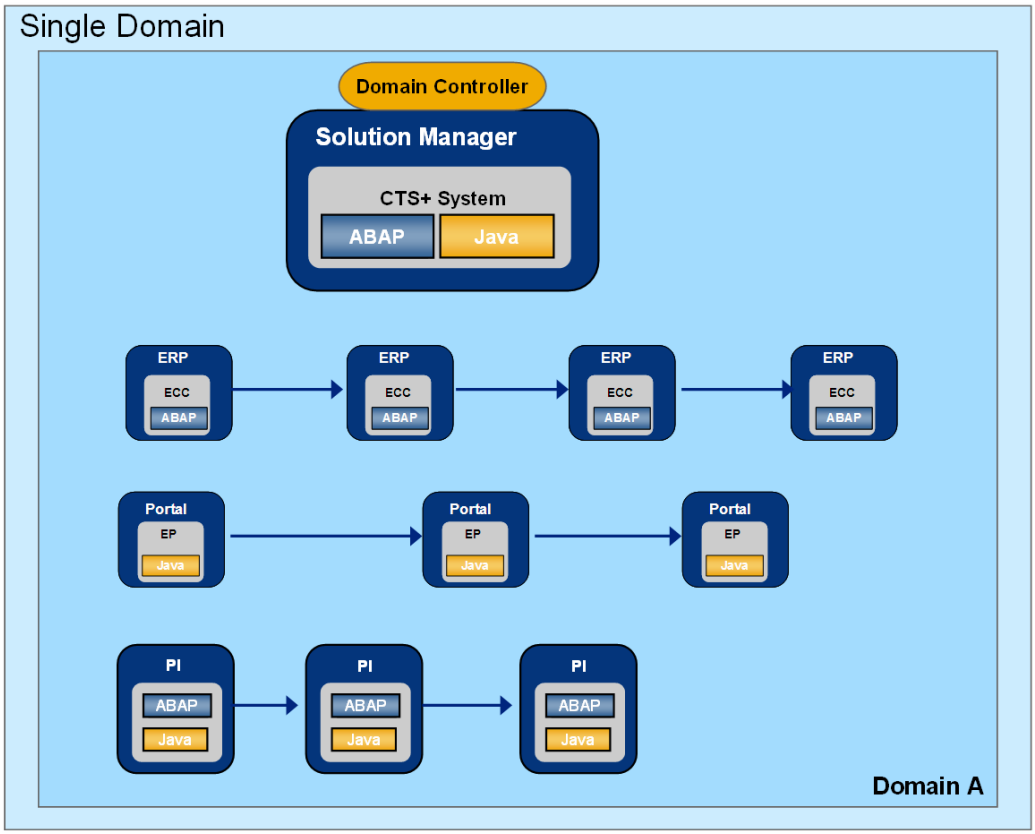


Figure 10: Single Domain

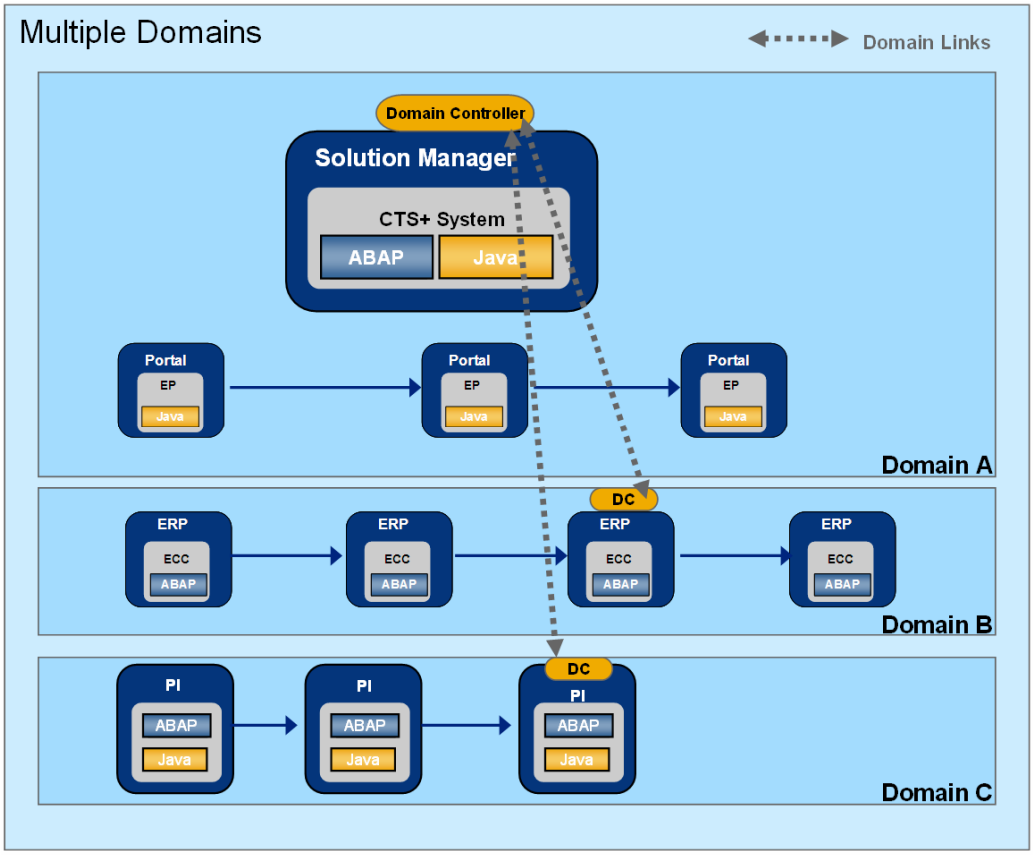


Figure 11: Multiple Domains with Domain Links

4.2 Landscape Patterns

4.2.1 Non-ABAP Landscape

The following section describes the configuration and setup of CTS+ for a pure non-ABAP system landscape, where the development and target systems are pure Java systems (e.g. EP, Java/J2EE with NWDI, KM use case).

The typical configuration for using CTS+ functionality for a pure non-ABAP landscape is to setup/use a separate CTS+ system (Example: SAPSID=DCS) that is configured as explained in chapter 3 (e.g. the Solution Manager system).

The landscape consists of at least one development system that acts as source system and one target system to which the changes are deployed.

Example

The following example describes the use case “Transport changes to Java runtime systems”. The example shows a three-system landscape with a development system (SAPSID: JDS), a quality system (SAPSID: JQS) and a production system (SAPSID: JPS). Here the use case “imports via SDM/Deploy Controller” is shown → *DEPLOY_URL*.

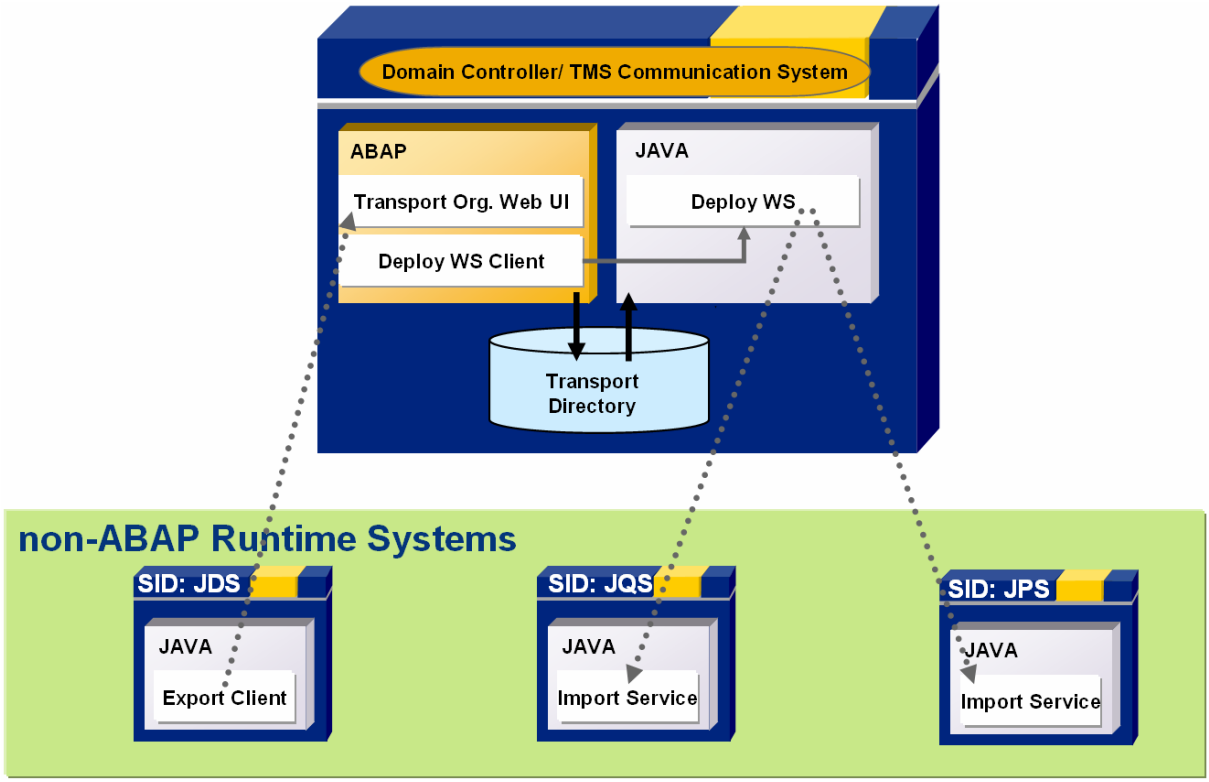


Figure 12: Example for non-ABAP system landscape

The system JDS is created as non-ABAP source system, JQS and JPS as non-ABAP target system in the domain controller system DCS. Below in Figure 13: Example for TMS configuration for non-ABAP landscape the TMS configuration for each system with the relevant *tp* parameters and the transport route is shown.

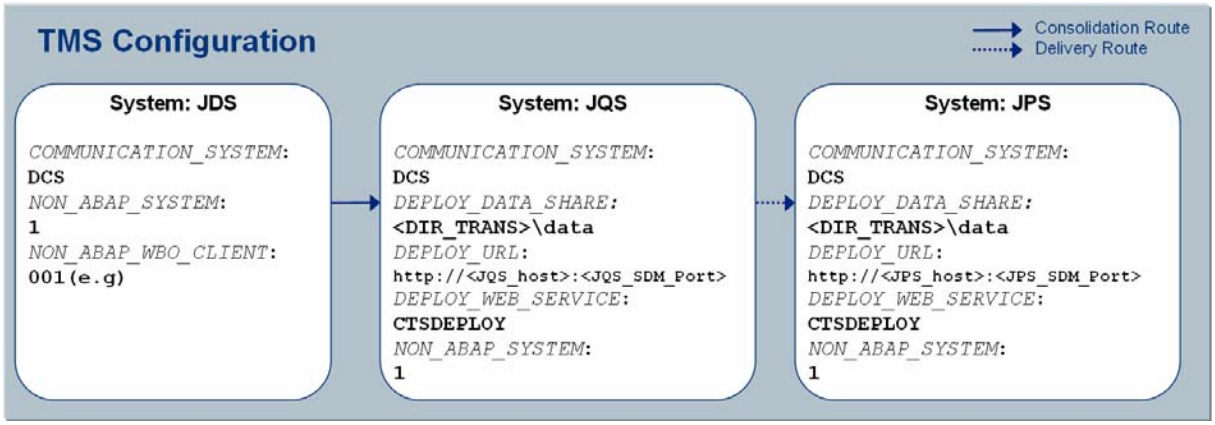


Figure 13: Example for TMS configuration for non-ABAP landscape

Add the SDM user and password for systems JQS and JPS using the wizard or via TMS menu.

4.2.2 Dual Stack Landscape

The following section describes the configuration and setup of CTS+ for a dual-stack system landscape, where the development and target systems are dual-stack systems (e.g. PI use case).

