Transport Tools (BC-CTS-TLS)

HELP.BCCTSTLS

Release 4.6C

Copyright

Icons

Icon	Meaning
⚠	Caution
💬	Example
➡	Note
🧭	Recommendation
SYN	Syntax
💡	Tip

Contents

Transport Tools (BC-CTS-TLS)

Use

This documentation is intended for **system administrators**.

SAP provides utilities for controlling Release upgrades and transports among SAP Systems.

This section explains how to use the transport programs `tp` and `R3trans` to transport SAP system objects between SAP Systems.

Transport Control Program tp

Use

The transport control program `tp` is a utility for controlling transports between SAP Systems and for upgrading SAP Releases. As a control program, `tp` uses some special programs that are required to perform complete transports.

The transport control program `tp` is normally called by other programs:

- Change and Transport System (CTS)
- Transport Management System (TMS)
- Upgrade control program R3up

Therefore, you do not have to directly use `tp`.

Before you directly call `tp`, check if the desired function is not offered by CTS or TMS.

tp Attributes

The program `tp` offers several advantages over direct use of the programs it uses:

- `tp` keeps track of transports.
- `tp` exports and imports objects in the correct order.
- `tp` ensures that imports into a target system are done in the same order as the exports from the source system(s). If imports are processed out of order, it can result in severe inconsistencies in the target system, which are difficult to diagnose
- `tp` lets you perform exports and imports separately.

During an export, the objects to be transported are extracted from the database of the source system and stored in files of the operating system.

During the import, the objects are added to the database of the target system (according to the transport function recorded in the task).

Always perform an export immediately after the change request has been released, so that the objects can be further modified.

An import often cannot be executed immediately. To maintain performance, stability, and avoid conflicts with the online SAP System, perform the import at night.

`tp` is currently available for the following operating and database systems:

- Operating systems: UNIX, Windows NT, AS/400
- Database systems: Oracle, Informix, SAP DB, Microsoft SQL Server, DB2/400, DB2 Universal Database, DB2 for OS/390

`tp` supports a transport network between UNIX and Windows NT systems. For more information, see Transport Control Under Windows NT [Page 112].

Installing the Transport Control Program tp

Prerequisites

To use `tp`, you must meet the following prerequisites:

- The transport directory must be installed. See File Structure [Page 73]
- The SAP Systems in your network must have different names.
- The central transport profile must be up to date. See Transport Profile [Page 42].
- The target system (for imports) must have at least two background work processes.
- The source system (for exports) must have at least two background work processes.
- To call `tp`, you have to log on as `<sapsid>adm` or `<SID>OFR` (for AS/400) on a host in the corresponding SAP System (source system for exports or target system for imports).

Procedure

To install `tp`:

1. Prepare the SAP Systems [Page 10]
2. Prepare the operating system users [Page 11]
3. Prepare the file systems [Page 13]
4. Integrate new SAP Systems [Page 18]

Preparing the SAP Systems

To prepare your SAP Systems for `tp`:

1. Ensure that all SAP Systems have unique names. You can transport only between SAP Systems that have different names.

   If you have several SAP Systems with the same names, transporting is impossible because the target system of a transport is not unique. In addition, an object can be transported only once between two SAP Systems with the same name. After the first transport, the original version would be in both source and target systems.

   Do **not** use any of the following names as the system name:

   ADD, ALL, AND, ANY, ASC, COM, DBA, END, EPS, FOR, GID, INT, KEY, LOG, MON, NOT, OFF, RAW, ROW, SAP, SET, SGA, SH0, SID, UID, VAR

   T11 for the test system and P11 for the production system

2. Configure the transport routes according to your requirements. To do this, use the transport route editor in the Transport Management System.

3. Initialize the transport dispatcher.

   Start the program RDDNEWPP once in every SAP System. Do this as user DDIC in client 000 and in all clients that are used as the source or target for a transport. `tp` can then start the background job RDDIMPDP in every SAP System if it is needed to perform a transport (export or import).

   For more information, see Communication Between tp and ABAP [Page 86].

Preparing Operating System Users

Use

If you want to install `tp`, the operating system users must fulfill the following requirements.

Procedure

Operating System Users - UNIX

The means provided by the operating system are used to check the authorizations for the files in the transport directory. To transport in a group of systems connected for the purpose of transport, `tp` gives the group belonging to the owner of a file all authorizations for that file. Group `sapsys`, containing all the owners of the SAP Systems (`<sapsid>adm`), plays a central role here.

- All owners of SAP Systems (at UNIX level) must be in the group `sapsys`.
- The group `sapsys` must have the same group identification number (`gid`) on all the computers of the transport group. (This is suggested, but not guaranteed by the installation tools.)

Operating System Users - Windows NT

Each user that wants to use the transport system requires read and write authorization in the transport directory.

If you call a transport program directly from the operating system level, only use the operating system user `<SAPSID> ADM`.

If you call a transport program from the SAP System (Change and Transport Organizer or Transport Management System), the user under which the SAP System was started is used. This user is `SAPService<SAPSID>` for a standard installation of the operating system user.

To check under which user the SAP System was started, choose *Control Panel* → *Services* → *SAP<SAPSID>_<Sysno.>* → *Startup* → .

