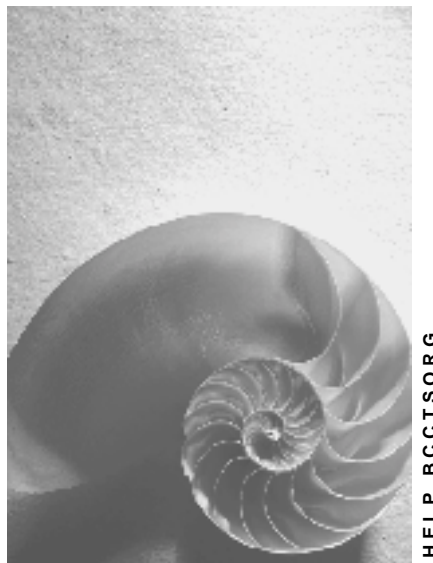


Transport Organizer (BC-CTS-ORG)



Release 4.6C



Copyright

© Copyright 2001 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®, NT®, EXCEL®, Word®, PowerPoint® and SQL Server® are registered trademarks of Microsoft Corporation.

IBM®, DB2®, OS/2®, DB2/6000®, Parallel Sysplex®, MVS/ESA®, RS/6000®, AIX®, S/390®, AS/400®, OS/390®, and OS/400® are registered trademarks of IBM Corporation.

ORACLE® is a registered trademark of ORACLE Corporation.

INFORMIX®-OnLine for SAP and Informix® Dynamic Server™ are registered trademarks of Informix Software Incorporated.

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







HTML, DHTML, XML, XHTML 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.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax
	Tip

Contents

Transport Organizer (BC-CTS-ORG)	6
Requirements for Working with the Transport Organizer	7
Processing After Installation of the CTS	8
Configuring Transport Routes	9
Setting the System Change Option	10
Client Control	12
Object Editing in Customizing and Repository	14
Request Types and Task Types	15
Object Directory	16
Development Classes	18
Editing Development Classes	19
Naming Conventions for Development Classes	21
Change Recording in the Repository	23
Creating Objects	24
Changing Objects	25
Changing Objects in the Original System	26
Changing Objects in a Non-Original System	27
Change Recording in Customizing	28
Working with the Transport Organizer	29
Creating a Request	30
Creating a Request by Editing an Object	31
Creating a Request in the Transport Organizer	32
Creating a Task	33
Request Attributes	34
Editing Attributes	35
Editing the Attributes of Change Requests	36
Attribute SAP_CTS_PROJECT	37
Including Objects in a Request Manually	38
Including the Object Lists of a Request	39
Including Object Lists from Multiple Requests	40
Selecting Objects Freely	42
Object List Editor	43
Locking Objects in Tasks and Requests	44
Using the Protection Function	46
Deleting Objects from Tasks and Change Requests	47
Object Checks When Releasing Requests	48
Setting Object Checks	50
Documentation of Tasks and Requests	51
Documenting Tasks and Requests	52
Releasing Tasks	53
Releasing Change Requests	54
Releasing Requests	56
Monitoring Transports	57
Return Codes in the Transport Log	58

Request Search.....	59
Functions of the Transport Organizer Tools	60
Transport Workflow (Development).....	63
Transport Proposal Inbox	64
Creating a Transport Proposal	65
Revising Transport Proposals	67
Confirming a Transport Proposal/Applying for Transport into the Next System	68
Finding Transport Proposals.....	70
Working with Projects	71
Defining Dependency Relationships Between Requests	73
Recording Changes in Overlapping Projects	75
Setting the Project Status Switch	76
Version Management of Repository Objects	77
Authorizations in Version Management	78
Creating Versions	79
Displaying and Using Versions.....	81
Archiving Versions	83
Functions in the Version Overview.....	84

Transport Organizer (BC-CTS-ORG)

The **Transport Organizer** provides you with functions for organizing software development projects. It is designed to support projects of all sizes, whether they are carried out centrally or in a distributed environment.

See also: [Change and Transport System - Overview \[Ext.\]](#)

Requirements for Working with the Transport Organizer

This section is intended for the **system administrator** who sets up the Change and Transport System (CTS).

To set up the CTS, you require the administration authorization S_CTS_ADMIN.

For information on system roles and on transport control, see [Change and Transport System - Overview \[Ext.\]](#).

Setting up the system group involves:

- [Processing after installation of the CTS \[Page 8\]](#)
- [Configuration of transport routes in the TMS \[Page 9\]](#)
- [Setting the system change option \[Page 10\]](#)
- [Setting the clients \[Page 12\]](#)

You also need to make a number of settings for controlling the transport programs at operating system level.

The CTS is set up once and then only has to be changed if:

- new SAP Systems are included in the system group
- the roles of individual SAP Systems or clients change.

If you have requested [namespaces \[Ext.\]](#) from SAP AG for your own developments in the ABAP Workbench, you must install these namespaces in your SAP Systems.

For more information on the Change and Transport System, see [Change and Transport System - Overview \[Ext.\]](#).

For more information on the Transport Management System, see [BC - Transport Management System \[Ext.\]](#).

Processing After Installation of the CTS

Processing After Installation of the CTS

If you have installed your SAP System as a copy of an existing SAP System then you must configure the CTS after installation.



In SAP Systems installed with R3setup from the SAP CD the basic settings for the CTS are generated during the configuration of the [Transport Management System \[Ext.\]](#). You do not need to perform any processing after installation.

Transaction SE06 provides the following functions for processing after installation:

- Generating basic settings of the Change and Transport System
- Setting the system change option
- Closing foreign requests and tasks

You must specify the installation type of the SAP System on the initial screen:

- Standard installation

The SAP System was installed from the SAP CD using R3setup. It is assumed that the SAP System is running as a correct, delivered version. No adjustments based on corrections or repairs have been made.



If you set up an SAP System that originated from a database copy using this option, problems could arise when you upgrade the system or modify objects with the Transport Organizer.

- Database copy or database migration

The SAP System was created on the basis of a copy. R3setup provides utilities to do this. The SAP System needs to be assigned a new and independent role within the SAP system group or outside of it.

Before you connect the new SAP System to the SAP system group, you have to give it a name that has not yet been used in the group.



Do not use an existing system name since this could cause serious conflicts and may lead to the loss of data.

Configuring Transport Routes

Prerequisites

Before you can configure the transport routes, the following prerequisites must be met:

- [The transport domain has been configured \[Ext.\]](#).
- All SAP Systems involved were included in the transport domain.

Functions

The configuration of the transport routes is managed in the SAP System that serves as the [transport domain controller \[Ext.\]](#), and can be distributed to and activated in all other connected SAP Systems in the transport domain.

The transport route configuration consists of:

- System attributes
- Consolidation routes
- Delivery routes
- Target groups

SAP provides two editors for configuring transport routes:

- [Graphical editor \[Ext.\]](#)

The SAP Systems and their transport routes are displayed graphically.

You can position and link the SAP Systems together by clicking and holding the mouse.

- [Hierarchical list editor \[Ext.\]](#)

The SAP Systems and their transport routes are displayed in a tree structure.

Setting the System Change Option

Setting the System Change Option

Use

The system change option controls whether Repository objects and cross-client Customizing objects are modifiable or not.

The system change option does not affect client-specific Customizing changes. To set whether these changes can be made or not, go to [Client Control \[Page 12\]](#).

You can make more precise settings for Repository objects: Each Repository object is in a [software component \[Ext.\]](#) and a [namespace \[Ext.\]](#) or [name range \[Ext.\]](#). For a Repository object to be modifiable, all the following settings need to be set to *modifiable* as well:

- the global setting
- the software component of the object
- the namespace or name range

You can set the Repository objects in a particular software component to *Modifiable*, *Restricted modifiability* or *Not modifiable*. *Restricted modifiability* means that you can create Repository objects in this software component as non-original objects only.



Restricted modifiability is the same as *Modifiable* for development classes.

You can set the Repository objects in a particular namespace to *Modifiable* or *Not modifiable*.

The SAP System logs this activity. Choose *Log* to display the settings log.

		Software Component		
		Modifiable	Restricted modifiability	Not modifiable
Namespace	Modifiable	Existing objects can be repaired	Existing objects can be repaired	
		New objects have the SAP System ID as the original system	New objects have SAP as the original system	
	Not modifiable	No changes possible		

Prerequisites

To set the system change option, you require the administration authorization in the CTS. It is contained in the delivered standard authorization S_CTS_ADMIN.

Procedure

To set the system change option, proceed as follows:

1. In the Workbench Organizer, choose *Goto* → *Transport Organizer tools*.

This takes you to the Transport Organizer tools overview.

2. Go to *Administration* and start the program *Set System Change Option*.

The *Global setting* option allows you to determine whether objects from the Repository or client-independent Customizing are globally modifiable or not.

Only if the global setting is set to *modifiable*, can you set the system change option of the software components and the namespaces and name ranges.

By choosing *Edit* → *Namespaces modifiable* and *Edit* → *Own namespaces modifiable*, you can set the namespaces and name ranges to *Modifiable* for all objects or for your own objects.



If you want to change the objects in your customer name range, set the [software components \[Ext.\]](#) LOCAL and HOME and the customer name range to *Modifiable*. This customer name range includes, for example, all reports beginning with Z or Y.



If you want to create or edit local [objects \[Page 24\]](#) in your SAP System, you must set the software component LOCAL and the customer name range to *Modifiable*.

Client Control

Client Control

You can use the Transport Organizer to record all changes to Customizing settings in change requests and mark them for transport.

The transport of Customizing settings to another SAP System is prepared. You only need to release the change request.

However, it is not always a good idea to record every single change made to the system. That is why many SAP Systems have a test, training, or demo client in addition to the production client. Recording changes is not appropriate here. In extreme cases, this could even result in unintentional transports that could destroy the target client.

To meet these sometimes contradictory requirements, you can assign each client appropriate attributes, which you can maintain in table T000 (Transaction SM30):

- **Role of the client:** Indicates whether the client is a production, test, training, demo, or Customizing client.
- **Changes and transports for client-specific Customizing objects:** You can specify for each client whether you want the changes to be recorded. As with the [system change option \[Page 10\]](#) in the Transport Organizer, changes can also be forbidden here altogether. You can make the following settings:
 - Changes without automatic recording
 - Changes with automatic recording
 - No changes possible
 - No transports possible



The setting selected for a client applies only to changes to its client-specific Customizing settings, not to client-independent settings. The changes made to client-independent Customizing settings are recorded together with the changes to Repository objects and do not require a client-specific entry in table T000.