The typical configuration for using CTS+ functionality for a dual-stack landscape is to setup/use the existing ABAP transport domain controller configuration. For using CTS+ functionality the CTS+ components need to be configured in addition to the existing ABAP transport configuration as explained in chapter 2. The main difference to a non-ABAP system landscape is that each target system triggers the imports using the *tp* program running on that target system (same behavior as in ABAP). Having a dual-stack system landscape with a *tp* running on each system leads to the fact that the corresponding *Deploy Web Service (DEPLOYPROXY)* should run and should be configured for the Java stack part of that dual stack system. Therefore the corresponding Deploy Web Service Client consisting of the *HTTP Connection* and the *Logical Port* needs to be configured on the ABAP stacks of all target systems.

Example

The following example describes the use case “Transports of PI objects”. Here for the non-ABAP target systems the DEPLOY_XI_URL parameter is important.

The example shows a three-system landscape with a development system (SAPSID: PID), a quality system (SAPSID: PIQ) and a production system (SAPSID: PIP).

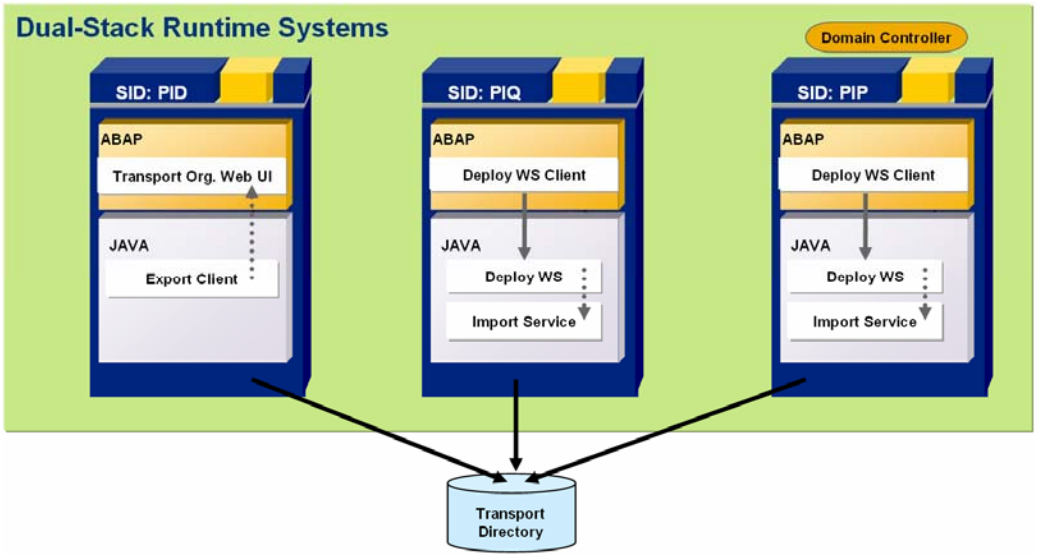


Figure 14: Example for dual-stack system landscape

> **Note**
>
> To use a common transport directory is recommended but not required. CTS+ can handle several transport directories as well. Please see TMS transport groups in the standard documentation.

> **Tip**
>
> If the quality system (here: PIQ) is used to repair bugs or do a hot fix, the Transport Organizer Web UI needs to be configured for that system as well. In that case the PIQ system is configured as source and target system which is supported by the wizard "Create Java Stack".

For system PID the Java stack source parameters are added. PIQ and PIP are target systems for which the Java stack target parameters are needed. The system PIP is configured as domain controller. Below the CTS+ relevant tp parameters are listed.

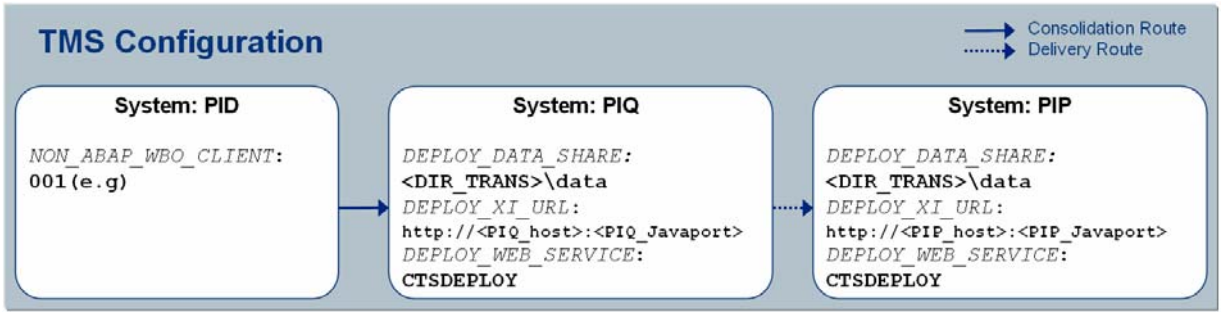


Figure 15: Example for TMS configuration for dual-Stack system landscape

Add the XI user and password for the systems PIQ and PIP using the wizard or via TMS menu.

4.2.3 Enhanced ABAP Landscape

The following section describes the configuration and setup of CTS+ for an existing ABAP landscape. This type of configuration is usually used, when an installation of a dual-stack system as CTS+ system is not possible and the existing ABAP TMS landscape should be enhanced to be capable of transporting non-ABAP objects as well (e.g. Portal or J2EE landscape).

There are several options how to configure the CTS+ components within such a landscape. These options are described in the following sections.

4.2.3.1 CTS+ Configuration on dedicated ABAP and Java System

To enhance an existing ABAP TMS landscape with CTS+ functionality the domain controller system is reused to setup and configure the non-ABAP systems and transport routes. One dedicated system, usually the domain controller system acts as communication system for all non-ABAP systems and here the *Transport Organizer Web UI* (CTS_BROWSER) is setup and activated for a specific client.

For the CTS+ functionality on the Java part, a dedicated Java system (usually the counterpart of the domain controller system, e.g. QAS) is used. For that system the *Deploy Web Service* has to be deployed (SCA DI_CMS). On the ABAP system that acts as TMS communication system for the Java system landscape the *Deploy Web Service Client* needs to be configured with a HTTP connection pointing to the Java stack and a Logical Port *CTSDEPLOY* with the Path Suffix as explained in 3.1.1.

Both systems need access to the transport directory, as shown in Figure 16: CTS+ enhanced ABAP landscape.

This is the recommended configuration when extending an existing ABAP TMS configuration with CTS+ functionality. Compared to the configuration explained in chapter 4.2.1, two dedicated systems (here: the quality systems ABAP and Java) act as central CTS+ system. Therefore it can be seen as a variant of the landscape pattern “non-ABAP Landscape” with one CTS+ enabled system. The CTS+ components are not distributed as in the dual-stack configuration. The drawback that could be considered for that scenario is that an import to other non-ABAP target systems in the transport route needs at least three systems up and running during import time – in our example Figure 16: CTS+ enhanced ABAP landscape: Import to JPS needs the systems AQS, JQS and JPS.

Example

The following example describes the use case “Transports of non-ABAP objects extending the existing ABAP TMS system landscape”. Here it is important to setup the CTS+ functionality on a dedicated ABAP system, usually the domain controller, and on a dedicated JAVA system as counterpart for that ABAP system.

The example shows a separate three-system landscape on ABAP and JAVA with two development systems (SAPSIDs: ADS and JDS), two quality systems (SAPSIDs: AQS and JQS) and two production system (SAPSIDs: APS and JPS). The following description concentrates on the configuration of the non-ABAP systems as the ABAP systems do not need any additional configuration except for the domain controller / communication system.

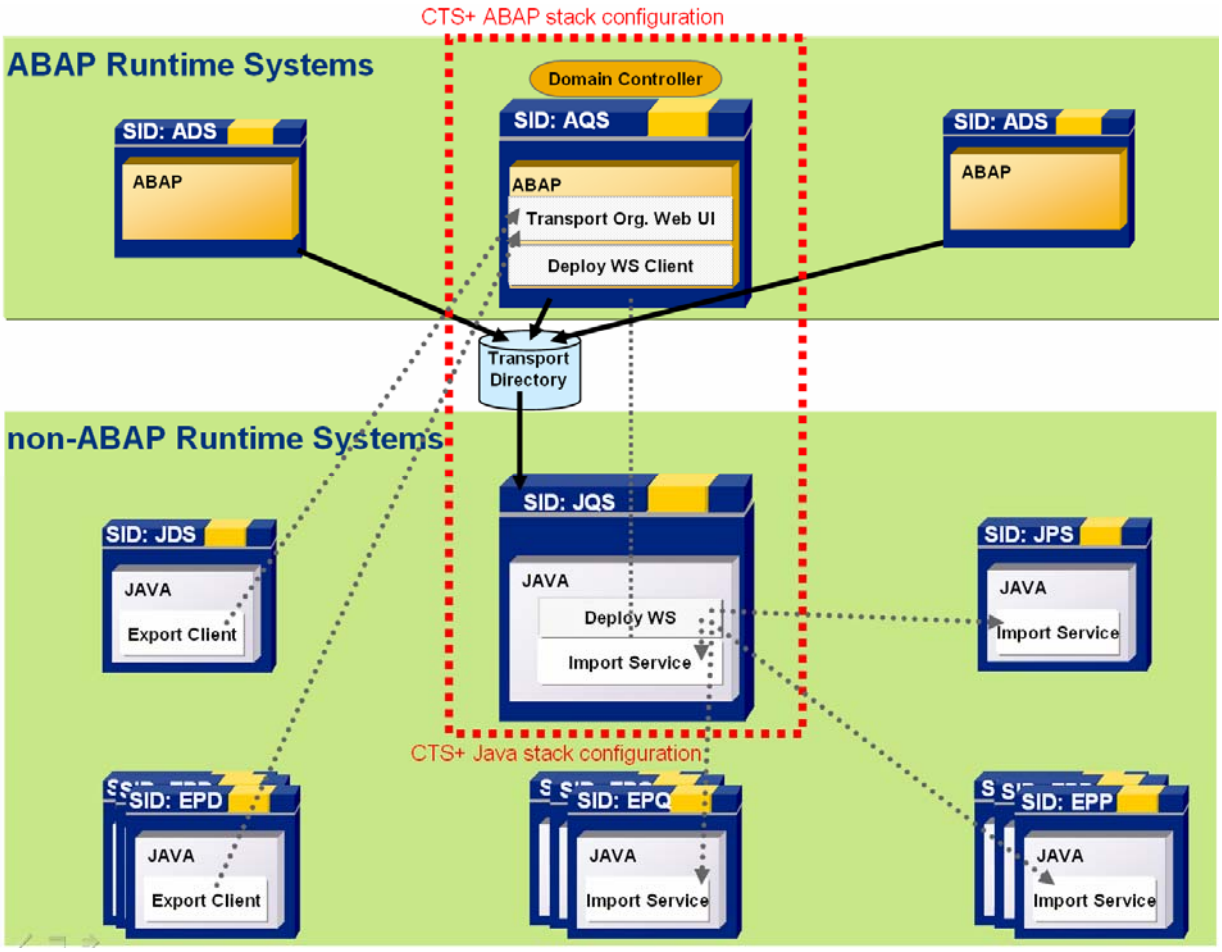


Figure 16: CTS+ enhanced ABAP landscape – Dedicated ABAP and Java system

The TMS configuration for the non-ABAP system landscape is similar to chapter 4.2.1 with the difference that the communication system is an ABAP only system that is configured as ABAP quality system in parallel. Furthermore the JAVA quality system act as Deploy Web Service system for all non-ABAP imports.

The following TMS configuration shows a system landscape for ABAP and a separate system landscape for Java with two transport routes. Using such a TMS configuration does not provide synchronization of ABAP and non-ABAP objects. A typical use case for such a configuration is to reuse the existing e.g. ERP ABAP landscape to be able to deliver e.g. JAVA/Portal content changes to the non-ABAP landscape as well. Here the changes of ABAP and the changes of the non-ABAP landscape have no relation to each other and the synchronization aspect is not important.