➡

For each SAP System that is part of the transport group, you must give read and write authorization to the following operating system users:

`<SAPSID> ADM` (for example, C11ADM)

`SAPService<SAPSID>` (for example, SAPServiceC11)

Operating System Users - AS/400

Each user who wants to use the transport system at the operating system level must be a group member in the group profile `<sapsid>OFR`.

This gives the user authorization to call transport programs. The programs themselves inherit the authority of user `<sapsid>OWNER`.

Note that at present user names and passwords must be exactly the same in all SAP Systems. This is a restriction required for remote access using `/QFileSvr.400` of OS/400.

Preparing the File Systems

Procedure

For `tp`, the following transport directories must fulfill certain prerequisites:

- Transport directory [Page 14]
- Transport subdirectories [Page 16]
- Transport subdirectory bin [Page 17]

Maintaining the Transport Directory

Procedure

For `tp`, the transport directory must fulfill the following requirements (depending on the operating system):

UNIX

In a transport network, there must be a single physical transport directory that can be reached from each host with the same path.

In UNIX systems, this directory can be a soft link, pointing to the correct transport directory or mount point.

You must store the chosen path in two parameters:

- Parameter `transdir` for the transport control program in the transport profile [Page 42]
- Parameter `DIR_TRANS` for the SAP System in the instance profile

The default value for the transport directory is `/usr/sap/trans` in UNIX.

Windows NT

Transport groups comprised of Windows NT systems or Windows NT and UNIX systems are supported.

For more information on creating a heterogeneous transport group, see Transport Control Under Windows NT [Page 112].

SAP allows you to name a central transport host on the Domain Name Server that is valid for all NT systems.

For more information on specifying a central transport host on the domain name server, see Specifying the Central Transport Host [Page 122].

Each transport group must have a single physical transport directory that can be accessed by all machines.

This directory is described by the parameters:

- `transdir` (transport profile)
- `DIR_TRANS` (instance-specific profile)

  If you have entered a transport host on the domain name server, the directory is under `\\$(SAPTRANSHOST)\sapmnt\trans` by default.

  Otherwise it is under `\\$(SAPGLOBALHOST)\sapmnt\trans` .

The transport directory is described by the following parameters:

- SAPTRANSHOST

  Name of the host configured as the central transport host on the domain name server

- SAPGLOBALHOST

  Name of the host on which the central system is installed

- sapmnt

  Global share that points to the `\usr\sap` file tree on the central instance

  ➡

  Note the different naming conventions used for the parameters `transdir` and `DIR_TRANS`:

  The path entered for the parameter `DIR_TRANS` does not end with a “\”, whereas the path entered for the parameter `transdir` does end with a “\”:

```
DIR_TRANS = T:
transdir = T:\
```

  ➡

  The `DIR_TRANS` profile parameter that you set and the parameter `transdir` in the transport profile must point to the same transport directory.

AS/400

In a transport network, there must be a single physical transport directory that can be reached from each host with the same path.

In AS/400 systems, this can be a soft link pointing to the correct transport directory or mount point.

You must store the chosen path in two parameters:

- Parameter `transdir` for the transport control program in the transport profile [Page 42]
- Parameter `DIR_TRANS` for the SAP System in the instance profile

  The default value for the transport directory is `/usr/sap/trans` in AS/400.

Maintaining the Transport Subdirectory

Procedure

The required subdirectories are `actlog`, `bin`, `data`, `log`, `buffer`, `sapnames`, `cofiles`, `olddata` and `tmp`. Additional optional subdirectories are `backup` and `serial`.

For `tp`, the transport subdirectories must fulfill the following requirements:

- **UNIX**

  In UNIX systems, the owner of all these subdirectories must be a `<sapsid>adm` user.

  All these subdirectories must also have read and write authorizations for the owner and the group.

- **Windows NT**

  In Windows NT systems, the subdirectories must have read and write authorizations for the users <SAPSID>ADM and SAPService<SAPSID> for each SAP System that is part of the transport group.

- **AS/400**

  In AS/400 systems, the subdirectories and the directory `/usr/sap/trans` must have the primary group `R3GROUP` or `<SAPSID>GROUP`.

Maintaining the Transport Directory bin

Procedure

For `tp`, the transport directory `bin` must contain the following files:

- the central transport profile [Page 42]. As of Release 4.5A, the Transport Management System administrates this file. (In releases **prior** to 4.5A, this file was called TPPARAM. As of 4.5A, this name is no longer used.) To maintain the transport profile, only use the Transport Management System.

  Under UNIX, you can create a soft link using the old name TPPARAM that points to the transport profile.

- `T_OFF.ALL` or `T_OFF.<SAPSID>` (possible). As long as such a file exists, exports from all systems or from a certain system (file name `T_OFF.T11`) are not permitted. Once the transport has been released, the first line of the file is displayed in the SAP System as an error message.

  In contrast to earlier releases, this subdirectory may no longer contain any executable programs.

  To ensure that you use the correct transport programs after each release upgrade, delete all the variants of the `tp` and `R3trans` programs from the directory. Both programs are taken from the executables directory on the application server.

Integrating New SAP Systems

To integrate a new SAP System to your transport network:

1. Change the name of the database and the SAP System if you have set up the SAP System as a copy of the database of another SAP System.
2. Identify the new SAP System to the existing SAP Systems. To do this, use the Transport Management System.
3. Define transport routes to and from the new SAP System. To define transport routes, use the transport route editor in the Transport Management System.