In a production client, client control does not influence the current settings (Customizing activities that can be accessed directly from the application menu). You can always change these settings in a production client; they are not recorded in Customizing requests.

- **Changes to cross-client objects:** You can specify for each client whether you want to allow changes to Repository objects and/or cross-client Customizing objects.

Different request types in the Transport Organizer allow you to distinguish between the two modes. Changes to cross-client Customizing objects and to Repository objects are recorded in change requests, whereas changes to client-specific Customizing objects are recorded in **Customizing requests**.

If you use only Customizing requests, you make sure that the results of a Customizing project can be transported to the target client of another system without affecting other clients there.

In contrast, no guarantee can be given that the results of transporting change requests will be restricted to one client. In this case, the import must be checked to establish whether it includes any cross-client objects. If such objects are found, we recommend that you adjust the

corresponding settings in the source and target systems in order to assess the affect on all other clients.

Other features that affect the client are:

- **Flag that locks the logon procedure.** It is set in the target client by the client copy program. This means that no work may be carried out in the target client when client copy is in progress.
 Only the users SAP* or DDIC are permitted to log on. Other regular users are warned with a corresponding message when they attempt to log on.
- **Protection against SAP upgrade** means that the client is no longer provided with SAP upgrades. Users can no longer work actively in the client after the system has been upgraded. The flag can only be set for a test client or an SAP reference client ([EarlyWatch \[Ext.\]](#)). This function is only intended for exceptional cases, when, for example, the basis for an adjustment is needed after an upgrade has been performed. Every "normal" client must be updated by an SAP upgrade, since otherwise data resources from system and Customizing tables required for transactions may not be available. However, you can prevent clients that need to contain certain tables purely for backup purposes (backup clients) from being supplied with SAP upgrades that are not necessary or not wanted. To maintain this flag, you require the administration authorization for the Transport Organizer (authorization S_CTS_ADMIN).
- Restricted permission to **execute CATT procedures**. CATT (Computer Aided Test Tool) allows you to restart recorded test runs repeatedly. This process changes the database and it is therefore necessary to declare this as a property of the client.

Object Editing in Customizing and Repository

Request Types and Task Types

When you transport an object in a request, the object's transport attributes must match the transport attributes of the request (local or transportable).

Different object types have different transport attributes:

- Repository objects and cross-client Customizing objects

Each Repository object has an [object directory entry \[Page 16\]](#). This object directory entry contains the [development class \[Page 18\]](#) to which the object is assigned. In turn, the development class is assigned to a [transport layer \[Ext.\]](#).

If a [consolidation route \[Ext.\]](#) leading from the current system is defined in the TMS for this transport layer, then the object is recorded in a task belonging to a **transportable** change request.

If **no** consolidation route leading from the current system is defined in the TMS for this transport layer, then the object is recorded in a task belonging to a **local** change request.

If the change request is transportable, the target of the request must be the same as the consolidation target of the object.

If the current system is the [original system \[Page 16\]](#) of the object, the object will be assigned to a task of the type [correction \[Page 25\]](#).

If the current system is **not** the original system of the object, the object will be assigned to a task of the type [repair \[Ext.\]](#).

- Client-specific Customizing objects

Client-specific Customizing objects are recorded in tasks that belong to [Customizing requests \[Ext.\]](#).

If a consolidation route leading from the current system is defined in the TMS for the standard transport layer of the system or the current client, then the object is recorded in a task belonging to a **transportable** Customizing request.

The system uses the consolidation route of the standard transport layer to determine the transport target of the Customizing request. However, you can change this target.

If **no** consolidation route leading from the current system is defined in the TMS for the standard transport layer of the system or the current client, then the object is recorded in a task belonging to a Customizing request without transport target.

Object Directory

Object Directory

Definition

The object directory is a directory of all the Repository objects that make up the SAP System.

These include the following:

- ABAP report programs
- Classes and interfaces
- Function groups
- ABAP Dictionary objects (domains, data elements, tables)

Use

The object directory contains standard SAP objects that already exist in a newly installed SAP System as well as your own objects, which you create while working with the ABAP Workbench.

When you create an object, the corresponding object directory entry is also created.

Some Repository objects consist of several sub-objects, which you can edit separately in the ABAP Workbench.

Objects	Sub-objects
Function groups	ABAP program source Screens GUI interface Text elements Function modules Global data (TOP include) and others

These sub-objects do not have their own object directory entry. Only the complete objects are listed in the directory.

Object attributes are also entered in the object directory. They are important for organizing development in the ABAP Workbench and for coordinating the distribution of developments between SAP Systems. The most important object attributes are the following:

- Development class

Every object is assigned to a [development class \[Page 18\]](#). When the object is created, the development class is queried by the SAP System. The development classes are used to structure the Repository and assign the objects to the various SAP components. It also controls the recording and transport of object changes by the Transport Organizer.
- Original system

The original system of an object is the SAP System in which the object was created and in which the object is edited for further developments and corrections. The original

system is unique for each object within a system group. You can change the original system of an object by making a [relocation transport \[Ext.\]](#).

- **Person responsible for object**
Each object is assigned a user as the person responsible. This means that when there you have questions concerning functions or if problems arise - even in an SAP System other than the original system - you can quickly determine the relevant contact person.
- **Original language**
Objects with language-specific components, such as text elements, have an original language. This is the logon language in which the object was created. This attribute is needed to control the translation process if you are developing in more than one language in your SAP System. The original language can be changed when the object is edited in another logon language and this is confirmed explicitly.
- **Generation flag**
In many application areas of the SAP System - in particular in the Customizing transactions - ABAP programs, ABAP Dictionary objects, or other objects are generated automatically from user entries, and are used later when the SAP System is operating productively. These objects are flagged as "generated" in the object directory. To display this flag, use the function *Object attributes* on the dialog box for displaying or maintaining the object directory entry.
- **Repair flag**
In problem situations, it may be necessary to change an object in an SAP System that is not the original system. Such a change is referred to as a "repair". This object is flagged as "repaired" in the object directory. To display this flag, use the function *Object attributes* on the dialog box for displaying or maintaining the object directory entry.

There are various access points for displaying and editing object directory entries. The most important are:

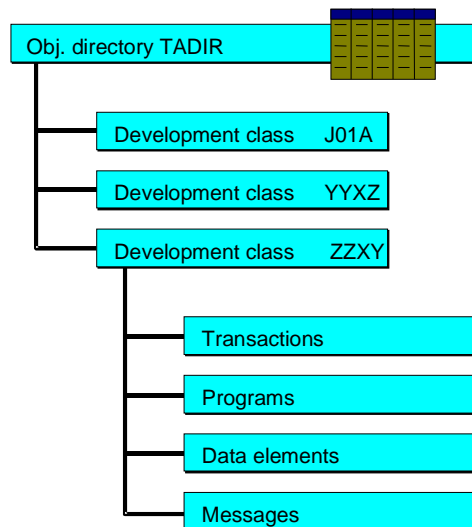
- **When you create an object**, you are prompted for its development class. The current user is proposed as the person responsible. When you access the ABAP Workbench using a predefined development class (*Object Navigator* → *Development class* → *Display*) you are not prompted for a development class.
- In the [Transport Organizer Tools \[Page 60\]](#) (Transaction SE03), there are various selection reports under the *Object Directory* node, with which you can edit directory entries of selected objects or change the person responsible for the objects.
- From most of the **object editors** in the ABAP Workbench, you can display the directory entry by choosing *Edit* → *Object directory entry*. From the ABAP Editor choose *Goto* → *Object directory entry* and from the **request overview** of the Transport Organizer, choose *Object* → *Object directory entry*.
- In the object overviews of the ABAP Workbench **Object Navigator** (SE80) you can change the development class to which individual or selected objects belong with the *Reassign* function.

Development Classes

Development Classes

All Repository objects in the SAP System are classified according to development classes. All objects forming a functional unit, such as programs, ABAP Dictionary objects, and message classes, are grouped together in a development class. The development classes are used to structure the Repository and assign the objects to the various SAP components. The assignment to a development class also allows you to control the recording and transport of object changes by the CTS. (See the section [Request Types and Task Types \[Page 15\]](#).)

The assignment of an object to a development class is recorded in the [Object Directory \[Page 16\]](#).



When development work starts, you must first create a development class to which the new objects are assigned. When you create a new object in the ABAP Workbench, the SAP System asks which development class you want the object to belong to.

The development class also helps you when you navigate through the objects in the Object Navigator of the ABAP Workbench (Transaction SE80).

To improve clarity when objects are displayed in this way, SAP advises you not to let the number of objects in a development class increase too much, but to distribute the objects to new development classes according to appropriate criteria as a project progresses.

The development classes are themselves objects in the ABAP Workbench. Changes to existing and newly created development classes are, apart from a few exceptions (see the section [Naming Conventions for Development Classes \[Page 21\]](#)), recorded by the Transport Organizer and can be transported into other SAP Systems. The development class of a development class is always itself.

Editing Development Classes

To display or edit development classes, go to the ABAP Workbench and proceed as follows:

1. Choose *Object Navigator* or call Transaction SE80.
2. Choose *Development class*.
3. Enter a development class and choose *Display*.



If you want to create a development class, first see the section [Naming Conventions for Development Classes \[Page 21\]](#).

4. Double-click the name of the development class.

The next screen shows the attributes of the development class:

- Short text

When you create a new development class, enter a description of the development area which you want the class to cover.

- Transport layer

The transport layer determines whether changes to objects in this class are recorded in [local or modifiable transport requests \[Page 15\]](#).

If they are recorded in local requests, the changes are only effective in the current SAP System.

If they are recorded in transportable requests, the changes are transported to other SAP Systems after the request has been released. They are transported along a defined transport route that depends on the transport layer.

These attributes of the transport layer are set by the transport administrator in the [transport route configuration \[Page 9\]](#) of the Transport Management System (TMS).

When a development class is created, a transport layer set in the TMS is assigned automatically by the SAP System. You can only change this transport layer with the CTS administration authorization S_CTS_ADMIN.

- Person responsible

This is the user who you consult about any development project being carried out in the development class.

When you create a new development class your user is proposed as the person responsible.