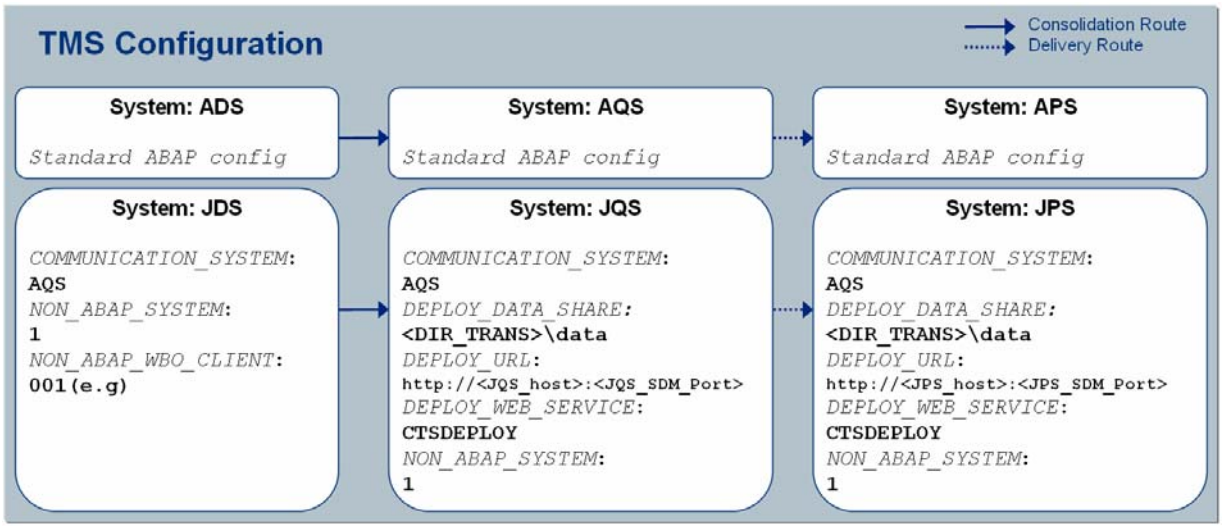


Figure 17: TMS Configuration – separate transport routes

Add the SDM user and password for system JQS and JPS using the wizard or via TMS menu.

4.2.3.2 Distributed CTS+ Configuration

The following chapter explains how to distribute the CTS+ configuration in case you want to limit the number of systems needed for the import steps. Furthermore the synchronization of ABAP and non-ABAP transports can be resolved using a distributed configuration. A typical use case for this scenario is the application ESS (Employee Self Service) of ERP where the UI is implemented on the Java stack and is connected to ECC ABAP backend systems.

The domain controller system is reused to set_up and configure the non-ABAP systems and transport routes (same as 4.2.3.1). This scenario can be seen as a variant of the landscape pattern “Dual Stack landscape” with the difference that the Java stacks run in separate systems.

- **Transport Organizer Web UI:** The Transport Organizer Web UI application (*CTS_BROWSER*) is typically setup and activated on the ABAP development system and this system acts as TMS communication system for the non-ABAP source system.
- **Deploy Web Service Client (ABAP) & Deploy Web Service (JAVA):** The Deploy Web Service Client is configured on each target system (e.g. QAS for ABAP) and for the corresponding target Java system the Deploy Web Service needs to be deployed. Whenever the CTS+ functionality is distributed, these two CTS+ components should be handled and configured as a pair. It is absolutely not recommended and supported to set_up more than one Deploy Web Service Client per communication system with different names that point to different CTS+ Deploy Web Services.

All target systems where the Deploy Web Service is deployed and used need access to the (global/shared) transport directory.

Example

Same landscape example as in 4.2.3.1. As the configuration is now distributed, the ABAP communication system is different for each non-ABAP system.

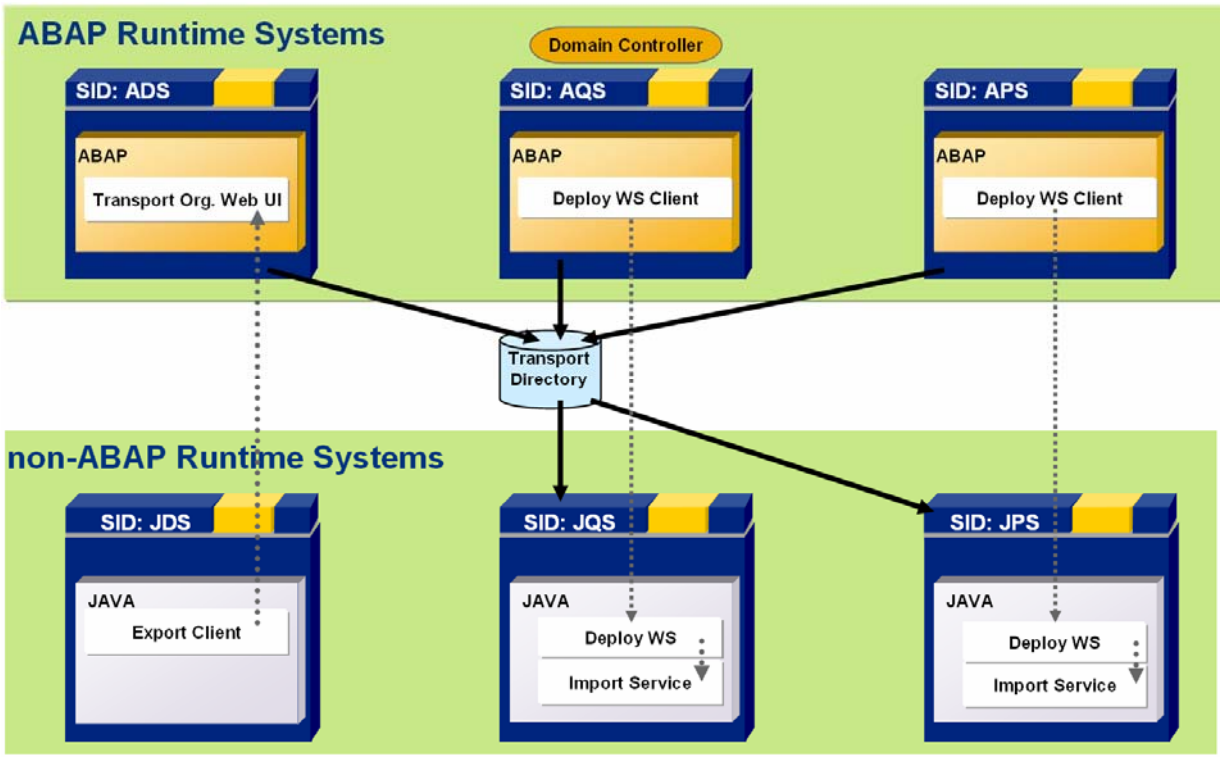


Figure 18: CTS+ enhanced ABAP landscape – Distributed Configuration

The following TMS configuration should explain how to enable the transport of non-ABAP objects along within the same transport route within one transport request. Here no non-ABAP TMS systems are configured and therefore, no separate transport route exists.

The existing ABAP systems are enhanced with the Java stack specific parameters. The additional parameters are needed to be able to attach non-ABAP objects in the ABAP transport request and to trigger the Deploy Web Service for the deployment of the non-ABAP objects. To get more details on how to configure the systems and which Java parameters are needed see Figure 19: TMS configuration – single transport route below. Here the existing ABAP configuration for system ADS, AQS and APS are extended with the JAVA stack parameters. The configuration is similar to a dual-stack configuration. The only difference is that the Java stacks run on a separate SID.

Important

When using a tight integration for example for the usage case EP, a TMS source system with the same SAPSID as the development system (e.g. DEV portal) is needed. In the configuration that is described here, the TMS source system for ABAP is reused to transport the non-ABAP changes together within one transport request. Therefore an additional Transport Tool parameter is needed with the name *CTS_SYSTEM_MAPPING*. Add this parameter in the configuration of the source system (here: ADS). The value of this parameter is the SAPSID of the Java development system (here: JDS).

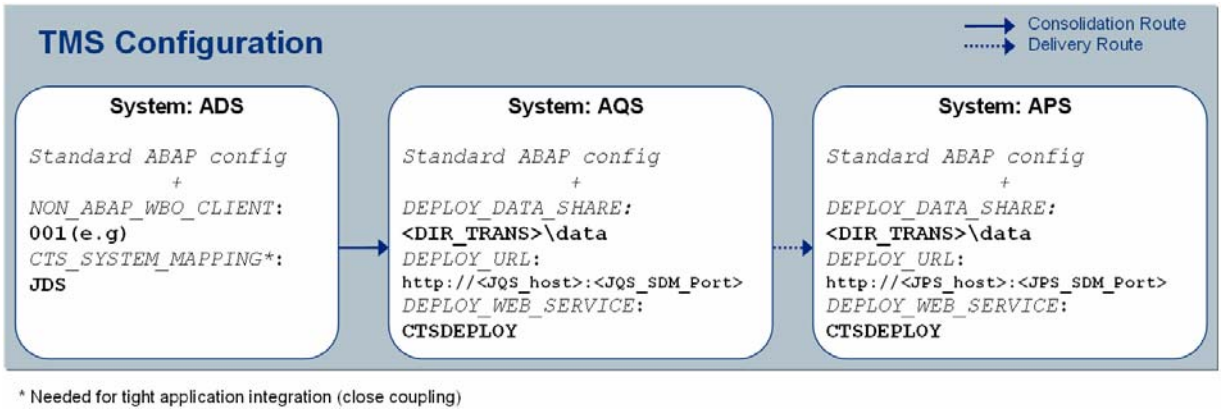


Figure 19: TMS configuration – single transport route

Add the SDM user and password for the runtime systems JQS in the system configuration AQS and for the runtime system JPS in the system configuration APS using the wizard or via TMS menu.

If the synchronization aspect is not important in your landscape scenario, you can use the following TMS configuration as well. Same as in Figure 17: TMS Configuration – separate transport routes - two separate transport routes one for non-ABAP and one for ABAP is configured, so that ABAP and non-ABAP objects can not be transported within one transport request. Here no *CTS_SYSTEM_MAPPING* parameter is needed for the tight application integration.

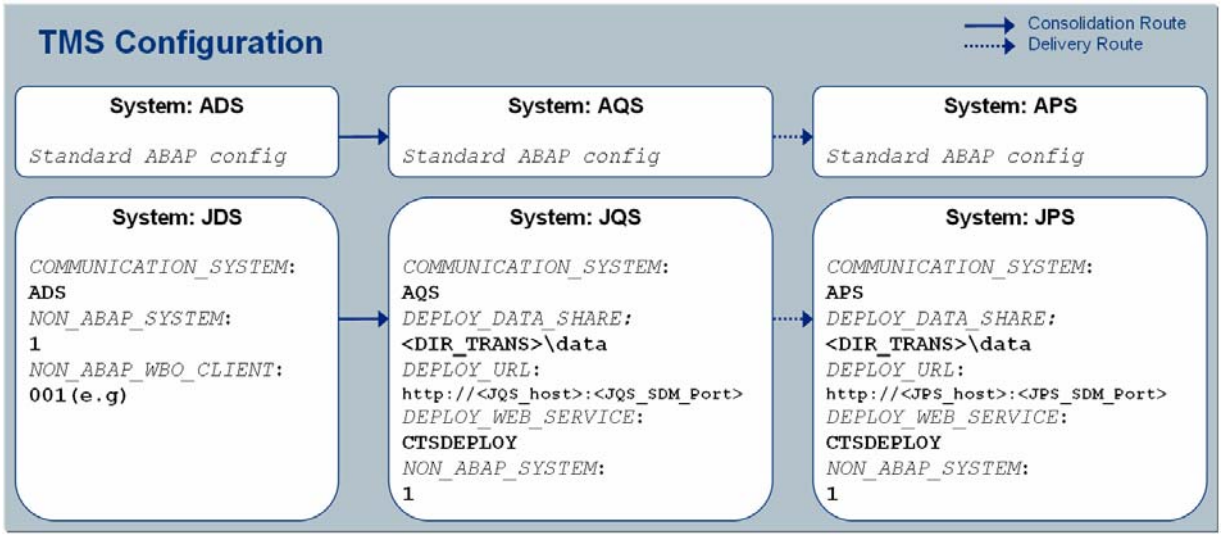


Figure 20: TMS Configuration - separate transport routes

Add the SDM user and password for the systems JQS and JPS using the wizard or via TMS menu.