Controlling Imports using tp

Use

The program `tp` offers several advantages over direct use of the programs it uses:

- `tp` keeps track of transports.
- `tp` exports and imports objects in the correct order.
- `tp` ensures that imports into a target system are done in the same order as the exports from the source system(s).
- `tp` lets you perform exports and imports separately.

➡

The transport control program `tp` is normally called by other programs:

- Transport Organizer
- Transport Management System
- Upgrade control program `R3up`

Therefore, you do not have to directly use `tp`.

Before you call `tp` directly, check whether the function you want is offered by the Transport Organizer or TMS.

Functions

`tp` enables you to:

- Perform imports
- Transport client-specific data
- Perform special transports

The Route of a Request into the Target System

Local requests are not transported and are therefore not included in this description.

A request is transported as follows:

- Release of request

  The transport of a request begins when the owner of the request releases it (see the documentation on the Transport Organizer).

- Unlocking and export

  After a request has been released in the Transport Organizer, it is unlocked and the data is exported at operating system level.

  `tp` controls the export. `tp` also registers that the request has to be imported into the target system.

  The target system is specified by the system control tables. There is no automatic mechanism that immediately imports a request into a target system. Normally, the requests of several users are released from a source system before they are imported into the target system.

  The transport control program `tp` ensures that the requests that have to be imported are registered in the same order as the exports. In this way, several requests are collected and have to wait before they are imported into their target system.

- Import

  You must perform this import using the Transport Management System (Transaction STMS). This calls `tp`. (You can also directly call `tp` on the operating system level.) We recommend performing all administrative tasks regarding transports using the TMS.

Planning Imports

Use

As already explained in The Route of a Request into the Target System [Page 20], there is no automatic procedure that imports a request immediately after the export. Therefore, plan the time of the import to coordinate the actions required after the import (for example, testing functions in the target system).

Procedure

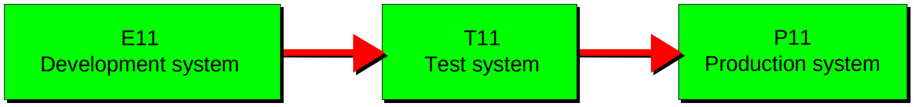
SAP recommends scheduling regular periods for complete imports into the target system (for example, daily, weekly or monthly). Shorter periods between imports are not advisable.

After the import into the target system, tests can be carried out there. You can correct errors by repeating transports from the source system.

Wherever possible, only correct errors in the source system. When the corresponding requests are released, they can be reimported with the command **`tp import all <sapsid>`** and are registered for import into further systems.

If you have defined further systems for the import, the requests are registered for import into these other systems after the main import has succeeded (see also Performing Imports [Page 23]).

If a subsequent system only receives a complete import after the test phase, the original request and the request containing corrections are imported in the correct order. This ensures that the discovered error does not occur in the target system.



In this example, perform the following steps during a development phase in the following order:

1.	Development in E11	(Monday - Thursday)
2.	Import into T11	(Thursday evening)
3.	Test in T11, correction in E11	(Friday)
	Repeated transport to T11	
4.	Import into P11	(Friday evening)

The time given between brackets is a suggestion for a development period that lasts one week. The production system will then be in a new, tested state at the beginning of the following week.

Performing Imports

Prerequisites

Imports are normally not performed by the developer who requested an export, but by the system administrator.

The administrator imports all change requests from a buffer of the SAP System into the target system. To do this, the administrator must log onto the operating system of a host in the target system as the “owner” of the system (<sapsid>adm for AS/400).

Procedure

You can start a complete import with the following command:

UNIX:

```
cd /usr/sap/trans/bin
tp import all <sapsid>
```

Windows NT:

```
cd \usr\sap\trans\bin
tp import all <sapsid>
```

AS/400:

```
cd ‘/usr/sap/trans/bin’
tp ‘import all <sid>’
```

The command processes all change requests in the <SAPSID> system buffer that have not yet been imported successfully. All necessary import steps are executed automatically.

Result

The steps in a complete import are as follows:

- **Import the command file with `R3trans`**

  The command file (the list of transport objects) is imported into the SAP System. This step is only executed if the transport profile parameter VERS_AT_IMP is not set to NEVER.

- **Creating Versions before the Import**

  In the SAP System, versions are created of the objects contained in the transport requests if there is no current version in the version database. This step is only executed if the transport profile parameter VERS_AT_IMP is not set to NEVER.

- **ABAP Dictionary Import with R3trans**

  ABAP Dictionary data is imported inactively. This makes it possible to import into an active SAP System.

- **ABAP Dictionary Activation**

  Nametabs are written only inactively. The SAP System can keep running until the activation step is completed. Enqueue modules are an exception.

- **Distribution Program**

  Once the activating program has executed the logical checks for the new ABAP Dictionary structures, the distribution program decides which actions are still needed to bring the new runtime objects into the running SAP System.

- **Structure Conversion**

  If changes to a table structure require their conversion, this is done here.

- **Moving the Nametabs (mvntabs)**

  In this step the new ABAP Dictionary runtime objects, which were only inactive up to now, are moved into the active runtime environment. The structures in the database are adjusted if necessary. Inconsistencies can occur in the SAP System beginning with the start of this step. These can only be corrected at the end of the main import.