- [Software component \[Ext.\]](#)

Your own developments are assigned to the software component *HOME*.

- Application component

The development classes in the SAP standard delivery are assigned to one component of the application hierarchy.

The assignment to a component is optional for your own development classes.

Editing Development Classes

- Changes are recorded

This flag controls whether changes to objects in the development class are recorded by the Transport Organizer or not. This flag is normally set automatically by the SAP System (see the section [Naming Conventions for Development Classes \[Page 21\]](#)).

When you save changed or newly created objects in the ABAP Workbench they are entered in a Transport Organizer change request. However such entries are only made if the classes are subject to the control of the Transport Organizer.

5. To switch to edit mode, choose .

Naming Conventions for Development Classes

The development classes from the standard namespace (without prefix) are divided into the following name ranges, which determine the functional characteristics of the development class:

- Development class beginning with A-S or U-X

These classes are reserved for SAP standard objects from the SAP name range. You may not create customer objects or objects from prefix namespaces in these classes. Changes to the objects are recorded in the Transport Organizer and can be transported, if you have configured the SAP System appropriately.

These development classes belong to the “SAP” transport layer. They are assigned to an SAP software component (for example, SAP_BASIS, SAP_APPL).

- Development class beginning with Y or Z

You can create customer objects from the customer name range in these classes. Changes to the objects are recorded in the Transport Organizer and can be transported, if you have configured the SAP System appropriately. (See the section [Setting the System Change Option \[Page 10\]](#).)

These development classes are assigned to the software component HOME.

- Development class beginning with T (private test class)

In this type of development class, you can create customer objects from the customer name range, or objects from a prefix namespace installed in your SAP System with the “producer” role (see the section [Releasing Namespaces \[Ext.\]](#)).

When you create a class of this type, you can specify whether or not you want it to be subject to the control of the Transport Organizer. If this is the case, objects that you modify are recorded in local requests by the Transport Organizer, which are not transported. The class is not assigned to a transport layer.

The objects may only be transported to other SAP Systems in special transport requests (transports of copies or relocation transports).

Newly installed SAP Systems always have a private test class called TEST, which is not linked to the Transport Organizer.

These development classes are assigned to the software component LOCAL.

- Development class beginning with \$ (local class)

In this type of development class, you can create customer objects from the customer name range, or objects from a prefix namespace installed in your SAP System with the “producer” role (see the section [Releasing Namespaces \[Ext.\]](#)).

Changes to objects of this class are not recorded in the Transport Organizer. The class is not assigned to a transport layer. The objects cannot be transported.

In newly installed SAP Systems, there is one local class called \$TMP.

Like other Repository objects, development classes can belong to a prefix namespace if the relevant [Namespace \[Ext.\]](#) is installed in your SAP System.

The prefix namespace is designated as follows:

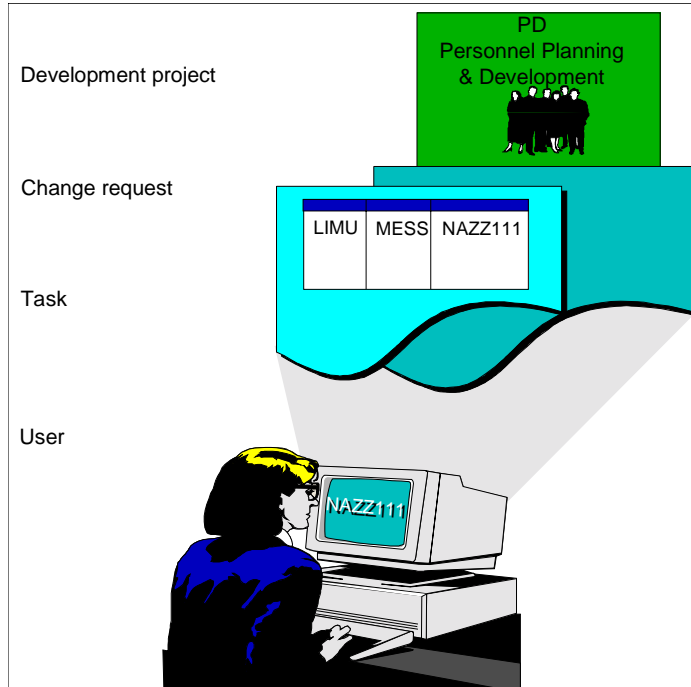
Development class begins with a namespace prefix enclosed in slashes (/).

Naming Conventions for Development Classes

In a development class of this type you can create objects that belong to the same prefix namespace. Changes to the objects are recorded in the Transport Organizer and can be transported, if you have configured the SAP System appropriately.

Change Recording in the Repository

Changes to Repository objects are subject to the control of the Transport Organizer.



The names of the objects changed by a user are entered in an object list. This object list is assigned to a **task** belonging to a user or project member.

One or more tasks belonging to various users are assigned uniquely to a higher-level **change request**, which in turn represents a development project.

Creating Objects

Creating Objects

1. In an editor of the ABAP Workbench, choose *Create*.
2. Give the object a name and maintain its [object attributes \[Ext.\]](#).
3. When you save the object, the dialog box *Create Object Directory Entry* appears. Use one of the following functions to assign the object to a development class:
 - Choose one of the existing development classes.

The development class determines the type of change request that the object will be included in:

 - In a local change request that is not transported to other SAP Systems
 - In a transportable change request that is transported to the target system specified in the system settings when it is released.
 - Choose *Local object*. This assigns the object to the development class \$TMP.

The Transport Organizer does not record changes to local objects. They are consequently not protected against changes by other users and are not transported to other SAP Systems.
4. Enter the object in another request.

After you have assigned a development class in which changes are recorded, a dialog box appears that prompts you to enter a change request.

When you choose *Own requests* you go to the [Request Overview \[Ext.\]](#) of the Transport Organizer. It shows a hierarchical list of the change requests that you are working on. You can only select change requests that can be used for recording changes. This is because of the [development class assignment \[Page 15\]](#) of the object.

Select a change request by double-clicking the appropriate line.

If you are not involved in work on any appropriate change request, you can create a new request in the hierarchical list or, alternatively, choose *Create request* in the dialog box that prompts you to specify a request.

Enter a short description that uniquely identifies the change request in your development project. This makes it easier to find your request in the hierarchical list.

Choose *Continue* to edit the object.

Changing Objects

Tasks are situated at a lower level than change requests and are represented in the request by the names of **users**.

A distinction is made between two different types of tasks for modifying objects:

- Objects are entered in a **development/correction** when you edit them in their original system. Generally, this is the system in which the objects were originally created. For more information, see [Changing Objects in the Original System \[Page 26\]](#).
- Objects are entered in a **repair** when you change them in a system other than their original system. This applies to all modifications to SAP standard objects. For more information, see [Changing Objects in a Non-Original System \[Page 27\]](#).

You can change the objects in a request only if you are involved in a task of the right type in this request.

Changing Objects in the Original System

Changing Objects in the Original System

Use

You can edit existing objects directly in the appropriate editor. When you switch to editing mode, the system checks to see whether you are allowed to edit the object, or whether it is already locked by another user.

Procedure

1. When you save the object, the Transport Organizer request query screen appears. As described in [Creating Objects \[Page 24\]](#), you can either select a change request from a list generated with *Own requests* or create a new request with *Create request*.
2. If you use an existing request that has not yet been assigned any attributes, it will now be given the attribute **local** or **transportable**, depending on the object and the system settings.
3. If you have selected an existing request that you are involved in, but all of your tasks have been released and therefore cannot be used, then a new task of the type **development/correction** is created automatically.

If you only have one task in the request and this has not yet been assigned to a particular type, it becomes a task of the type **development/correction**.

Result

The name of the object is now added to the task object list. From this point onwards, the object is locked by the change request and only users with tasks in the request are allowed to edit the object.

When the object is edited again by any of these users, the request query screen no longer appears. The object is, however, added to the task of the user involved, if it has not already been entered there.

Users working together on a change request constitute a development team working on the same project.

Changing Objects in a Non-Original System

Use

You must only modify objects outside of their original system in urgent cases. These changes later have to be made to the originals as well, so twice as much work is involved.



In particular, you should avoid modifying SAP standard objects.

Inconsistencies may also occur, if the modified objects do not match the new standard. In both cases, you have to make time-consuming adjustments.

Patches to SAP standard objects are an exception. They are organized in consultation with the SAP hotline and will be incorporated in the next SAP maintenance level.

Procedure

1. When you save your changes, the request query screen appears. The procedure is the same as for original objects; you can either select an existing request or create a new one.
2. A change request that has not yet been assigned is given the attribute **local** or **transportable**. If necessary, a **repair** is created for you or an unclassified task is classified as a **repair**.

Result

The name of the object is added to the object list of the repair. From this point onwards, the object is locked by the change request and only users with tasks in the request are allowed to edit the object. In addition to this, the [repair flag \[Ext.\]](#) is set for the corresponding [object \[Page 16\]](#) to prevent the object from being overwritten by imports into the system.

When the object is edited again by any of these users, the request query screen no longer appears. The object is, however, added to the repair of the user involved, if it has not already been entered there.

Users working together on a change request constitute a development team working on the same project.

Change Recording in Customizing

Change Recording in Customizing

Changes to Customizing settings are recorded in different requests, depending on the category to which the Customizing objects belong:

- Client-specific Customizing objects are recorded in **Customizing requests**. Most of the changes made in Customizing fall into this category.
- Cross-client Customizing objects, like Repository objects of the ABAP Workbench, are recorded in change requests of the Workbench Organizer.

In rare cases, changes to cross-client Customizing settings also affect ABAP Workbench objects (for example, due to generated objects). The appropriate procedure for handling such Customizing objects is described in the sections [Creating Objects \[Page 24\]](#) and [Changing Objects \[Page 25\]](#). The following section explains the standard Customizing procedure:

Unlike ABAP Workbench Repository objects, Customizing objects entered in requests are not locked against access by other users. Customizing objects do not have an original system; it is not possible to distinguish between a development and repair.

Changing Client-Specific Customizing Settings

[Client control \[Page 12\]](#) determines:

- whether changes can be made to client-specific Customizing settings in a client
- whether the Customizing settings changed in a client are recorded automatically in Customizing requests.



In a production client, the current settings are not regarded as Customizing settings, but as application objects. These current settings are therefore independent of client management and are not controlled by the Transport Organizer.