4.3 Combined Landscape Scenarios

The following sections describe some scenarios, where different use cases are combined.

4.3.1 SLD and PI Landscape

For the PI transport scenario typically SLD transports are needed in combination. Depending on the existing landscape of the system landscape directory (SLD) of SAP NetWeaver the transport scenario looks different. The configuration options will be explained in the following.

For more information about SLD landscape setup and changes in SAP NetWeaver 7.1, see the documentation *Planning Guide – System Landscape Directory* available in SAP Developers Network at http://sdn.sap.com/irj/sdn/nw-sld.

4.3.1.1 Single productive SLD System

The first scenario shows a landscape with three PI runtime systems and just one central SLD productive system. Therefore no SLD CTS+ transport is required at all, as only one single productive SLD is in use.

For the TMS configuration of the PI systems see Figure 15: Example for TMS configuration for dual-Stack system landscape.

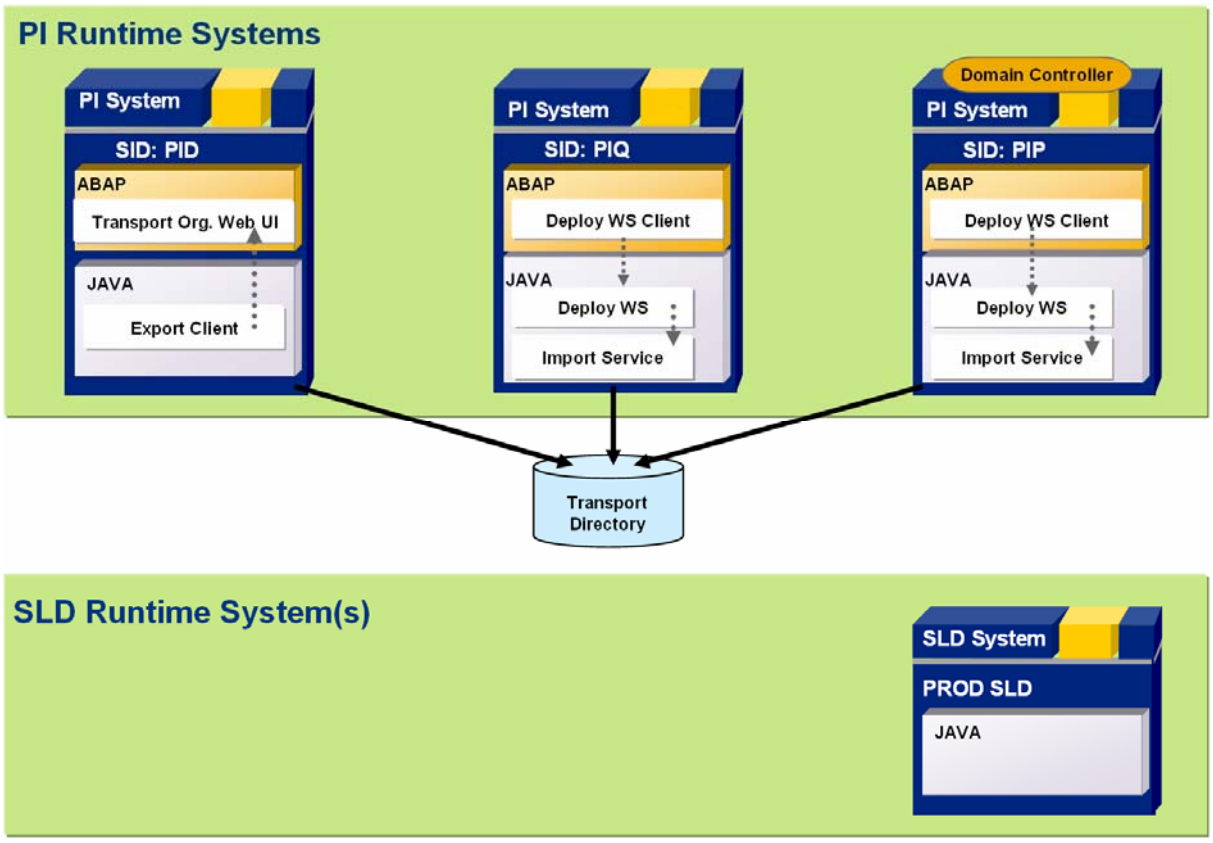


Figure 21: PI & SLD Landscape - Single PROD SLD System

4.3.1.2 Separate SLD Landscape

The following scenario describes a system landscape where the SLD transport landscape is completely independent and not coupled to the PI transports. The system that acts as CTS domain controller and/or the system that provides the components of the CTS+ functionality can be a separate system (e.g. the Solution Manager) or it can be reused within one domain in case a dual stack is in place. This use case corresponds to the landscape pattern 4.2.1 Non-ABAP Landscape.

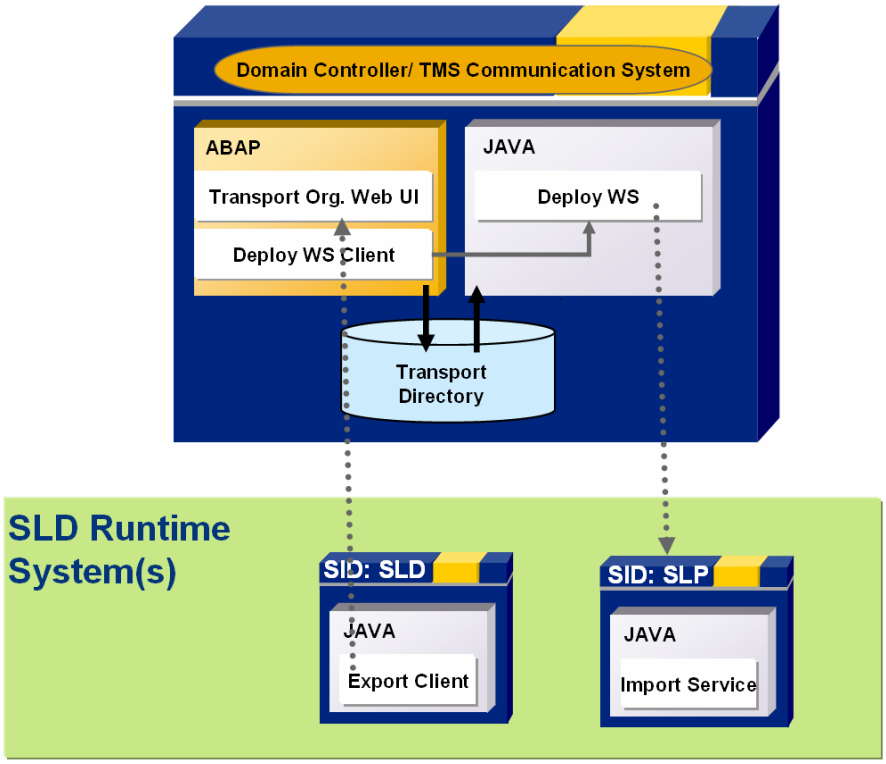
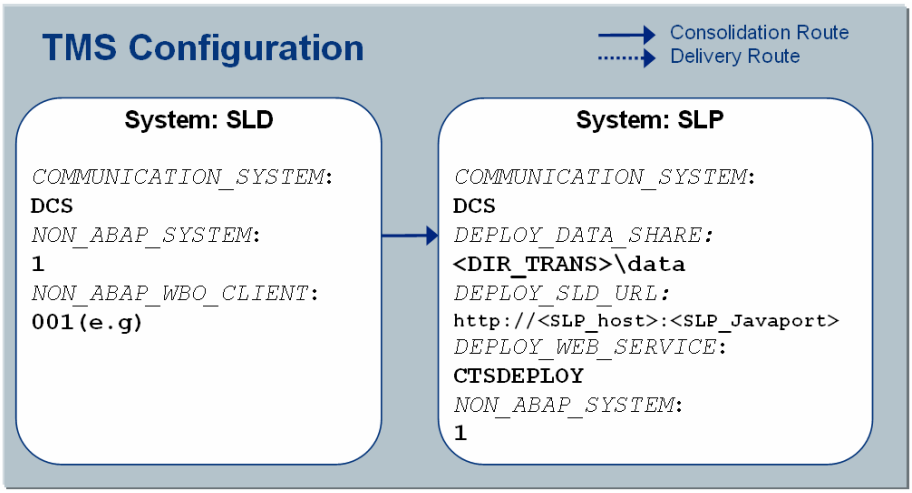


Figure 22: Separate SLD Landscape

The TMS configuration consists of one source system (SAPSID: SLD) and one target system (SAPSID: SLP). The configuration is similar to what is described in Figure 13: Example for TMS configuration for non-ABAP landscape. The only difference is that instead of using the DEPLOY_URL for SDM deployments, the DEPLOY_SLD_URL is used.



Add the SLD user and password for the systems SLP using the wizard or via TMS menu.

4.3.1.3 Combined PI and SLD System Landscape

The following scenario describes a system landscape where on every PI system a corresponding SLD is running and used.

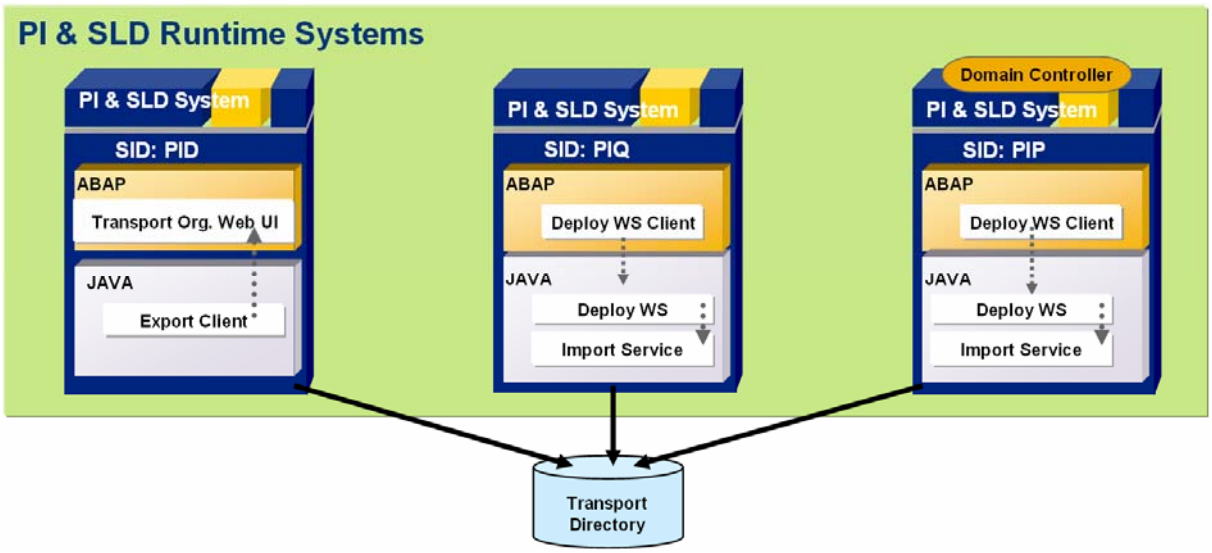
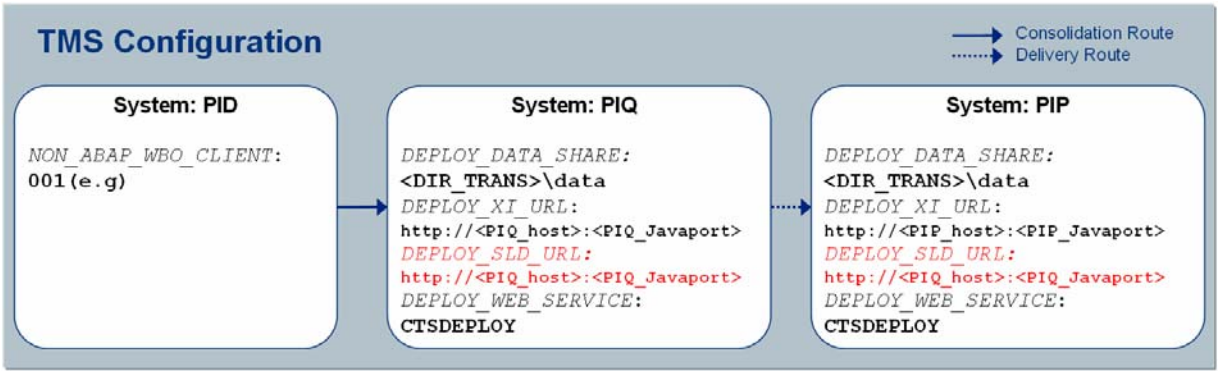


Figure 23: PI & SLD Landscape - Combined PI and SLD Systems

Here the TMS configuration is done as described in 4.2.2 – which is the PI use case. In addition every target system in the transport route needs the *DEPLOY_SLD_URL* parameter. The advantage of this scenario is that the SLD configuration objects can be transported together with the XI configuration objects within one transport request.