- **R3trans Main Import**

  All the data is imported. The SAP System should be in a consistent state again when this step has ended successfully. If so, the automatic transport to other systems takes place.

- **Activation of Enqueue Modules**

  These objects cannot be activated in the same way as the rest of the ABAP Dictionary and therefore have to be activated after the main import in this separate step. They are then used directly in the running SAP System.

- **Structure Conversion of Matchcode Objects**
- **Import Application-defined Objects**
- **Creating Versions/Setting Version Flags**

  As of Release 4.5A, versions of the objects in the transport requests are created in this step if the transport profile parameter `VERS_AT_IMP` is not set to `NEVER`. Otherwise, version flags are set for all objects in this step.

- **Executing XPRA Programs**
- **Generating Reports and Screens**

  During this step, you can work in the SAP System again.

- **Removing Successfully Imported Requests from the Buffer**

  This step is not performed for single imports (`tp import <change request> <sapsid>`). These change requests remain in the buffer until the next complete import takes place or `cleanbuffer` is called and they are not shown when you call `showbuffer`.

  ➡

  In SAP Systems with Releases prior to 3.0, some of the steps described cannot be executed during operation. If the target system of an import request has a Release level lower than 3.0, these steps are not executed. In this case, use the call **`tp put <sapsid>.`**

  For a short description of the steps performed in target systems with Releases prior to 3.0, see Online Help for tp [Page 31].

➡

Generally, you can restart all `tp` import commands (*import*, *put*, ...).

If an error occurs during the import causing `tp` to cancel further actions, you can correct the error and restart the same `tp` call again. The transport control program `tp` records the point at which processing should restart. When `tp` uses the default parameters delivered, the import is interrupted whenever a step has a return code greater than **8**.

Automatically Starting Customer-Defined Actions

In the SAP System, there are 2 background events ("batch events") that are triggered by `tp` at the beginning and/or end of an import with `tp import all` and with `tp put`:

- SAP_IMPORT_START
- SAP_IMPORT_STOP

You can trigger customer-defined actions automatically using the import. Define background jobs that wait for one of these events. For example, you can reschedule and unschedule other background jobs using `BCTRNS1/2`.

> The continuation of the import does **not** depend on whether these events can be successfully performed and **not** on whether the jobs started by these events are successfully executed. The transport control program `tp` triggers these events, but keeps on running.
>
> Therefore, we recommend that you do not control any critical jobs using these events.

Transporting Client-Specific Data

Procedure

If client-specific routes are configured in the TMS (this means, if extended transport control is active), the following parameters are **automatically** set:

- ctc [Page 56]
- tp_version [Page 56]
- nbufform [Page 56]

If extended transport control is **not** activated, client-specific data, particularly Customizing data, is exported out of the source client and into the client with the same number.

If you do not want this, you can specify the target client for the import when you call `tp`:

```
tp import all <sapsid> client=<target client>
```

The client entered here refers to all client-specific objects in the requests registered for import.

If there are several source and target clients, making it impossible to assign transports on the basis of corresponding numbers, you must work with individual imports (also see Special Transports [Page 28]). To do this, you need to know the target clients of the individual requests.

- Request E11K904711 has to be imported into client 5, but all other requests registered have to be imported into client 6. To carry out the imports, use the following commands:

tp import E11K904711	<sapsid> client=5
tp import all	<sapsid> client=6

- The request E11K904712 has to be imported into the clients 11, 12, and 13:

tp import E11K904712	<sapsid> client=11 u0
tp import E11K904712	<sapsid> client=12 u0
tp import E11K904712	<sapsid> client=13

Performing Special Transports

Use

An import is performed with a single `tp` call that imports all waiting change requests:

```
tp import all <sapsid>
```

`tp` automatically ensures that change requests are imported in the correct order (in their export sequence).

However, in practice you often have to give individual change requests priority.

Prerequisites

Particularly with change requests, there are two aspects which can cause problems:

- Does the change request that is given priority overtake another change request that contains the same objects, but in an older version?
- In which order do the change requests have to be imported, if there is a delivery system?

If you are sure that no change requests with older versions of the same objects are overtaken, then these individual imports do not cause problems.

Otherwise, you have to ensure that the objects imported with an individual transport into the SAP System are not overwritten by an older version, when the regular import of all waiting requests is started.

Procedure

1. Perform individual imports with the **unconditional mode 0** (U0), except in the special situations mentioned above:

   ```
   tp import <change request> <sapsid> U0
   ```

   In this mode, the change request remains in the list of transports registered for import. Therefore, when the regular transport occurs, the request is automatically imported again.

   This is why change requests that are automatically forwarded to other SAP Systems are reimported, even if they have already been transferred to the systems as individual imports. This guarantees that export and import sequences are always the same.

2. Do not subsequently change an object that has been included in an individual import (for example, by repairs). Experience has shown that the authors of a change request do not expect their repairs to be overwritten by a repeated import. If necessary, inform the authors of individually imported change requests of the possible consequences.

➡

Limit special transports to when it is really necessary.

tp Information Commands

- `tp checkimpdp <sapsid>`

  The scheduling type for the transport daemon RDDIMPDP in the background processing of the SAP System <SAPSID>is displayed on the screen.