If changes are allowed to client-specific Customizing settings in a particular client, you can edit these on a maintenance screen without a request prompt window.

If you have activated automatic recording for a client, whenever you save changed client-specific Customizing settings a Transport Organizer query window appears. Here you can choose *Own requests* to generate a list of Customizing requests that contain your tasks or, alternatively, *Create request* to create a new request for the client in question.

The ID of the changed Customizing object is included in the object list of the task. A list of all changed table keys is also included in the task.

Changing Cross-Client Customizing Settings

When you save changed cross-client Customizing settings, a Transport Organizer dialog box appears, irrespective of any client management settings made. Here you can either select a change request from a list generated with *Own requests* or create a new request with *Create request*.

When the work on individual parts of a project, represented by tasks, has been completed and released by all users, you can release the change request itself. When you release a transportable Customizing request you automatically export the objects it contains.

Working with the Transport Organizer

This section gives a detailed description of the steps involved when you work on development and Customizing projects with the **Transport Organizer**. These include the following:

- Recording changes in **change requests**.
- Working with **change requests** from their creation to release.
- Checking whether transports of released transport requests were successful.



To work with the Transport Organizer you need special authorizations.

If you are not sure whether you have these, read the section on [Authorizations \[Ext.\]](#). It gives you an overview of all existing authorizations. Compare these with your own user profile.

Creating a Request

Creating a Request

You can create a request in the following ways:

- [Creating a Request by Editing an Object \[Page 31\]](#)

When you create or modify objects in the ABAP Workbench or Customizing, a window appears asking you to assign the object to an existing change request (*Own requests*) or to create a new change request (*Create request*).

The request type is defined automatically.

- [Creating a Request in the Transport Organizer \[Page 32\]](#)

Use the function *Create request* in the initial screens and the request overview of the Transport Organizer.

Creating a Request by Editing an Object

Prerequisites

You have edited an object and want to save your changes.

Procedure

1. Choose *Save*. A request dialog box appears.
2. Then choose *Create request*.
3. Write a short description of the new request.
4. If necessary, add tasks for other users.
5. Choose *Save*.

Result

The request type required is defined automatically. You can now save your edited object under this request.



You can also create change requests from the initial screens or a request overview of the Transport Organizer. For more information, see [Creating a Request in the Transport Organizer \[Page 32\]](#).

Creating a Request in the Transport Organizer

Creating a Request in the Transport Organizer

Prerequisites

You are in the initial screen or the request overview of the Transport Organizer.

Procedure

To create a request, proceed as follows:

1. Choose *Request/task* → *Create*.



In the request overview you can position the cursor on a project, a request or a request type. The system automatically saves the appropriate attributes.

2. In the *Create request* dialog box, select a request type and choose *Copy*.
3. Enter a short description that will help you find your change requests again.



Enter a short text for the request that indicates the project it belongs to.
(Example: **FI/CO Customizing**).

4. If several users are to work on this request, enter their user names.
5. Save your entries.

Result

The change request is now available for further work.



You can also create change requests directly when editing objects. For more information, see [Creating a Request by Editing an Object \[Page 31\]](#).

Creating a Task

Prerequisites

You can add new tasks to an existing change request by choosing *Add user*. The tasks are assigned to the user that you enter here.

If you have the appropriate [authorizations \[Ext.\]](#), you yourself may add a task to an existing request belonging to another user.

Procedure

To create a task for a request that is not your own, proceed as follows:

1. In the request overview, choose *Request/task* → *Other requests*.
2. In the *Enter User Name* dialog box, enter the user name of the request owner.
The request overview of the user specified appears.
3. Position the cursor on the request and choose *Request/task* → *Request* → *Add user*.

Result

A task is created for the selected request. You can now edit the request.

Request Attributes

Request Attributes

You can use attributes to identify and describe a change request. Unlike the documentation about the request, you can use attributes to analyze requests.

SAP delivers the following standard attributes:

- [SAP_CTS_PROJECT \[Page 37\]](#)
- SAPCORR
- SAPNOTE
- SAPOSS

You can define other attributes yourself.

For more information, see [Editing Attributes \[Page 35\]](#) and [Editing the Attributes of Change Requests \[Page 36\]](#).

Editing Attributes

Prerequisites

You require the administration authorization in the CTS to edit attributes.

Procedure

To edit attributes, proceed as follows:

1. Call Transaction SE03, or choose *Goto → Transport Organizer tools* in the Transport Organizer.
2. Choose *Administration → Display/Change Request Attributes*.

You can then define and maintain attributes.



Make sure that the attributes you create begin with the letters **Z** or **Y**.



If you want to create the attribute ZPROJECT1, choose *New entries*. Enter ZPROJECT1 and then enter a description in the next field.

If you want the attribute to be obligatory for all change requests, select *Attribute obligatory for requests*. This means that you cannot release change requests that do not have this attribute.

If it is essential that a value is specified for the attribute, select *Attribute value obligatory*.

If you do not want the attribute to be maintained in the SAP System, select *Attribute assigned externally*.

If you want the attribute to be assigned once only for each request, select *Attribute can only be assigned once for each request*.

Editing the Attributes of Change Requests

Editing the Attributes of Change Requests

To edit the attributes of a change request, proceed as follows:

1. Call the Transport Organizer and display your requests in the hierarchical [request overview \[Ext.\]](#).
2. Position the cursor on one of your change requests and choose *Request/task* → *Request* → *Request attributes*.

The attribute maintenance dialog box appears.

In this dialog box you can:

- add new attributes
- delete attributes
- change the value of the attributes.

3. Save your entries.

Attribute SAP_CTS_PROJECT

Use the attribute SAP_CTS_PROJECT to assign change requests to projects.

The attribute SAP_CTS_PROJECT can have the following settings:

- *Attribute value obligatory*
This setting is activated and cannot be changed.
- *Attribute can only be assigned once for each request*
This setting is activated and cannot be changed.
- *Attribute obligatory for requests*
This setting is not activated. If you activate it you can use only change requests that are assigned to a project.



This setting applies to all clients in a system.

- *Attribute assigned externally*
This setting is not activated, and you should not normally activate it. This function is useful only if you use external software to create change requests assigned to projects.

For more information, see [Editing the Attributes of Change Requests \[Page 36\]](#).

Including Objects in a Request Manually

Including Objects in a Request Manually

You do not normally need this function for change requests, since objects are included automatically. You only need to include objects in a request manually for special actions (such as when you relocate individual objects, or when you merge requests).

To include objects in a request, you have the following options:

- Include the [object list from another request \[Page 39\]](#)
- Select and include the [object lists from multiple requests \[Page 40\]](#)
- [Free selection of objects \[Page 42\]](#) (for example, by object type, development class, or the person responsible for the object)
- Manual entry using the [object list editor \[Page 43\]](#)



In all cases, the objects are initially entered in the object list without locks.

At the latest, the SAP System tries to lock the objects when the request is released. If conflicts then occur with other change requests that are locking some of the objects because they are still being edited, then you need to release these change requests first. You can then lock the objects.

This makes sure that your transport request does not transport objects in an undefined intermediate state.



This does not apply to:

- Transports of copies
- Change requests if you have the authorization TABL in authorization object S_CTS_ADMI
- Relocation transports if you have the authorization TABL in authorization object S_CTS_ADMI

In these cases you can display the errors and/or release the requests.

Including the Object Lists of a Request

Prerequisites

You are in the request overview.

Procedure

To include object lists from another request, proceed as follows:

1. Position the cursor on a modifiable request.
2. Choose *Request/task* → *Object list* → *Include objects...*
3. In the dialog box *Include Objects in Request <Request number>*, select *Object list from request*.
4. Enter the number of the request whose object list you want to include and continue by choosing *Copy*.

Result

The entire object list of the specified request is included.

Also read the warning in the section [Including Objects in a Request Manually \[Page 38\]](#).

Including Object Lists from Multiple Requests

Including Object Lists from Multiple Requests

Prerequisites

You are in the request overview.

Procedure

To include object lists from several requests in a single request, proceed as follows:

1. Position the cursor on a modifiable request.
2. Choose *Request/task* → *Object list* → *Include objects...*

The *Include Objects in Request <Request Number>* dialog box appears.

3. In the dialog box, select the *Object lists from multiple requests* option and continue by choosing *Copy*.

The *RSWBOSSR* screen appears.

4. Enter your selection criteria and choose *Execute*.

The requests that are found are displayed in a hierarchical request overview.

5. Edit the list until it contains the requests that you want to merge.

To do this, use the following functions:


- Extend the list

You can extend the list, as in the request search, by making more selections.

- Reduce the list

Select the requests whose object lists you do not want to merge and delete them from the display.



You can use the *Sort sequence* function to find certain requests more quickly. To do this, choose  in the request overview. You can sort your requests by status, request type, transport target, source system, owner, project and client.

- Select the requests that you want to include

To do this, choose *Edit* → *Selection* → ...

6. Choose *Merge*.

Result

The object lists you selected are included in the specified request.



You can also find this function in the Transport Organizer Tools under *Requests/tasks* → *Merge object lists*.

Also read the warning in the section [Including Objects in a Request \[Page 38\]](#).

Selecting Objects Freely

Selecting Objects Freely

Use

You can use this function for Repository objects only. It does not apply to Customizing requests.

Prerequisites

You are in the request overview.

Procedure

To include objects of your choice in a request, proceed as follows:

1. Position the cursor on a modifiable request.
2. Choose *Request/task* → *Object list* → *Include objects...*
The *Include Objects in Request <Request Number>* dialog box appears.
3. In the dialog box, select *Freely selected objects* and continue by choosing *Copy*.
The *Object Selection* screen appears.
4. Enter your selection criteria and choose *Execute*.
The objects found are displayed in a hierarchical overview.
5. In the list, select the objects that you want to include.
6. Choose *Objects* → *Save in request*.

Result


The selected objects are included in the request.



You can also choose the function for choosing objects freely from the Transport Organizer tools under *Objects in Requests* → *Include Objects in a Transport Request*.

Also read the warning in the section [Including Objects in a Request \[Page 38\]](#).

Object List Editor

Access the object list editor from the request overview by positioning the cursor on a request or task, and choosing . If you selected one of your own requests or tasks, you can switch to change mode.

However, the object list editor requires basic knowledge about the syntax of the entries. Entering a large amount of objects is also a laborious task.