Add the XI and SLD user and password for the systems PIQ and PIP using the wizard or via TMS menu.

4.3.1.4 Separate PI and SLD System Landscape – One Transport

The following scenario describes a system landscape where each PI system has a corresponding SLD system, but compared to scenario 4.3.1.3 the SLD systems run on separate systems (SAPSIDs). Nevertheless it is possible to transport the SLD changes together with the PI changes. The configuration and transport route is based on the PI systems. This means no additional non-ABAP TMS systems are created for the SLD systems. Instead the TMS configuration for the PI systems is extended with the SLD deploy parameter so that the imports to the SLD target systems are triggered from the corresponding PI target system.

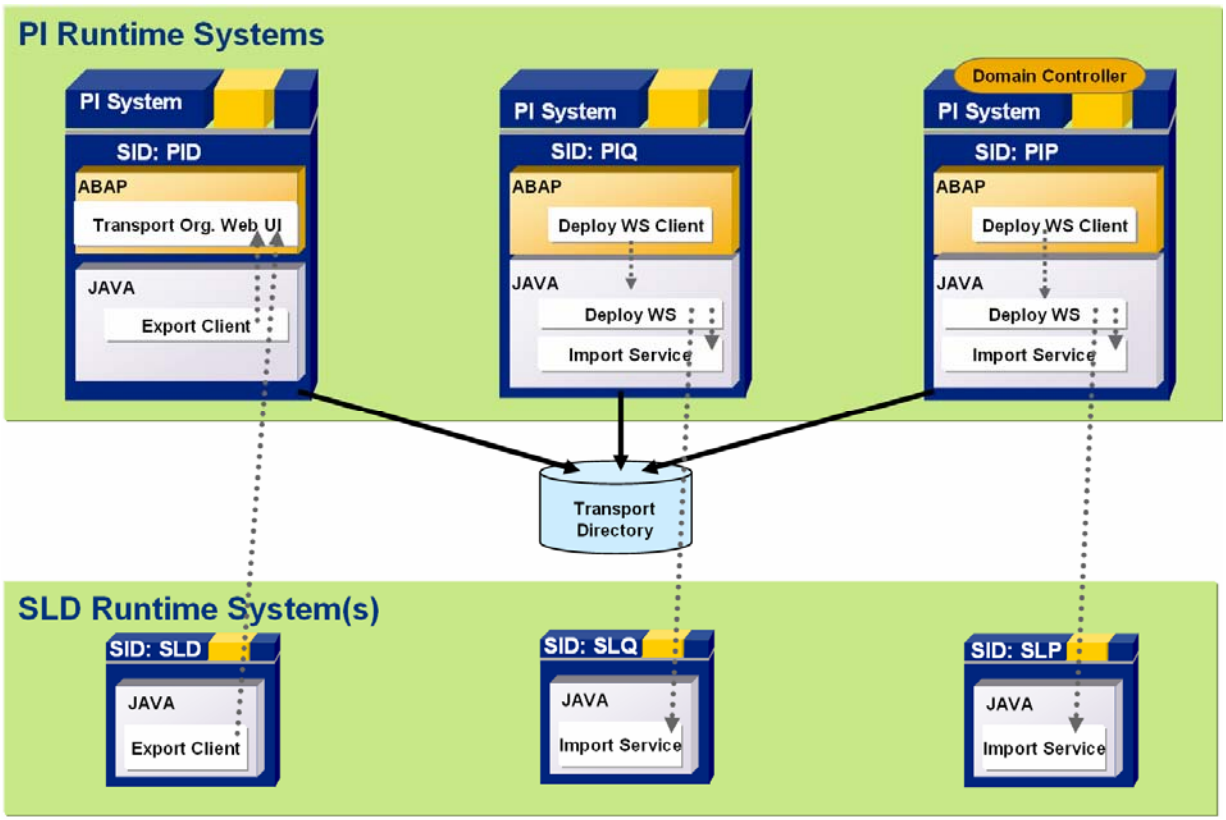
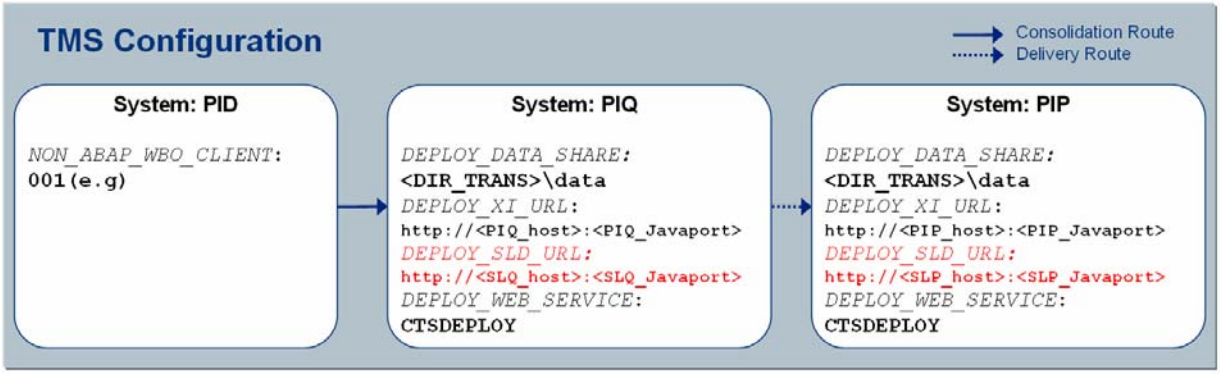


Figure 24: Separate PI and SLD System Landscape - one transport route

Here the TMS configuration is done as described in 4.3.1.3. The only difference is that the *DEPLOY_SLD_URL* points to the separate SLD systems. The advantage of this scenario is that the SLD configuration objects can be transported together with the XI configuration objects within one transport request.



Add the XI and SLD user and password for the systems PIQ/SLQ and PIP/SLP using the wizard or via TMS menu.

4.3.1.5 Combined PI and SLD System Landscape – separate Productive Systems

The following scenario is similar to scenario 4.3.1.3 where on every PI System a corresponding SLD is running and used except for the productive system. For the productive system the SLD is running on a separate AS Java system.

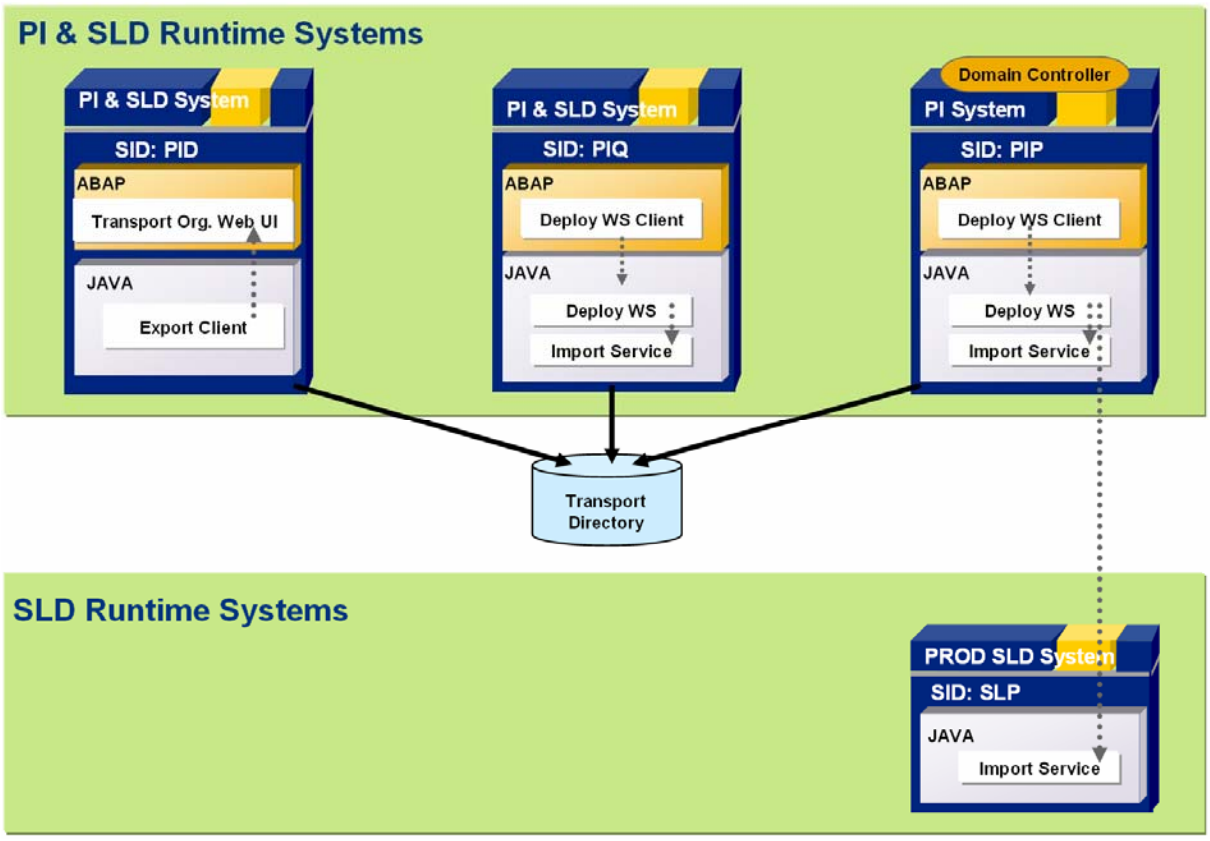
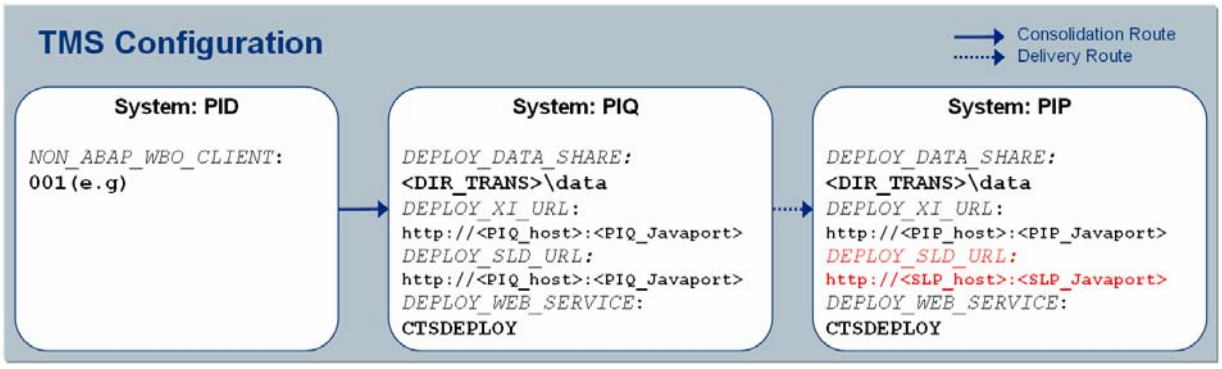


Figure 25: Combined PI & SLD Landscape - Separate PROD SLD System

The TMS configuration for this scenario is done exactly in the same way as described in the previous scenario 4.3.1.3. The only difference is that the *DEPLOY_SLD_URL* parameter for the productive system points to the SLD URL on the separate Java system. Also for this scenario where not all SLD systems are running on the same systems as PI, the transports of PI and SLD objects can be combined in one transport request.