  `tp` also calls `R3trans`. `R3trans` also tries to log on to the database, and displays a message indicating whether this was successful.

- `tp connect <sapsid>`

  The transport control program `tp` tries to log on to the database of the <SAPSID> SAP System and informs you if it has been successful or not with a short message on the screen. If necessary, `tp` then logs off again.

- `tp count <sapsid>`

  The requests flagged for import into the <SAPSID> system are counted. The result is displayed.

- `tp go <sapsid>`

  The logon on of the various programs to the correct database is generally controlled by environment variables. `tp` computes the environment required for logging on to the database of the <SAPSID> system using the values in the global parameter file. Before `tp` logs on to the database, it executes the command **go**. This writes operating system commands to the screen that are executed to set the environment to correspond with the values `tp` has set for itself.

  ➡

  The command **go** does not include logging on to the database.

- `tp showinfo <request>`

  The header information of the `<request>` is displayed on the screen.

Calling Online Help for tp

The transport control program `tp` offers you online help that can be accessed at three different levels:

- To get a brief overview of the most important `tp` commands, including a reference to the second level, call `tp` without any arguments .
- The call **`tp help`** creates general syntax information for `tp` calls. You also receive a list of all `tp` commands and a reference to the third level.
- To display a description of the syntax and of this command function (**`<command>`** must be a valid `tp` command), call `tp` with **`tp <command>`** .

To find out the meaning of a particular `tp` return code, call `tp` with **`tp explainrc <rc>`**.

➡

The online help is only available in **English**.

tp: Additional Functions

In addition to the `tp` basic functions, you also have the following options:

- Cleaning up the transport directory [Page 33]
- Additional tp commands [Page 34]
- Additional tp options [Page 37]
- Synchronizing buffers [Page 40]

Only use the above mentioned commands and options in exceptional situations and only when you are very familiar with how the transport system works, otherwise undesired side effects may occur.

Cleaning Up the Transport Directory

Use

Over time, a large number of potentially large files can accumulate in the subdirectories `cofiles`, `data`, and `log` of the global transport directory. These contain important information, but they also require space and become obsolete.

Procedure

1. To generate a list of all deletable files in these directories, enter the command **`tp check all`**. Files are considered deletable when the accompanying request is not marked for import into an SAP System. This list is generated in subdirectory `tmp` and can be read.
2. To delete files from the subdirectories `cofiles`, `log`, and `olddata`, and to move files from the subdirectory `data` to the subdirectory `olddata`, enter the command **`tp clearold all`**. (This command works using the list generated in step 1.)

   ➡

   For more information on the parameters for these commands, see Transport Profile [Page 42]. To get the current values, use the command **`tp showparams <sapsid>`**.

tp Commands

- `tp addtobuffer <request> <sapsid>`

  As of Release **4.6A**, an explicit `addtobuffer` command leaves the position of the entry **unchanged**, if the relevant transport request is already in the buffer. Only the status of the entry is changed:

  – Unconditional modes appearing in the command line are added to the unconditional modes in the buffer.

  – All the steps of the entry are re-initialized.

  – If the entry found is not an initial entry, the import step for the command file is set to `*4000`.

  ⇨

  Before you call this command and its options, consider the effects of changing the import sequence.

- `tp cleanbuffer <sapsid>`

  This deletes successfully imported change requests from the list of requests that are marked for import into the SAP System <SAPSID>. This function is contained in the commands **`tp import all <sapsid>`** and **`tp put <sapsid>`**.

- `tp delfrombuffer <request> <sapsid>`

  If the specified request is marked for import into the specified SAP System, this flag is deleted from the import list.

  ⇨

  Before you call this command, consider the effects of changing the import sequence.

  ⚠

  As of Release 4.6A: If a transport request was only partially imported, you can delete this type of transport request from the buffer. However, if a deleted transport request is placed in the buffer (without having been initially or completely imported), the attributes of the old entry are reactivated. This means the return codes, the unconditional modes and tags of the deleted entry appear in the new entry.

- `tp delstopmark <sapsid>`

  If the list of requests marked for import into the specified SAP System contains a STOP mark, the mark is deleted.

- `tp export <request>`

  This exports the complete request from the source system. This command starts the export of a request from the operating system level. Only use this command in exceptional cases. The SAP System uses the command `expwbo` to release requests from CTS transactions.

- `tp reformatbuffer <sapsid> [options]`

This automatically reformats the **entire** buffer.

As of Release 4.5A, there are 2 different formats for buffer entries. (As of Release 4.5A, `tp` can work with both formats, and with a puffer containing heterogeneous entries. As of Release 4.6A, `tp` can work with 3 formats.)

However, in certain cases entries with the old formatting have to be reformatted during the normal run of `tp`. In this case, the entire buffer is automatically reformatted.

If you use the options `downtorelease` and `Iagree` for command `reformatbuffer`, you can format the buffer so that its formatting is compatible with an older release:

- Option `downtorelease <Release>`

  The command `reformatbuffer` with this option formats the buffer so that the formatting corresponds to the one used in an old release. This is necessary if `tp` writes buffer entries to the buffer of another system, but uses a buffer format that the transport control programs does not understand in this system.

  ⚠

  Since older formats cannot store the same amount of information as the newer ones, some information contained in the original buffer may not be stored in the unformatted buffer. This may lead to data loss.