To place objects in a request more easily, use the functions described under [Including Objects in a Request \[Page 38\]](#).

Locking Objects in Tasks and Requests

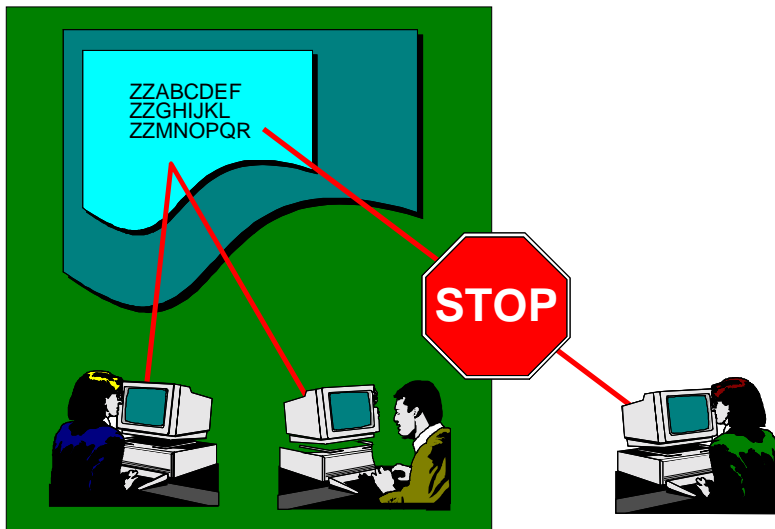
Locking Objects in Tasks and Requests



This section only refers to lockable objects (all Workbench objects are lockable; some cross-client Customizing objects are also defined as lockable).

If you create or change an object as part of a development project, the object is assigned to a change request.

The first user who starts work on an object has to specify a change request where the work will be recorded. This locks the object so that it can only be changed by users who are working on a task within the change request.



All users working on the object have a corresponding entry in the object list of their task. This enables you to determine which users have actually edited the object.

Object locks are deleted when change requests are released. The objects can then be edited again by all developers within the Transport Organizer.



If you enter objects manually in the object list of a task or request, these objects are not locked. You can prevent them from being accessed by other users by choosing

Request/task → Object list → Lock objects.

You can usually only release requests when all the objects they contain have been locked. This makes sure that only released object versions are exported and that inconsistent intermediate versions are not imported into other SAP Systems. Exceptions: You have administration authorization or the request is a transport of copies. For more information, see [Releasing Requests \[Page 56\]](#).

Using the Protection Function

Using the Protection Function

If you want only the owner of a request to be able to add more users, then you can protect the request.

To do this, select the request and choose *Request/task* → *Request* → *Protect*.

Use the *Remove protection* function to remove this restriction.

Deleting Objects from Tasks and Change Requests

To delete an object from the object list of a task or request, proceed as follows:

1. Go to the request overview and choose *Edit → Expand*.
2. Position the cursor on the object name.
3. Choose *Object → Delete entry*.

If the last version of the object in the version database does not correspond to the active version, a warning appears. It informs you that the object has been changed since it was entered in the task or request.

If the object is locked, you must confirm the deletion in the dialog box that now appears. The object is then removed from the object list.



Any changes that you or other users in the same project have made remain, and are not reset automatically.

The changes to these objects are not transported and are not recorded by [Version Management \[Page 77\]](#).

This can lead to serious inconsistencies in subsequent systems, the source of which cannot be identified.

Therefore, enter the object in another request.

Object Checks When Releasing Requests



The following section only applies to Workbench requests.

When you release a Workbench request, you can subject the objects in the request to various checks.

The object checks currently consist of an extended program check and ABAP Dictionary checks. The ABAP Dictionary check checks whether all objects in the request have the status *Active*.

Setting the Object Checks

As a user with the CTS administration authorization (S_CTS_ADMIN), you can do the following:

- Activate or deactivate the object checks for all users
- Leave it up to the user to decide whether to activate or deactivate the checks

For more information, see [Setting Object Checks \[Page 50\]](#).

Checking Objects

To check the objects in the Transport Organizer before the request is released, position the cursor on the request or task, and choose *Request/task* → *Overall checks* → *Objects (syntax check)*.

Object Checks When Releasing the Request

If you have activated the checks, the following occurs when you release the request:

- If the check results in the cache are up-to-date and no errors were found, the release process is continued.
- If the check results in the cache are up-to-date and errors were found, a dialog box informs you that there were errors. You can then:
 - continue the release process anyway
 - cancel the release and display the errors
 - cancel the release and return directly to the request overview.
- If the check results in the cache are not up-to-date, the remaining checks must now be made. For technical reasons, the release is terminated. You can either start the checks immediately or - recommended for large requests due to the long runtime - schedule them in the background:
 - If you run the checks in the foreground, the results of the checks are displayed automatically. You can then repeat the release process.
 - If the checks are run in the background and the objects do not contain any errors, the release is started automatically in the background afterwards. After the background processing has been completed, a dialog box informs you about what actions have been performed.

Setting Object Checks

Setting Object Checks

Prerequisites

You need administration authorization to activate or deactivate object checks.

Procedure

1. In the initial screen of the Transport Organizer, choose *Goto* → *Transport Organizer tools*.
2. Under the *Administration* node, call the *Global Customizing Transport Organizer* tool.
3. Select the appropriate options.

Result

If you select the option *Can be set for specific user* the individual users can activate or deactivate the checks themselves, either from the initial screen of the Transport Organizer under *Settings* → *Transport Organizer*, or from the request overview under *Utilities* → *Settings*.

Documentation of Tasks and Requests

We recommend that you write documentation for all tasks and requests while working on requests. For more information, see [Documenting Tasks and Requests \[Page 52\]](#).

You can add to the documentation by choosing *Goto* → *Documentation*. This is no longer possible after the request has been released.



Writing clear, relevant documentation for tasks and requests helps all developers to understand internal development processes. The documentation is also one of the requirements for revising development work.

Documenting Tasks and Requests

Documenting Tasks and Requests

To create documentation, proceed as follows:

1. In the request overview of the Transport Organizer, position the cursor on the task for which you want to write documentation.
2. Choose *Goto* → *Documentation*.

The documentation maintenance screen appears where you can enter information such as:

- purpose of the development project
- current status of the development project
- persons responsible and contacts
- references to other internal documentation or internal instructions
- details on implementation
- dependencies on other development projects.

Result

The documentation on the tasks of a request is copied to the documentation of that request when you release the task.

Releasing Tasks

Use

When you release a task, the object entries it contains are copied to the object list of its change request. The objects are still locked. This means that only those users involved in the change request can edit the objects.

Procedure

To release tasks, proceed as follows:

1. Go to the Transport Organizer and choose *Display* to find out which change requests you are assigned to.

To view all the tasks assigned to a particular change request, double-click the node beside it.

2. Position the cursor on the task you want to release and choose *Release*.

If you have not yet written any documentation on the task, this takes you to [documentation maintenance \[Page 51\]](#).

3. Document your changes, save your entries, and go back.

Result

After you have saved the documentation and left the screen the CTS releases the task.

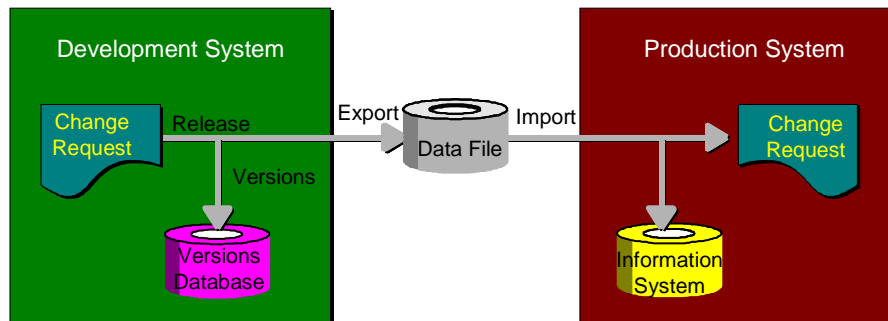
Released tasks are highlighted with a particular color in the request overview (see the legend: *Utilities* → *Color legend*).

Releasing Change Requests

Releasing Change Requests

Once you have [released all the tasks \[Page 53\]](#) in a change request, you can release the change request itself. At this point, the object list of the change request contains all objects that have been changed by the users involved.

To release the change request, position the cursor on the request and choose *Release*.



The following occurs when the change request is released:

- The current version of all objects entered in the request is stored in the version database. This means that the sequence of the change requests under which an object is edited corresponds to the various object versions archived in the database.
- If the change request contains repairs, the repair flag for these objects is reset when you release the request, as long as no sub-objects are locked in another change request. If this is the case, the object's repair flag will be reset only when the last change request is released.

After the repair flag has been reset, the objects are no longer protected from being overwritten by imports into the system.

- When you release a transportable change request, its objects are exported out of the SAP System and copied to an operating system file. The request is also marked for import into the target system.
- The objects entered in the request are unlocked. They can now be changed again.

In the case of transportable change requests, the overview of transport logs for the request is displayed automatically. The physical data export is generally still in progress at this time and a log cannot yet be displayed.

When the export has finished and you have refreshed the display, you can branch to the logs by double-clicking the export steps.

To ensure a smooth export, the CTS must be correctly installed. This is described in the section [Requirements for Working with the Change and Transport System \[Page 7\]](#). A detailed log is created for the export and all subsequent import steps. The section [Monitoring Transports \[Page 57\]](#) describes how you can use the log to check whether the transport was performed successfully.



For the system administrator:

Releasing Change Requests

When you export application-defined objects (such as change documents, hypertext structures, number ranges) you must schedule the transport dispatcher program RDDIMPDP as a background job in the client 000 and in the client from which the change request is being exported.

The program must be scheduled once in each client. To do this, log on in the appropriate client as a user with administration authorization (S_CTS_ADMIN) and execute the report RDDNEWPP in the initial screen of the ABAP Editor (Transaction SE38).

If you export more objects at a later time, you do not need to reschedule the program unless the scheduling information was deleted manually.

For more information, refer to the program documentation of the transport dispatcher program RDDIMPDP and the documentation on the [transport control program tp \[Ext.\]](#).

Importing a Change Request into the Target System

A transportable request is not automatically imported into the target system, since this could disrupt work significantly, particularly if the target system is an SAP System used for productive operation.