Add the XI and SLD user and password for the systems PIQ and PIP using the wizard or via TMS menu.

4.3.1.6 Combined PI and SLD System Landscape – no QA SLD System

The following scenario is similar to scenario 4.3.1.5 but here no quality SLD exists. The configuration is done as described in 4.3.1.5 where the *DEPLOY_SLD_URL* points to the separate AS JAVA SLD system.

The difference here in this scenario is that no *DEPLOY_SLD_URL* will be configured for the QA system, as no corresponding SLD exists.

The PI systems for DEV and QA are connected against the same SLD – in that case the DEV SLD. The XI and SLD objects are created and recorded in the DEV system and can be transported together in one transport request to the QA and PROD systems.

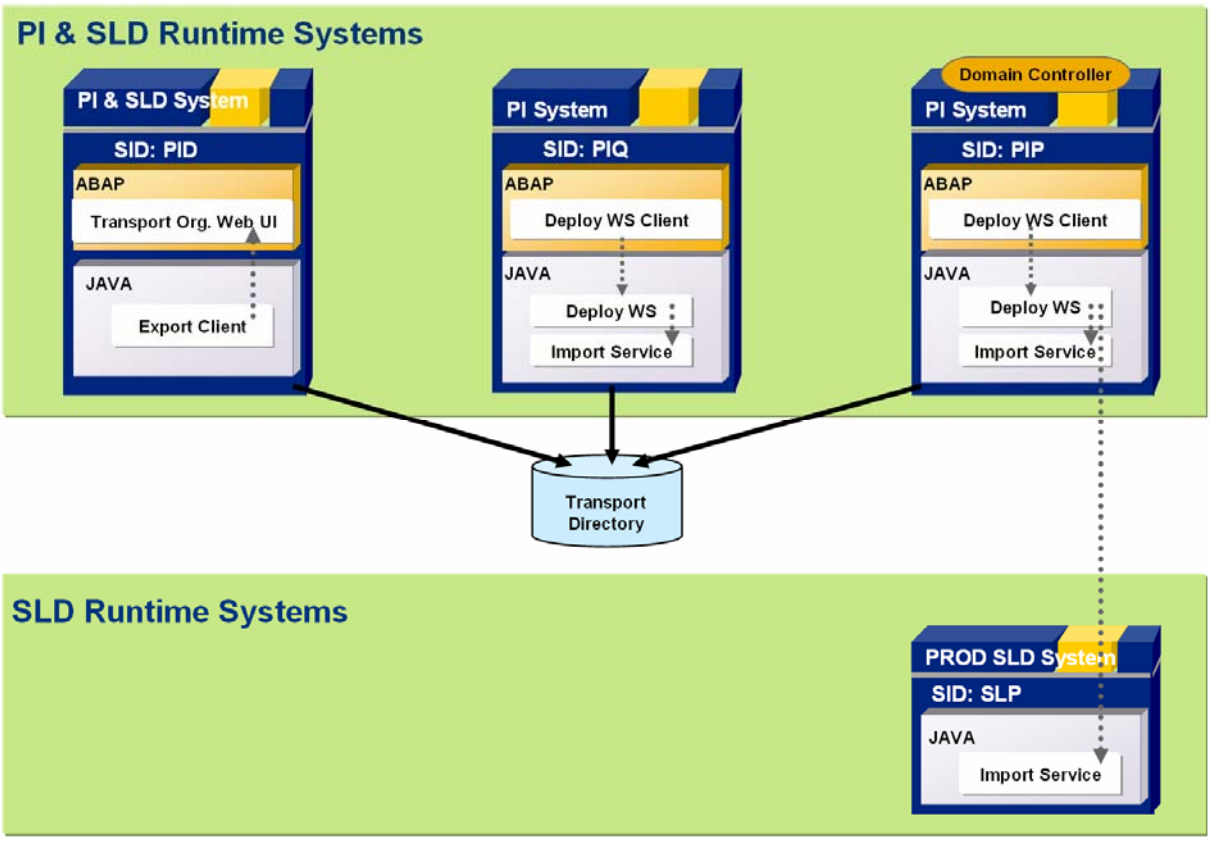
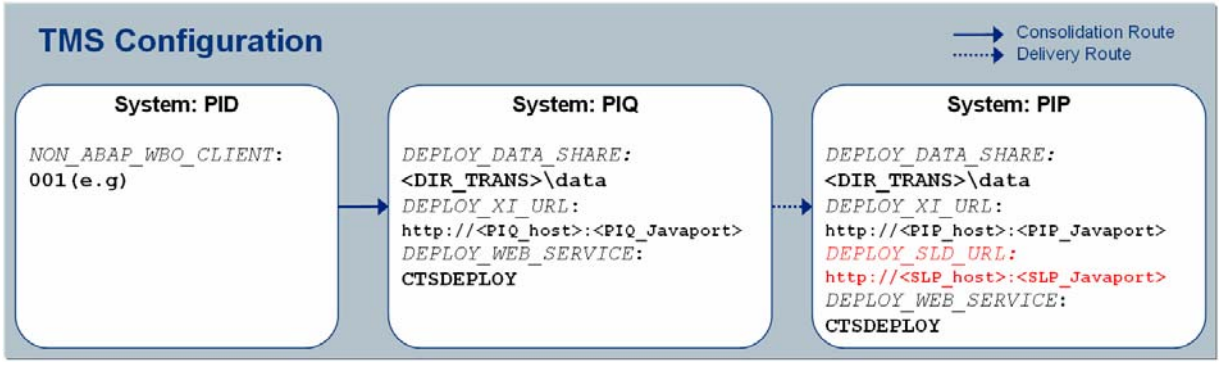


Figure 26: Combined PI & SLD Landscape - no QA SLD System

The TMS configuration for this scenario is shown below. Because of the missing *DEPLOY_SLD_URL* for QA Systems there will be no deployment for SLD content on QA – it is just left out.



Add the XI and SLD user and password for the system PIP using the wizard or via TMS menu.

4.3.1.7 Combined PI and SLD System Landscape – no DEV SLD System

The following scenario is similar to scenario 4.3.1.6 but here no development SLD System exists. The configuration is done as described in 4.3.1.5 where the *DEPLOY_SLD_URL* points to the separate AS JAVA SLD (here: SLP) system. The PI systems for DEV and QA are connected against the same SLD – in that case the QA SLD (here: PIQ). XI objects are created and recorded in the DEV PI system (here: PID). SLD objects are created in the second system which is used for SLD development but acts as the PI QA system (here: PIQ). One transport cannot contain XI and SLD objects together, so no synchronization between SLD and XI is possible.

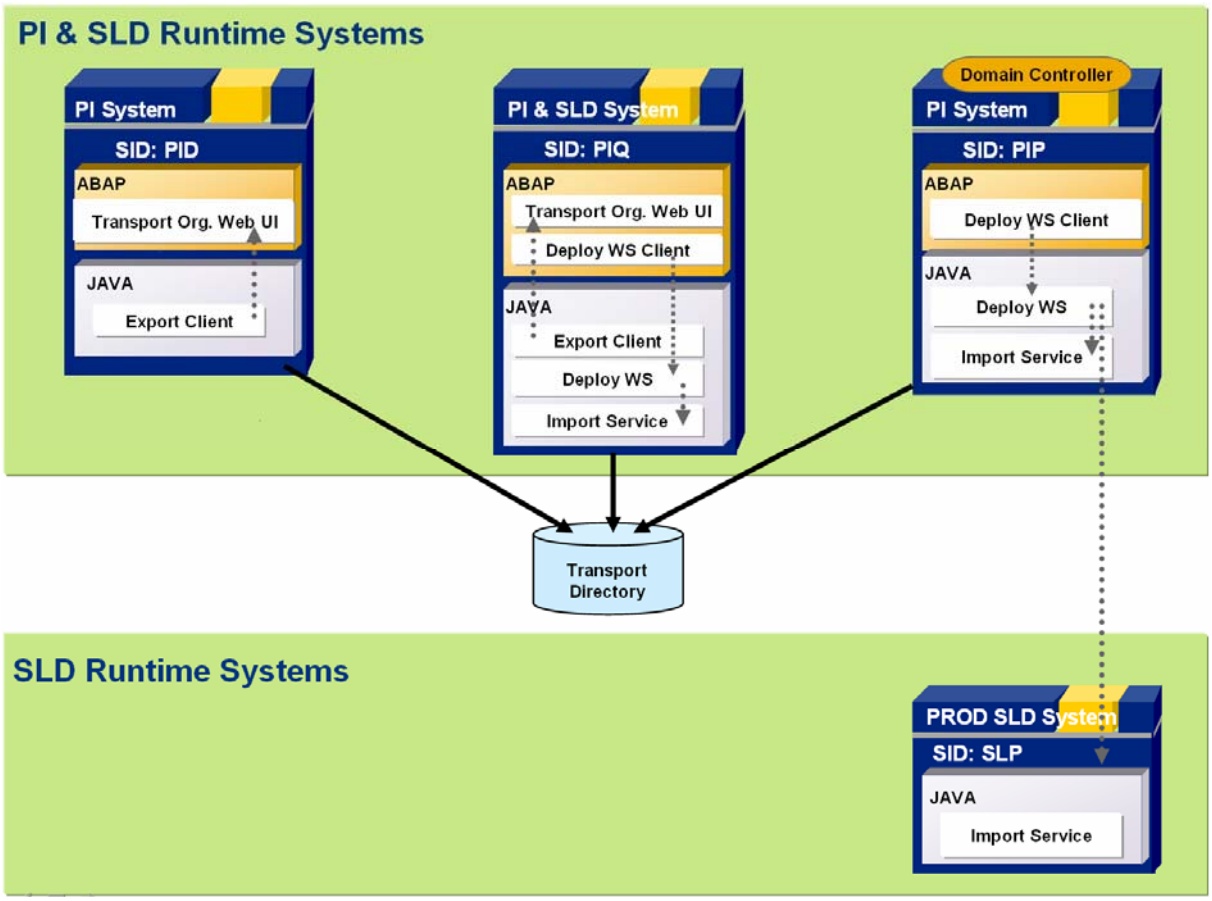


Figure 27: Combined PI & SLD Landscape - non DEV SLD System

As the SLD transport starts with the second system in the landscape the transport route and source system needs to be configured separately.

The typical and recommended configuration can look like this: The quality system is configured as target system for the PI transports and as source system for the SLD transports. Therefore the Transport Organizer Web UI needs to be configured for the QA system (here: PIQ) as well (parameter *NON_ABAP_WBO_CLIENT*).

A new transport layer and consolidation route from the SLD DEV/QA system (here SAPSID: PIQ) to the target PROD system (here SAPSID: PIP) is needed for the transport of SLD objects. The configuration of the PIP system included the tp parameter *DEPLOY_SLD_URL* that points to the real SLD runtime system (here: SLP).