  To avoid data loss, the original buffer is stored. In addition, you have the following functions:

  - Displaying the unformatted buffer (using `showbuffer <SID>`)
  - Displaying the original buffer (using `showbuffer buffer=<SID>SAV`)
  - Accepting the unformatted buffer (using `reformatbuffer` and the option `Iagree yes`)
  - Resetting the changes (using `reformatpuffer` and the option `Iagree no`)

  If you call `reformatpuffer` using `downtorelease` , `tp` displays in detail if information was lost by reformatting, and what that information was. `tp` also names the commands that is has to execute to accept or reject the changes.

  ➡

  If you use the option `downtorelease`, you must also specify the option `Iagree` with `yes` or no. You cannot use the transport system until you decide to accept or reject the changes.

- Option `Iagree`

  Here are the following values for this option:

Value	**Meaning**
`yes (true)`	**Accept** the change made by `tp reformatbuffer <SID> downtorelease <rel>` In this case, `tp` deleted the buffer semaphore file that was lifted for security reasons at the first call of `reformatbuffer`

`no` (`false`)	**Reject** the change made by `tp reformatbuffer <SID> downtorelease <rel>` In this case, `tp` copies the backup of the original buffer back and deletes the buffer semaphore file that was lifted for security reasons at the first call of `reformatbuffer`

- `tp setstopmark <sapsid>`

  A special STOP mark is added to the list of requests registered for import into the specified SAP System. The commands **`tp import all <sapsid>`** and **`tp put <sapsid>`** only process the requests **in front of** this stop mark.

  – If there is no stop mark, it is automatically generated by the commands **`tp import all <sapsid>`** and **`tp put <sapsid>`**.

  – When all the requests in front of the stop mark have been imported successfully, at the end of all the calls the related marks and the mark itself are deleted.

  A STOP mark can only be placed at the end of the change requests that have already been marked. In addition, it can only be set if no other mark of the same kind already exists.

- `tp showparams <sapsid>`

  This `tp` function displays which values the individual parameters of the transport profile have for the current SAP System.

- `tp verse <request>`

  This creates versions of the objects in the specified request. This transport step is contained in command `export`.

  As of Release 4.5A, more actions that prepare the export are performed in this step.

tp Options

You can assign additional options to all `tp` commands. To find out which commands understand which option, use the online help for the `tp` program. To do this, call `tp` with the desired command and do not specify any parameters.

Unconditional Modes

These modes are intended for special requirements in which various rules of the Transport Organizer and Transport Management System are ignored. An unconditional mode is represented by a digit between 0 and 9. To use the unconditional mode when calling the `tp` program, a **`U`** is added to the command line and all the required digits are appended to this **`U`**.

```
tp import T11k904711 P11 U06
```

The individual modes are:

0	Overtaker:	Import from the buffer without deleting and set unconditional mode 1 in the buffer to allow another import at the correct location.
1	During export:	Ignore the incorrect status of the command file.
	During import:	Ignore that the change request was already imported.
2	During export:	Do not expand the selection with TADIR brackets.
	During import:	Overwrite the originals.
3	During import:	Overwrite system-dependent objects.
6	During import:	Overwrite objects in unconfirmed repairs.
8	During import:	Ignore restrictions based on table classification.
9	During import:	Ignore that the system is locked for this transport type.

Clients

An export (at the operating system level) takes place from the export client by default. You can change this default value by explicitly defining the client in the command line of `tp`. Add either **`client<client>`** or **`client=<client>`** to the command line.

When requests are released from the SAP System, this client output is automatically filled with the logon client.

```
tp export T11K904711 client123
```

As described in Transporting Client-Specific Data [Page 27], you can configure client-specific transport routes (extended transport control is active). In this situation, additional options are required that describe the handling of clients during import.

Since extended transport control is switched on/off in the transport profile using parameter `ctc`, the following differentiates between `ctc=1` and `ctc=0` (ctc = client transport control).

Command Options

Command option CLIENT

Possible values: Numerical value of a client

`CTC=0`: All requests are imported in the specified client.

`CTC=1`: Imports requests that do not yet have a target client in the buffer into the specified clients. From the requests in the buffer that already have a target client, the requests are imported whose target client matches the specified client.

If `ctc=1`, the option `client` is handled like the entry `defautlclient` and `CLIRES` with the same client. (Example: `client=133` is handled like `defaultclient=133 clires=133`).

➡

Never specify the option `client` together with one of the `clires` options or `defaultclient`.

Command option CLIRES

Possible values: A list of clients divided by commas

➡

You can only use this option, if the transport profile parameter `ctc` is set.

The option `CLIRES` (client restriction) enables you to specify a list of clients divided by commas. This limits the command to a specific number of clients.

If you use the command **`import all`** and 3 different target clients appear in the import buffer, you can specify the option `CLIRES` to limit the effect of the command on the buffer entries for the specified clients.

Command option DEFAULTCLIENT

Possible values: Numerical value of a client

If the new transport profile parameter `ctc` is set, entries in the import buffer may already have specified its target client. If this is not the case, `tp` rejects the import of this entry, unless you specify for all these entry types the clients in which these requests are to be imported. Use the option `DEFAULTCLIENT` to make these entries.