The import must be started by the system administrator at an appropriate point in time. The system administrator can decide whether only particular requests or all requests waiting for import should be imported into the target system. The import is performed with the [Transport Management System \[Ext.\]](#) and is described in its documentation.



For the system administrator:

At the time of an import, the transport dispatcher program RDDIMPDP must be scheduled as a background job in client 000 and in the target client into which the change request is being imported.

The scheduling procedure is exactly the same as for the export.

Releasing Requests

Releasing Requests

Use

If you are not the owner of the request, release your own task(s) and ask the request owner to release the request.

If you are the owner of the request, proceed as follows:

Prerequisites

- If it is a Customizing request or a Workbench request, all the tasks in it must have been [released \[Page 53\]](#).
- All lockable objects must be locked.

Exceptions: You have administration authorization or it is a transport of copies. In these cases, the SAP System informs you that not all objects could be locked when you release the request. However, you are allowed to continue with the release.

Procedure

1. Position the cursor on the request.
2. Choose *Release directly*.

Result

The request is released. Afterwards, [monitor your transports \[Page 57\]](#).



If your request contains a lot of objects, it may be best to release it in the background. In this case, choose *Request/task* → *Release* → *Background*. The request released in the background. A dialog box informs you when the release is complete.

Monitoring Transports

The aim of your development work is to introduce new or improved functions into the production system. As a project manager, you therefore want to be informed about the transport of your requests into subsequent systems, and particularly about any transport errors that occur. It is equally important that you know about any repairs made to your objects in a subsequent system, since the import protection for repaired object can interfere with subsequent transports.

For this reason, you can easily access an overview of your transports and repairs on the initial screen of the Transport Organizer.

The stoplight icons indicate transport errors (stoplight is red) or repairs including objects protected against imports (stoplight is yellow).

To access the overview of your transports, choose *Transports*. A hierarchical list appears with your released transports and their transport steps, grouped according to target system.

You can see whether the individual transport steps were successful or not from:

- the color of the entry
- the comment
- the return code

If you have activated extended transport control, then the client you have chosen is displayed in the client-specific transport steps.

If there are faulty transports, analyze the cause of the error. The transport log contains information that may be useful. You can access the transport log by double-clicking the transport step.

When you have found out what caused the error and corrected it, you can give the request the attribute *Error corrected*. This action is recorded in the action log.

Even when transports have been performed successfully, check that they work correctly in the target system. After you have tested the functions in the target system, assign the request the attribute *Tested*.

Transports with the attributes *Error corrected* or *Tested* can be deleted from the display. If the request deleted from the display is imported into another SAP System, it is automatically shown again in the display.

Displaying Transport Logs for Your Own Change Requests

If you display your released change requests in the request overview of the Transport Organizer, you can display the transport logs from here by choosing *Goto → Transport logs*.

Displaying Transport Logs for Any Change Request

When you display individual requests in the Transport Organizer, you can display their transport logs by choosing *Transport logs*. To do this, you need to know the number of the change request. Alternatively, you can enter another user name in the initial screen of the Transport Organizer and list this user's released change requests. Position the cursor on a change request and display its transport logs by choosing *Goto → Transport logs*.

Return Codes in the Transport Log

Return Codes in the Transport Log

To establish whether a transport has been successful, you need to know the meaning of the return codes generated by the programs used for the transport.

Return code	Meaning
0000	Transport performed without errors
0004	Warnings were issued. All objects were transported successfully. There were special actions for individual objects that may not have been intentional, for example, a warning is issued during the export if the request contains an object deletion. Read the warnings.
0008	Individual objects could not be transported successfully. You must analyze and correct the errors. Examples of import errors: <i>Original object was not overwritten</i> , <i>Repaired object was not overwritten</i>
0012 or higher	A critical error has occurred, probably not caused by the contents of the request. You must inform your system administrator.

A return code always has a detailed long text. Read this text for information on the transport error.

Request Search

Use

The request search allows you to select requests of all types with any criteria. The requests that you find are displayed in a hierarchical request overview where you can edit them directly.

Features

The request search lets you make more complex request selections. Access the search from any initial Transport Organizer screen by choosing *Request/task* → *Find requests*. Alternatively, you can also use the request search function in the [Transport Organizer tools \[Page 60\]](#).

As well as letting you change the way in which the request display is sorted, and your personal settings for the request overview, the request search also provides the following functions:

- You can select some of the requests and delete them from the display.
- You can add requests to the list displayed by making another selection.

You can print the request overview at any time.

Functions of the Transport Organizer Tools

Functions of the Transport Organizer Tools

For a short introduction on how to use the Transport Organizer Tools, see [Transport Organizer Tools \[Ext.\]](#) in the section Change and Transport System - Overview.

The tools are subdivided into areas:

- Objects in Requests
 - Search for Objects in Requests/Tasks
Use this program to search for objects in requests/tasks.
 - Analyze Objects in Requests/Tasks
Enter a request or task here, and you can display the request header and the objects in the request. For each object, the object directory entry, development class, original system, person responsible are displayed.
 - Include Objects in a Transport Request
Use this program to select objects and include them (or a subset of them) in a request.
- Objects
 - Modification Browser
This program determines the SAP standard objects modified in the customer system, grouped by development class.
 - Objects in the Customer Namespace
Use this program to generate a list of all customer objects in the current SAP System.
 - [Namespace Information System \[Ext.\]](#)
Use this program to:
 - Display and search for namespaces
 - Display and search for naming conventions
 - Display object attributes
 - Display Repaired Objects
This selects all objects for which the repair flag is set. If the repair flag is set, the object cannot be overwritten by an import.
- Object Directory
 - Change Object Directory Entries
Use this program to maintain the development class and person responsible for objects in the object directory.
 - Change Object Directory Entries of Objects in a Request
Use this program to change the development class and person responsible for objects in a request.

Functions of the Transport Organizer Tools

- Change Person Responsible for Objects

Use this program to change the persons responsible in the object directory. The entries you change are included in a change request.
- Requests/Tasks
 - Find Requests

Use this program to search for requests and tasks according to different criteria (such as owner, status, request type) and display them in a hierarchical list. You can edit the requests from this list, as you can in the Transport Organizer.
 - Merge Object Lists

Use this program to take object lists from selected requests, sort and compress the objects, and place them in a new request.
 - Unlock Objects (expert tool)

Use this program to unlock objects in a request or task.

Careless use of this function may lead to inconsistencies in the target system. First read the documentation on this program in the SAP System.
 - Import Application-Defined Objects (ADOs)

Use this program to import ADOs into the SAP System.

ADOs are imported automatically when a request is imported. If the ADO import was not successful, for whatever reason, you can use this program to restart or repeat the import.
- Administration

To use the following tools, you require the administration authorization in the Change and Transport System.

 - Set System Change Option

The system change option determines which objects of the ABAP Workbench can be changed with the Transport Organizer in the current SAP System.
 - Display/Change Namespaces

Use this program to [create new namespaces \[Ext.\]](#), and display or change the attributes of existing namespaces.
 - Display/Change Naming Conventions

Calls extended table maintenance for [naming conventions \[Ext.\]](#).

Use this program to display, create and change reserved naming conventions in the ABAP Workbench.
 - Global Customizing Transport Organizer

Functions of the Transport Organizer Tools

Use this program to maintain the global Customizing settings for the Transport Organizer for the areas *Transport error at logon*, *Check objects at request release* and *Release request in the background*.

- Display/Change Request Attributes

You can use this program to display or change request attributes, and define new attributes.

Transport Workflow (Development)

Purpose

Use the transport workflow to automate your transport process. It is an efficient method of transporting a selected number of requests into a group of transport targets, and uses clearly defined approval steps to ensure the quality of your target systems.

Prerequisites

- Transport administration has [configured the transport workflow \[Ext.\]](#).
- You have a user in the [Workflow Engine \[Ext.\]](#) system/client.

Process Flow

After you have [released the requests \[Page 56\]](#), you must create a [transport proposal \[Ext.\]](#) in the Transport Organizer. To do this, specify the requests, and the target systems/clients into which you want to transport them. You can also specify an import time, import options and notes for the transport administration.

When you [create a transport proposal \[Page 65\]](#), you place it in the [TMS worklist \[Ext.\]](#) for the transport administration. Transport administration can [approve the transport proposal \[Ext.\]](#); it is then imported automatically into the target systems. The transport proposal then appears again in your [transport proposal inbox \[Page 64\]](#), for you to [confirm \[Page 68\]](#). You can complete the proposal by confirming it, or apply to have it transported into other systems.

However, transport administration can also reject the transport proposal. It then appears in your transport proposal inbox again, possibly accompanied by a note. You can then [modify \[Page 67\]](#) the transport proposal and place it in the workflow worklist of the transport administration.

Transport Proposal Inbox

Transport Proposal Inbox



Definition

Access to [transport proposals \[Ext.\]](#) for developers.

Use




The transport proposal inbox contains the transport proposals that need to be edited by the development team. From the inbox you can edit transport proposals that have come from transport administration, forward them to other users, or resubmit them at a different time. You can also display any notes, and the workflow log.

Structure

To edit a transport proposal, double-click it, or select it and choose . To refresh the worklist, choose .

To use the following functions, first select a transport proposal, and then choose the function:

Functions in the Transport Proposal Inbox

Forward transport proposal to another user	
Resubmit the transport proposal	
Display the workflow log	

See also:

[Workflow Inbox \[Ext.\]](#)

Creating a Transport Proposal

Use

Use the transport workflow to automate your transport process. It is an efficient method of transporting a selected number of requests into a group of transport targets, and uses clearly defined approval steps to ensure the quality of your target systems. To use the transport workflow, you must first create a [transport proposal \[Ext.\]](#).

Prerequisites

- You have [configured the transport workflow \[Ext.\]](#) for your SAP System.
- You have a user in the [Workflow Engine \[Ext.\]](#) system/client.
- Released requests exist.

Procedure

1. Go to the [Transport Organizer \[Ext.\]](#) and select released requests.
2. Select a request that you want to transport and choose *Utilities* → *Create transport proposal*. The dialog box *Create Transport Proposal* appears.



If you are using the transport strategy [workflow-controlled transports \[Ext.\]](#), the transport workflow starts automatically when you release a transport request, and the screen *Create Transport Proposal* appears. The requests are then released implicitly when the transport proposal is sent to the transport administrator.