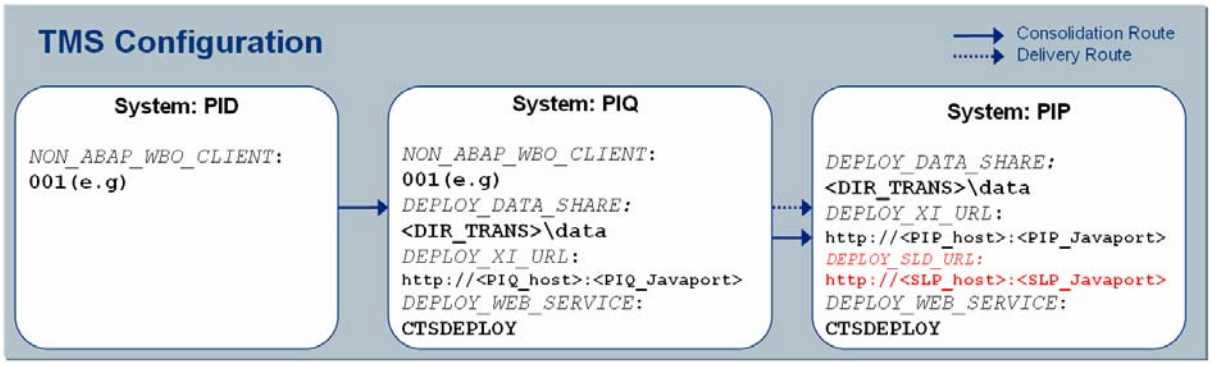


Figure 28: TMS configuration - two routes to PIP

The configuration shown in Figure 28 gives the possibility to transport SLD objects together with PI objects that were changed in the quality system (repair scenario). If PI repairs should be prevented for the PI quality system, it should be done in the application itself (e.g. via permissions). ABAP objects are transported using a separate transport layer. As long as this layer is not configured for the consolidation route, no ABAP repairs in the quality system are possible. For the transport of SLD and PI objects the same transport layer is used – the standard transport layer. This might change in future releases.

In case you want to split the transports of PI and SLD objects explicitly, you can setup a configuration as shown in Figure 29. Here an additional non-SAP system for the productive SLD system is needed. That leads to the fact that two import queues need to be maintained. Furthermore it is not possible to repair PI objects in the quality system (her: PIQ) and transport them to the production PI system. They would be transported to the SLD production system and the import would fail without an error message. For repairs of PI objects in PIQ a consolidation route from PIQ to PIP is required. If that is not the case, repairs of PI objects should be prevented by permission in the PI quality system.

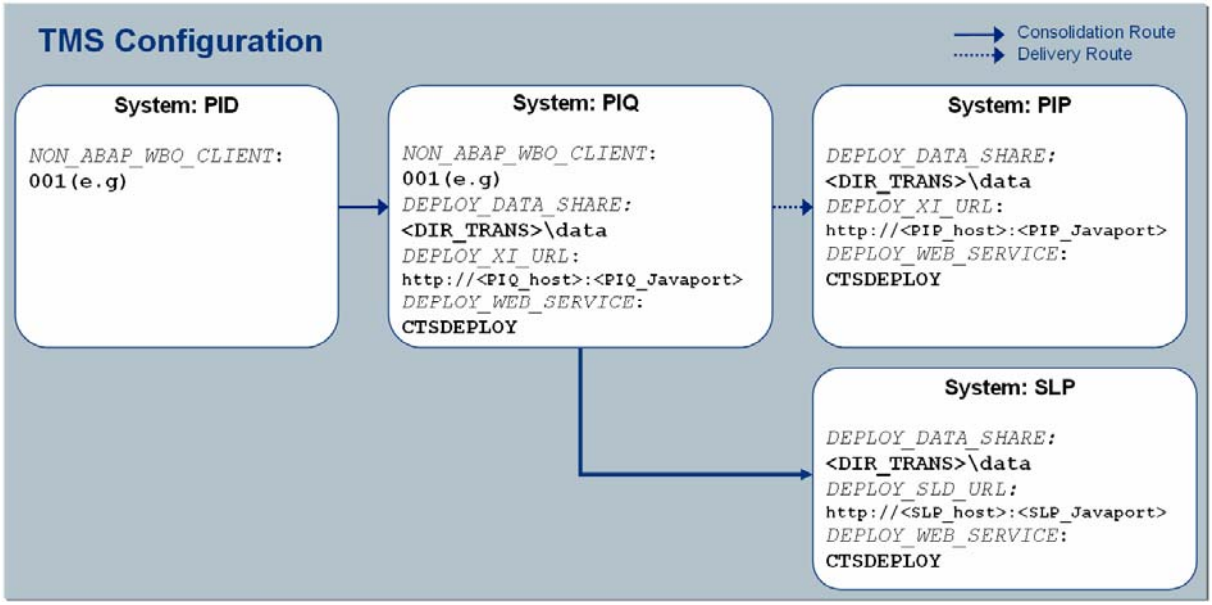


Figure 29: TMS configuration - additional non-ABAP target system

Add the XI user and password for the systems PIQ and PIP and the SLD user and password for the system SLP using the wizard or via TMS menu.

4.3.2 NWDI and SAP Portal Landscape

A typical scenario where use cases are combined is the transport of portal content (e.g. Portal iView) together with their Java applications (e.g. Java Web Dynpro application). Depending on the existing Portal and Java landscape the transport scenario looks different. As the recommended development infrastructure for Java based developments is the NWDI and NWDI is tightly integrated with CTS+ the combined use cases NWDI and SAP Portal are shown in this chapter. The configuration options will be explained in the following.

4.3.2.1 Combined SAP Portal and NWDI/Java landscape

To be able to transport portal content changes along with the corresponding Java based application, the deploy target needs to be the same. That means that the runtime systems in the landscape for the portal (usage type EP) and the Java stack (usage type AS JAVA) are the same SAPSIDs. This type of transport configuration is shown in Figure 30. There are two dedicated systems, one act as NWDI server and the other one as central CTS+ server.

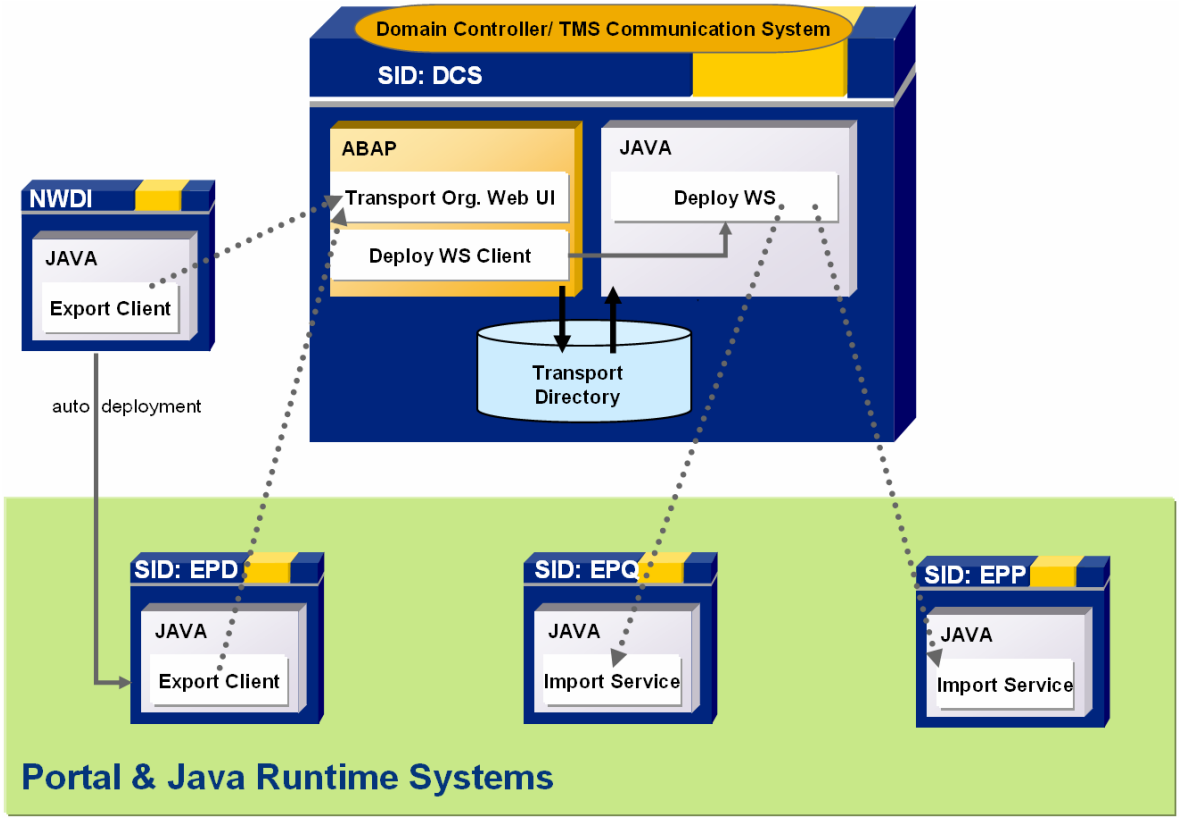


Figure 30: Combined SAP Portal and NWDI/Java landscape

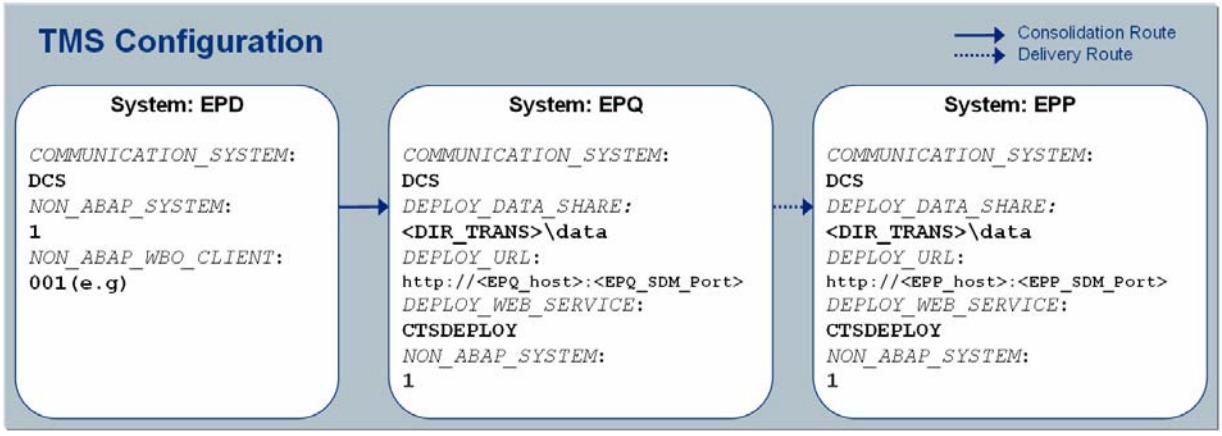


Figure 31: TMS configuration

Export Client Configuration for System NWDI and EPD is needed as described in chapter 3.3. For the NWDI Track the CTS Upload System EPD has to be specified.

4.3.2.2 Separate Landscape for SAP Portal and NWDI/Java landscape

The following scenario describes a system setup and configuration with transport route for portal systems and a separate transport route for Java systems. Therefore a combined transport of portal content with their Java based applications in one transport request is not possible.