Using a Special Parameter File:

By specifying the option `pf=<parameter file>`,`tp` uses the explicitly specified file as a transport profile.

You can use this option, for example, if you do **not** want to call the program from the default directory `/usr/sap/trans/bin` (UNIX) or `\\$(SAPTRANSHOST)\sapmnt\trans\bin` (Windows NT) containing the global transport profile.

Changing Individual Parameter Values

If you want to change individual parameter values for calling the `tp` program, forward the new value to `tp` using the option **`-D<value assignment>`**.

The `<value assignment>` must adhere to the same syntax as a line of the transport profile. For more information, see Transport Profile [Page 42].

```
tp import T11K904711 P11 “ -D buffreset=true“
```

AS/400:

```
tp ‘import T11K904711 P11 -D buffreset=true’
```

Synchronizing Buffers

Use

For performance reasons, data that is not changed very often in the database is held in internal buffers in the SAP System. This data is accessed in these buffers instead of it being read from the database each time. If buffered data is changed, the relevant buffer contents must be invalidated.

The procedure used for this is called *buffer synchronization*.

Two software components are involved in controlling this synchronization:

- Transport control program `tp` called by `R3trans`
- the SAP System

Procedure

Each of these components has parameters controlling their behavior:

In the transport profile, the parameter `buffreset` can have the values *1* or *0*. If the parameter is not specified explicitly in this file, `tp` takes the default value *1*.

SAP System

Synchronization is controlled in the SAP System with the profile parameter `rdisp/bufrefmode`. This parameter controls whether:

- The application servers of the SAP System write synchronization records (`sendon`) or not (`sendoff`)
- The various application servers react to synchronization records that they have written themselves (`exeauto`) or not (`exeoff`).

Values recommended for `bufrefmode` are:

- Central system (only one application server):

  ```
  bufrefmode = sendoff,exeauto
  ```

- An SAP System with several application servers:

  ```
  bufrefmode = sendon,exeauto
  ```

  ➡

  Also note the SAP profile parameter `rdisp/bufreftime`.

  It regulates the intervals at which the application server is checked to see if buffered data has been invalidated. Only when each application server has read the entry for the invalidation of the data is the changed date the same throughout the SAP Systems.

Known Problems

When changed ABAP Dictionary objects are imported, transactions may terminate. The same can occur when the respective changes are made in the SAP System. The transport programs `tp`/`R3trans` should be regarded as an additional application server.

As a result of structural changes, an instance may regenerate an ABAP program due to nametab changes before being able to synchronize its own nametab buffer. At runtime (now with a current nametab), the table work area no longer matches the structure used at generation time. The program terminates with `GETWA_CANT_CLEAR`.

Solution

Postpone imports of transports with changed ABAP Dictionary objects to a time when there is not much system activity, for example at night.

Transport Profile

Definition

The transport profile is a global parameter file for the program `tp` and is administrated by the SAP System using the Transport Management System (TMS).

The profile is in the transport directory `bin`.

The profile name corresponds to the convention `TP_<domain>.PFL`, where `<domain>` is the name of the transport domain configured in TMS. (You can find the domain in the initial screen of Transaction *STMS*.)

Use

You control the program `tp` using the transport profile.

Each time `tp` is started, it must know the location of the global parameter file. You can specify the location with the option **`pf=...`** (see the section *Using a Special Transport Profile* under tp Options [Page 37]).

UNIX:

```
tp pf=/usr/sap/trans/bin/TP_DOMAIN_PRD.PFL import all P11
```

Windows NT:

```
tp pf=\\<HOST1>\sapmnt\trans\bin\TP_DOMAIN_PRD.PFL import all
P11
```

AS/400:

```
tp ‘pf=/usr/sap/trans/bin/TP_DOMAIN_PRD.PFL import all P11’
```

If the transport profile is not specified in this way, `tp` searches for the transport profile in the current directory.

Structure

The transport profile contains the following information:

- Databases from various target systems
- Parameters describing the frequency of the transport
- Additional information for system maintenance

Every line in a transport profile that is not preceded by a # contains a parameter definition.

The various parameters in the transport profile have the following validity:

- Global (for all SAP Systems in your network)
- Local (only for one SAP System)

- Operating system-dependent
- database-dependent

If a parameter does not have a specified value, the SAP System uses the default value. Global parameters override defaults. Local parameters override global parameters. You can therefore specify a parameter value for all but one SAP System and a different value for one special system.

Parameters that you specify with the option *-D* (command line call) deactivate parameters of the transport profile (see tp Options [Page 37]).

Global

Pure parameter definitions such as **<parameter>=<value>** are always evaluated. However, you can limit the validity of the SAP System parameters in different ways. You can set restrictions for a particular SAP System, an operating system or database system. `tp` reads transport profile from beginning to end and skips all the entries that are not relevant. If a parameter for the current environment (<SAPSID>, operating system, and database system) occurs more than once, the last value read is used.