3. Enter a short text, the target system and any other requests. The target systems apply to **all** requests that are entered.



Note that any requests that you place in the transport proposal have to have been released. To display the standard targets, choose . These are the SAP Systems that are on the standard transport route of the request. To display all SAP Systems into which you can transport objects, use the F4 help on the field *Tgt systems*.

Choose to display all target systems that follow on directly from this system. You can use this function to distribute transports throughout your system landscape, system-by-system.

TMS can make only one entry in the import queue for [virtual \[Ext.\]](#) and [external \[Ext.\]](#) SAP Systems. If you are transporting into an external SAP System, you need to make the import manually in the domain of the external system.




To display the requests, choose , and to display the transport logs, choose .

4. If necessary, specify the import time and the [import options \[Ext.\]](#).




Creating a Transport Proposal

If you do not make any entries here, the import is started immediately. The import option *Leave transport request in queue for later import* is used for Customizing requests and transportable change requests, as long as you have not set [workflow-controlled transports \[Ext.\]](#) or [single transports \[Ext.\]](#) as your [transport strategy \[Ext.\]](#) in TMS.

5. To write a note for the transport administration, choose .
6. To check your entries, choose .
7. To make a test import into the target system, choose .



You can make a test import into another transport group, only if you have the authorization S_CTS_ADMIN (from the profile S_A.SYSTEM). If you do not have this authorization, ask the system administration to make the test import for you.

8. To create the transport proposal, and then forward it automatically to transport administration, choose .

Result

A transport proposal is created. The SAP System gives this transport proposal a proposal number, and then places it in the TMS worklist of the transport administration.

Revising Transport Proposals



Use

If the transport administration rejects your [transport proposal \[Ext.\]](#), it appears again in your transport proposal inbox. You can cancel the transport proposal, or revise it and send it back to transport administration.


Prerequisites




- Transport administration has rejected your transport proposal.
- You have a user in the [Workflow Engine \[Ext.\]](#) system/client.

Procedure

1. In the Transport Organizer, choose the tab page *Transport proposals*.
2. Choose  *Inbox*. Double-click the transport proposal that you want to revise.
3. If there are any notes, you can display them by choosing .
4. When you revise a transport proposal, you can also change its requests, the target systems, the import times and the import options.



If you want to cancel the transport proposal, choose . This transport proposal now has the status *canceled* in the database and cannot be edited any more.

5. To check the transport proposal, choose .
6. To add a note for transport administration, choose .
7. To save the transport proposal, and then forward it automatically to transport administration, choose .

Confirming a Transport Proposal/Applying for Transport into the Next System

Confirming a Transport Proposal/Applying for Transport into the Next System








Use

Once the transport administration has approved your [transport proposal \[Ext.\]](#), the import is started and the transport proposal appears in your transport proposal inbox again. Check that the transport requests have been imported correctly into the target systems. If the transport was successful, you can either confirm the transport proposal, or apply for transport into other systems.


Prerequisites

- Transport administration has approved your transport proposal.
- You have a user in the [Workflow Engine \[Ext.\]](#) system/client.

Procedure

1. In the Change and Transport System, choose the tab page *Transport proposals*.
2. Choose  *Inbox*. Double-click the transport proposal that you want to confirm.
3. If there are any notes, you can display them by choosing .
4. Check whether the requests were imported correctly into the target systems. To do this, read the transport logs by choosing , and check the [return codes \[Page 58\]](#).
5. You have the following options:
 - To confirm the transport proposal, choose .
 - To apply for a transport into the next system(s), enter the target systems, import time and import options, as described under [Creating a Transport Proposal \[Page 65\]](#). To write a note for the transport administration, choose . Choose  to check your entries and then choose  to send it to transport administration.



If you are using the transport strategy [workflow-controlled transports \[Ext.\]](#), choose  to display all target systems that follow on directly from this one. You can use this function to distribute transports throughout your system landscape, system-by-system.

Result

- If you confirm the transport proposal it is stored in the database.
- If you have applied for a transport into the next system, the transport proposal is placed in the TMS worklist of transport administration for approval.

Confirming a Transport Proposal/Applying for Transport into the Next System

Finding Transport Proposals

Finding Transport Proposals



Use

To obtain an overview of the [transport proposals \[Ext.\]](#), you can use the search function.

Prerequisites

You have [configured the transport workflow \[Ext.\]](#) for your SAP System.

Procedure

9. To find a transport proposal, proceed as follows:
 - In the Transport Organizer, choose the tab page *Transport proposals*. To display the screen *Find Transport Proposals*, choose .
 - In the initial screen of the Transport Management System, choose *Monitor* → *Transport proposals*.
2. The field *Created by* is filled with the name of the current user as a default. You can change this name if necessary. You can also search for the date on which the transport proposal was created, and its short text.
3. Select at least one proposal status.
4. To start the search, choose .

Result

A list of proposals appears. You can expand this list and see which change requests have been imported into which transport targets. Double-click the proposal number to display the transport proposal, and then go to the action log and the notes accompanying the transport proposal.

Working with Projects

Purpose

If you [control your transports in projects \[Ext.\]](#) you can plan your development work and Customizing activities in structured projects. You can structure changes independently of each other in different projects, and then import them independently into target systems. We recommend that you do this if, for example, you want to go live with different projects at different times, or if you want to link development work in a single area.

Process Flow




1. First define an IMG project in [IMG Project Management \[Ext.\]](#) and then activate a corresponding CTS project.

In the IMG Project administration go to the tab page *Transport requests*, where you can:

- activate a CTS project
- Display the project data and change the description of the project
- Assign and edit CTS requests
- Display the CTS project piece list
- [Set the project status switch \[Page 76\]](#)
- Complete the CTS project

2. Choose *Assigned CTS requests*. The dialog box *Requests in project <project name>* appears.

Here you can:

- choose  to create a new request and assign it to a project
- choose  to assign an existing request to the project
- choose  to remove a request from a project

3. When you go from the IMG project to the corresponding Customizing activities, the [Transport Organizer \[Ext.\]](#) proposes only those change requests that you have previously assigned to the project.

The project piece list contains all objects that you are editing in a project. When requests are released, the object list of the request is added to the piece list.



Ideally, projects do not share objects. If your projects do share objects, you must [define dependency relationships between requests \[Page 73\]](#).

4. Released change requests are imported into the [QA system \[Ext.\]](#) project-by-project. This is where you test your changes and approve the requests. Requests are also imported into the production system project-by-project.
5. After you complete the project, you can no longer assign any change requests to it.

Working with Projects

The system informs you if there are any unreleased requests in the project that you want to complete.

You can reactivate the CTS functions even after you have completed a project. However, the lock flags are no longer present.



Once you have completed the project, the system no longer recognizes any overlaps it has with other projects. When you reactivate the project, the system recognizes overlaps with newly edited objects only.

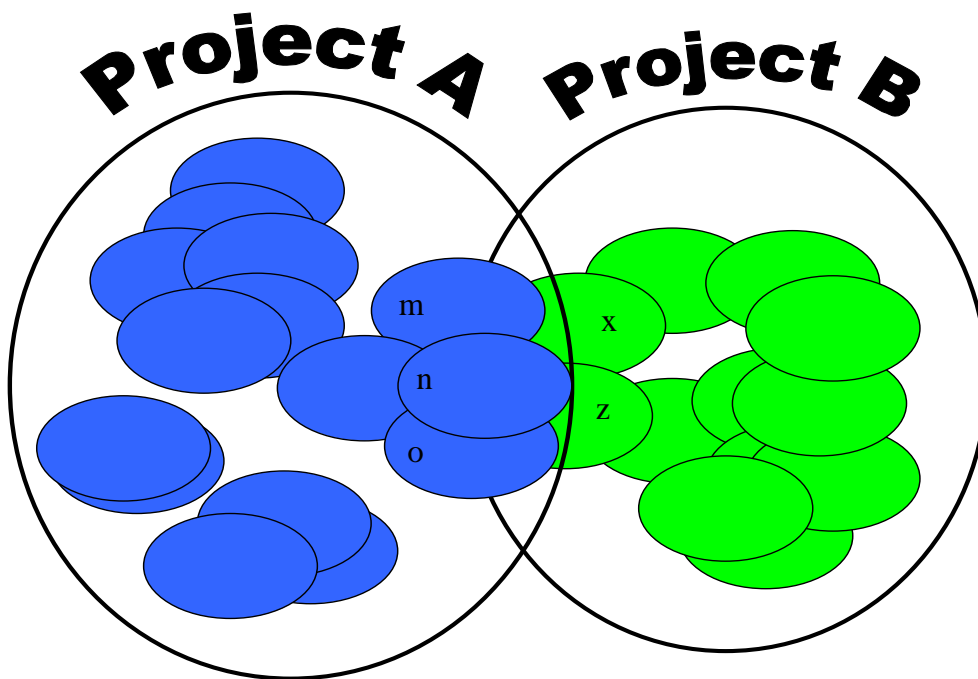
For more information, see [Importing All Requests from a Project \[Ext.\]](#) and [Attribute SAP_CTS_PROJECT \[Page 37\]](#).

Defining Dependency Relationships Between Requests

Purpose

If you use projects to control your transports, but your projects overlap, then you must define dependency relationships between the objects that belong to different projects but contain some of the same objects.

When you import requests, the system analyzes the dependency relationships and lets the transport administrator know that more requests have to be imported to ensure the correct sequence. This is just a warning though, and the import can be continued.



In this graphic, requests A_m , A_n and A_o overlap with requests B_x and B_z .

Prerequisites

You are working with [Transport Control Using Projects \[Ext.\]](#).

Process Flow

When you release a request, the system automatically checks whether the request's object list overlaps with the object lists of released requests from other projects that have already been released.

If requests do contain the same object, the system identifies which requests the object was transported in. A dialog box proposes these requests as predecessors of the current request.

Defining Dependency Relationships Between Requests

You can also edit the dependency information of a request yourself, and add predecessor and successor relationships independently of the system proposals. You can also create circular relationships, meaning that the requests have to be imported together.

To do this, go to the request overview in the Transport Organizer and choose *Goto* → *Dependency information*.

For more information, see [Importing All Requests in a Project \[Ext.\]](#).

Recording Changes in Overlapping Projects

If you [use projects to control transports \[Ext.\]](#), you can only use requests assigned to your current project to record changes.