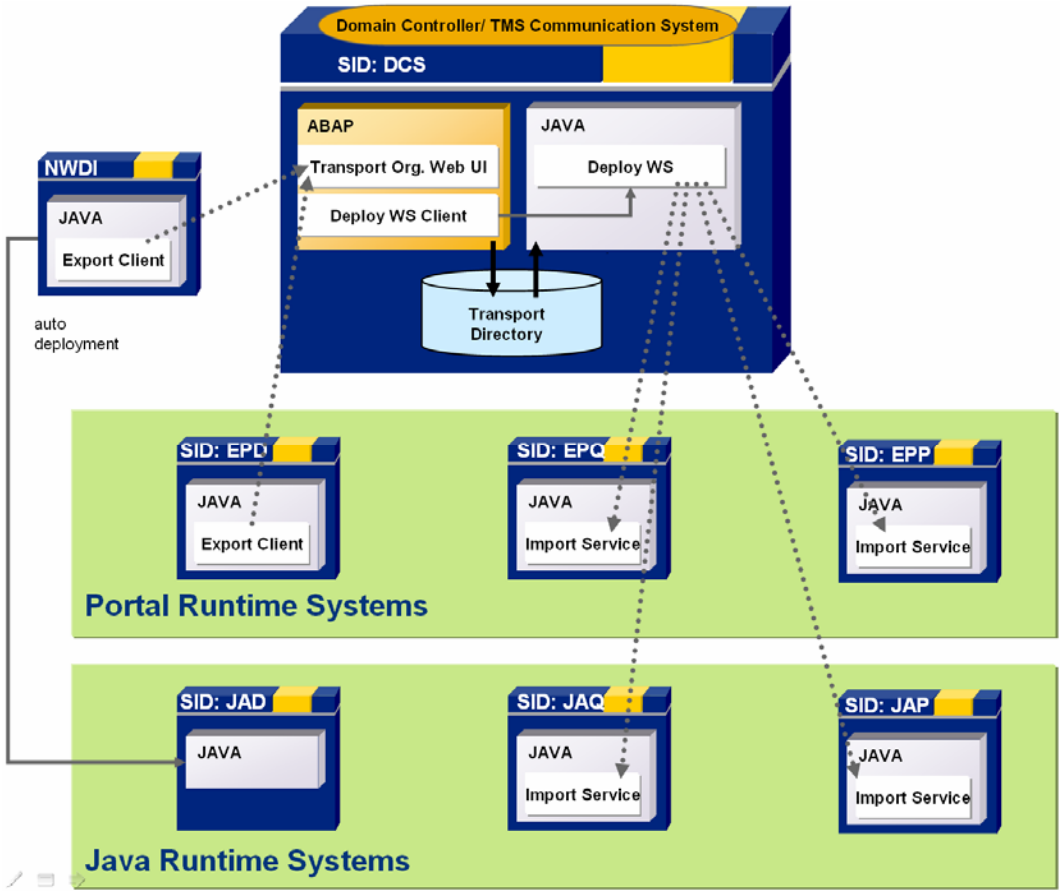


Figure 32: Separate EP and Java system landscape

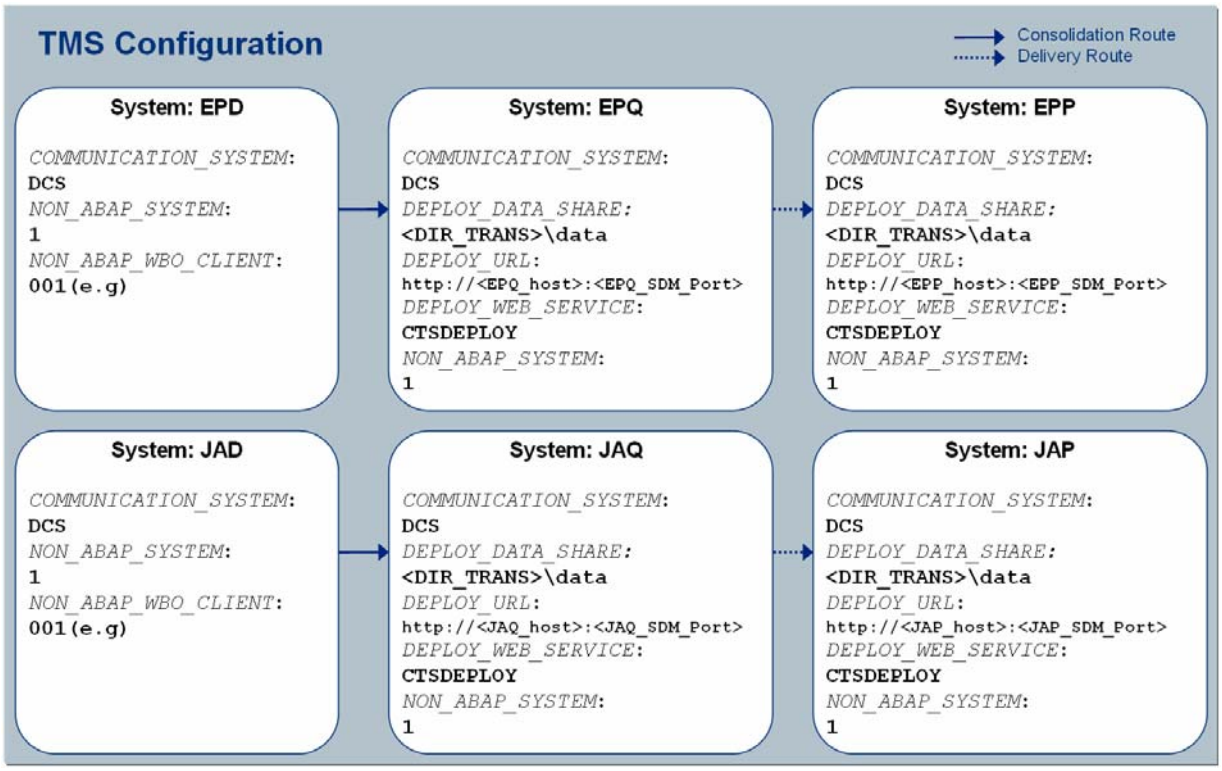


Figure 33: TMS configuration EP and Java landscape

Export Client Configuration for System NWDI and EPD is needed as described in chapter 3.3. For the NWDI Track the CTS Upload System JAD has to be specified.

4.3.3 Enhanced CTS Configuration in a Complex Landscape

The following scenario should explain the aspect of a complex landscape. The screenshots below combines a Portal, Java, PI and ABAP ECC system landscape. Here a separate CTS+ system (e.g. the Solution Manager) in combination with the distributed CTS+ functionality on the PI dual stack systems is used. Please keep in mind that the system configuration is done in the domain controller system.

> **Note**
>
> From CTS+ point of view, it is not important to configure everything within one domain. Domain links can be used as well in case multiple domains are use within the landscape.

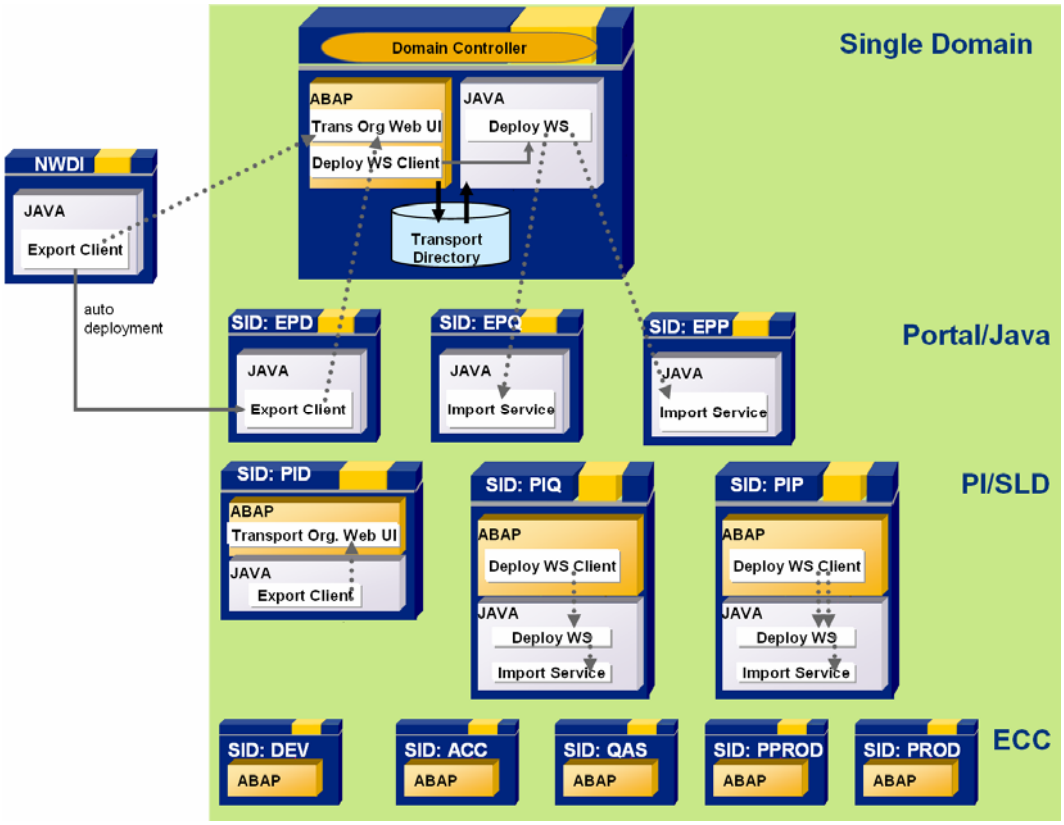


Figure 34: Complex system landscape – Single Domain

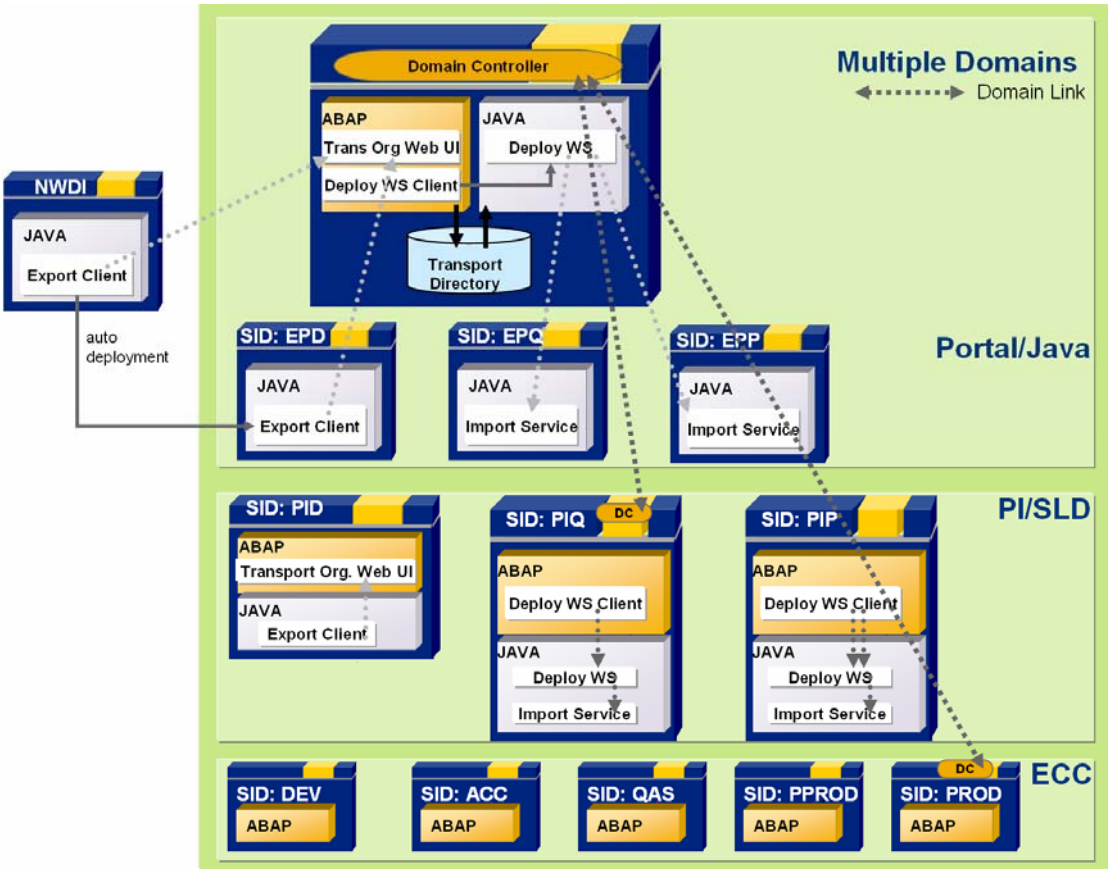


Figure 35: Complex System Landscape - Multiple Domains

www.sdn.sap.com/irj/sdn/howtoguides

THE BEST-RUN BUSINESSES RUN SAP™