When you edit [lockable objects \[Page 44\]](#), conflicts may arise if the object that you want to change is already locked in a request that is not assigned to the project you are working on. This lock prevents the object from being edited simultaneously in another request. You must also meet the requirements of the project functions.

A dialog box informs you if a conflict of this type has arisen. You can remove the conflict in one of the following ways:

- If the object that you want to edit is locked in a request that is not assigned to a project, then you can assign the request to your project. You must be the owner of the request to do this.
- If the object that you want to edit is locked in a request that is assigned to another project, then you can only record changes in the other request.

We recommend that you create [dependency information \[Page 73\]](#) between this request and the request you would have saved the changes in if the lock conflict had not arisen.

You can also add the object to the project object list of your current project if it is not already there. This makes the object a part of your current project.

Setting the Project Status Switch

Setting the Project Status Switch

Use

In certain situations (for example, to prevent tests in the QA system from being affected by development work), project administrators need to restrict what work can be done in a project. They can control this with the project status switch as follows:

- Change requests in a project cannot be recreated
- Change requests in a project cannot be released
- Change requests in a project cannot be imported

By default, the project status switches are set to *Allowed* for all systems.

Prerequisites

- You have the project administration [authorization \[Ext.\]](#) in the systems where you want to make settings.
- You are on one of the following screens:
 - in change mode in the IMG project administration transaction, have chosen *Transport requests*
 - in the Transport Organizer and have chosen *Goto → Edit project*.


Procedure

1. Choose *Project status switch*.

The dialog box *Project Status Switch: Select Systems* appears.

2. Either select the radio button *System* and enter your system, or select *All configured transport targets*. Choose *Continue*.

The screen *Set Project Status Switch* appears. Here you can make the following settings for the client or system:

- in the source client of the project
 - Allow or forbid requests to be created
 - Allow or forbid requests to be released
 - in all other clients/systems: Allow or forbid requests to be imported
3. To make the setting, expand the tree structure as required. The icon in front of the text shows you whether the setting is currently Allowed or Forbidden. To change the setting, choose .



If you want to set the project status switch for a system that you are not logged on to, the system displays the appropriate logon screen.

Version Management of Repository Objects

Use

Version management is active for all objects of the ABAP Workbench.

Version management serves different purposes and user groups:

- The developer can keep track of his or her work ("What have I changed?")
- The developer can restore a previous version by reactivating it.
- The system administrator can monitor work ("Which objects were changed and how in a specific time interval?")
- It provides a basis for any auditor who requires a complete history of changes.
- It enables customers to use the system to adjust data after they have upgraded to a new release.

Activities

You can access version management from the following:

- Object Navigator (SE80)
- Transport Organizer (SE09)
- Display and maintenance transactions for Repository objects.

Authorizations in Version Management

The authorization for accessing version management is covered by the authorizations for the ABAP Workbench and the Transport Organizer. It is only for the remote comparison with other SAP Systems that you require a special authorization.

To compare versions of with a remote system, use the user TMSADM, delivered by SAP. This user requires the authorization S_TMSADM_SHO (display authorization for Repository objects) in the remote system. This authorization is in the SAP authorization profile S_A.TMSADM of the user TMSADM.

If the user TMSADM does not have this authorization, then a logon screen appears when you use the remote version comparison. You can enter user and password on this screen. The user that you enter must have the authorization for displaying Repository objects. Otherwise the SAP System rejects the remote comparison.

The user SAPCPIC is used for the remote comparison with SAP Systems which have a release prior to 4.5A. The calling SAP System must have an RFC destination, in which the user is SAPCPIC is entered with the corresponding password.

This user requires the authorization S_RFC_VERS in the target system. This authorization is delivered by SAP in the authorization profile S_A.CPIC of the user SAPCPIC.

If you want to prevent a remote version comparison with a particular SAP System, then delete the authorization S_TMSADM_SHO from the profile S_A.TMSADM and the authorization S_RFC_VERS from the profile S_A.CPIC in the system in question.

To allow the remote comparison again, just add these authorizations to the appropriate profiles.

Creating Versions

The aim of version management is to record a complete change history of a Repository object. Versions are created automatically at the following times:

- Before a Repository object is changed

Each object changed is entered in a change request. If the latest version in version management is not the active version, the version of the object is saved in version management before it is changed. These backup versions are indicated in the version overview with an **S** or **I**.
- When a change request is released

When you release the change request with the changed objects, versions are created. The request number is displayed in the version overview for the relevant version.

A version is not created when objects are imported for performance reasons. This is not necessary, since the change history of the object was recorded fully in the development system. However, a comment is added to the version overview if an object has changed as the result of an import.

If you want, you can create extra versions of objects during import. This makes sense, for example, if the development system is reinstalled at regular intervals. This leads to the loss of the versions recorded there.

To create these versions, go to the Transport Management System and set the [profile parameter \[Ext.\] VERS_AT_IMP](#) to ALWAYS (the default is NEVER).

As well as the times mentioned above, versions are also created at the following times:

- Before the import

If the newest version in the version database does not match the active version, then a backup version is made immediately before the import. These versions are indicated in the version overview with an **S**.
- After the import

After the import a version is created of each imported object. The number of the imported request is displayed in the version overview for the created version. If an object exists in several requests that are imported at the same time (for example, with `tp import all`) only one version is made of this object, after the import of the final request.

You can use these versions to check what was the active status of an object at which time.

You can also set the SAP System so that versions are not created every time you import objects, just when you import objects as part of a relocation with development class change. To do this, set the profile parameter `VERS_AT_IMP` to `C_ONLY`.

In addition to the versions created automatically, you can also create temporary versions at any time. To do this, use the *Generate version* function in the maintenance transactions for the Repository objects. You can use these temporary versions to restore the previous version of an object, even while you are developing it. When a request is released, the temporary versions are deleted and replaced by the version active at this time.

Creating Versions

Displaying and Using Versions

Version Management

You can use version management to compare two versions of a Repository object, and display old versions of objects.

You can access version management whenever you are maintaining an object by choosing *Utilities* → *Version management*.

You can also access version management from the Transport Organizer (Transaction SE09):

- In the initial screen
 - Choose *Environment* → *Version management*. An overview of versionable objects appears. You can display individual object types and search for particular objects.
- In the request overview
 - Position the cursor on an object in the object list of a task or request and choose *Object* → *Versions*.

The version display is object-specific and is similar to the display used in normal object maintenance. Object versions can therefore be easily recognized.

Version Overview

The version overview shows the versions stored in the development database (Repository), as well as the historical versions stored in the version database. Both areas can be empty, such as in the following situations:

- An object has been newly created and no versions have yet been created.
- An object was deleted after versions were stored

Active and/or revised versions are stored in the development database.

The version overview also provides information on the time and release level when the versions were created.

If a transport request is specified for a version in the version database, this version corresponds to the state of the object

- when the request was released, if the request is from the current SAP System
 - or
- after the request was imported, if the request with which the object was transported is from another SAP System.

If a request is specified for an active or revised version, the object is currently being edited in connection with this request, or has been imported with the specified request.

The *Cat* column in the version overview specifies the reason why the version was created. The following values are possible:

Value	Description
-------	-------------

Displaying and Using Versions

" "	Version created when request was released
I	Version created during import
S	Version created due to a system request (for example, for a backup copy before inclusion in a correction or repair)
U	Version was created due to a user request (as an intermediate version) at some point in time. When the request is released, these versions are deleted and replaced by a " " version.

The *Fla* column in the version overview specifies how the version is flagged. The following values are possible:

Value	Description
" "	No special flag exists
I	The version last created is not the active version, since the active version was overwritten by an import

In the version overview, you can choose the following functions:

- Display a selected version
- Compare two selected versions
- Restore a selected version
This function requires change authorization for the Repository object concerned.
- Remote comparison with versions of the object in another SAP System (see the section [Authorizations in Version Management \[Page 78\]](#)).

Versions are displayed and compared in object-specific screens.

Archiving Versions

Prerequisites

If the version management data tables VRSX and VRSX2 take up too much space on the database, SAP recommends that you start archiving versions.

Archived versions are displayed in the versions overview, and are marked with an X in the column *Arch*.

If you choose the options *Display* or *Compare* for an archived version the SAP System restores the selected versions from the archive to the database, and then displays them.

You can also restore the complete archive files to the version database.



Archived versions can only be restored to the SAP System from which they were archived.

Procedure

To archive versions, or restore complete archive files from the database, proceed as follows:

1. In the initial screen of the SAP System, choose *Tools* → *Administration* → *Administration* → *Archiving*.
2. Enter the archiving object **VERSIONS**.

Result

You can choose from several functions for editing archives. You can also read documentation on these functions.

Functions in the Version Overview

Display:

The display mode of the ABAP Editor is used to display versions. You cannot, however, change versions here.

Using the editing functions such as the clipboard, you can copy a previous version, or parts of it, to another window.

Comparison:

This display compares two different versions that have some coding different and some identical.

The *Settings* function lets you choose the following layouts for the display:

- **Single-column or parallel**
 - Single-column:
Blocks with identical, changed, inserted, or deleted lines are shown below each other.
 - Parallel display:
The two versions are shown next to each other. The lines that differ are shown in color.
- **Display same program sections**
 - Display all:
Both versions are displayed in full.
 - Matches condensed:
Longer blocks with identical lines are shortened. Lines of programming that differ are displayed in full.
 - Display differences only:
Only the lines of programming that differ are displayed.
- **Line numbering**

You can choose whether you want to show or hide the line numbers. This is particularly useful for making the parallel display more concise.
- **Indentation and comments**

You can choose whether you want to ignore differences in comments or indentations in the comparison.

Retrieve:

The way in which the *Retrieve* function works depends on the object type:

- Dictionary objects: This function is activated when you call version management from object maintenance and are working in change mode. First a dialog box appears that explains the retrieve function. After you confirm the dialog box, the selected version is loaded into the editor. You can then use the editor to make more modifications. When you save these changes you overwrite the active version with the contents of the editor. For security, version management stores a temporary version of the previously active version. To change an

Functions in the Version Overview

object, you need a request. If you do not already have a request, you will be prompted to select or create a new one when you save your changes.

- Other Repository objects: You can always use the *Retrieve* function for these objects. A temporary version is generated of the active version. The active version is then overwritten by the retrieved version.