SAP System-Specific

If a parameter is preceded by an SAP System name, this assignment is only valid for this SAP System. The way in which the SAP System name is separated from the parameters indicates that the name used in the prefix is the name of an SAP System. SAP System names and parameters must be separated by a "/" character: **`<sapsid>/<parameter>=<value>`**

```
C11/dbhost=host001
```

Operating System-Specific

You can also restrict the use of a parameter value to a certain operating system. You must place an acronym for the name of the operating system in front of the parameter. Here, the special separator is the "|" character: **`<cpu>|<parameter>=<value>`**

```
wnt|transdir=....
```

Acronyms that are supported for the different operating systems are: `aix`, `hp-ux`, `osf1`, `sinix`, `sunos`, `wnt` (Windows NT), and `as4` (AS/400)

Database-Specific

You can restrict the use of the value to a particular database by placing an acronym for the name of the database system before the parameter. Here, the special separator is the ":"character.

```
<db>:<parameter>=<value>
```

Acronyms that are supported for the different database systems are:

`ora` (Oracle), `inf` (INFORMIX), `ada` (SAP DB), `mss` (Microsoft SQL Server), `db2` (DB2 for OS/390) `db4` (DB2/400), `db6` (DB2 Universal Database).

Variables in the Transport Profile

In the transport file, you can use variables by entering parameters.

Enter these variables in the following format: **`$(xyz)`**. You can mask the brackets with the “\” character, if required.

The following variables are available:

- `$(cpu)`

  The CPU name (`alphaosf`, `hp`, `rm600`, `rs6000`, `sun`, `wnt`, `as4`) This can be important for operating systems in heterogeneous networks.

- `$(cpu2)`

  This variable contains an acronym for the name of the operating system. The same acronyms as those used for specifying operating-system-specific parameters are used: `aix`, `hp-ux`, `osf1`, `sinix`, `sunos`, and `wnt`.

- `$(dname)`

  Abbreviation for the day of the week (SUN,MON,...)

- `$(mday)`

  Day of the current month (01-31)

- `$(mname)`

  Abbreviation for the name of the month (JAN,FEB,...)

- `$(mon)`

  Month (01-12)

- `$(system)`

  Name of the SAP System

- `$(wday)`

  Day of the week (00-06, Sunday=00)

- `$(yday)`

  Day of the current year (001-366)

- `$(year)`

  Year (four digits)

- `$(syear)`

  Short form of the year (two digits)

- `$(yweek)`

  Calendar week (00-53)

Windows NT

- `$(SAPTRANSHOST)`

  If you have identified a central transport host on the domain name server (DNS), use the entry `transdir` only with this acronym.

Parameter TRANSDIR

You must define the parameter `TRANSDIR` identically for all SAP Systems in the transport environment.

`TRANSDIR` specifies the name of the transport directory which must be mounted on all the computers of the transport environment and must be available there under the same name. All relevant data for a transport between any SAP Systems is stored in subdirectories of the transport directory and managed by `tp`.

> This parameter value is overridden with the value of the SAP System profile parameter `DIR_TRANS` for all `tp` calls from the SAP System.

If you want start `tp` from the operating system, ensure that the transport file entry managed by the TMS is correct. If necessary, make any required changes. For more information, see Changing the Transport Control Program Parameters [Ext.].

Values for Parameter TRANSDIR Dependent on the Operating System

UNIX and AS/400

```
TRANSDIR=/usr/sap/trans/
```

Windows NT

When using a central transport host with ALIAS SAPTRANSHOST:

```
TRANSDIR=\\(SAPTRANSHOST)\sapmnt\trans\
```

otherwise:

```
TRANSDIR=\\<transport host>\sapmnt\trans\
```

Parameters for Setting Up the Database Connection

Each SAP System in the transport environment needs parameters that enable `tp` to log on to the databases belonging to the SAP Systems. These parameters depend on the particular database system and should therefore be defined specifically for each SAP System.

`tp` identifies the database system type on the basis of the parameter `dbtype`, whose default value corresponds to the acronym of the database. The value of this parameter can be one of the acronyms allowed for specifying database-specific parameters.

The following applies for all databases:

`CONNECT_MODE`

Default: Not set

Possible values: `use_env`

You can use this parameter to stop `tp` from setting environment variables. `tp` requires some environment variables to set up the connection to the database.

If you have not set `CONNECT_MODE`, the parameters set in the transport profile are evaluated to set these environment variables correctly.

If you set `CONNECT_MODE = use_env`, the environment variables are not changed. In this case, you need to set the appropriate parameters correctly at the operating system level.

`dblibpath`

UNIX

Default: `/usr/sap/$(system)/SYS/exe/run/`

Possible values: Paths

This parameter lets you inform the transport control program of the value of the directory in which the dynamic SAP database interface is located. In the standard system, these files are always in the Executable directory of the application server. This means that you do not have to change the default value. The transport control program `tp` derives the value of the environment variables `dir_library` from the value of this parameter.

Windows NT

Under Windows NT, the parameter `dblibpath` is not evaluated.

The dynamic SAP database interface is searched in the following sequence:

a. In the directory in which the started executable `tp.exe` is located

b. In the work directory from which the program `tp` was called

c. In the Windows NT directory

d. In the path

DB2 for OS/390

The following parameters are required for DB2 for OS/390 databases:

➡

Note that how you use these parameters depends on your operating system (AIX, Windows NT or OS/390 Unix).

- `dbhost`

  Operating system: AIX, Windows NT

  Value range: Text

  Here you can enter the host name where the database runs.

- `icli_signon_file`

  Operating system: AIX, Windows NT

  Value range: Text

  Default: `/usr/sap/<SAPSID>/SYS/exe/run/iclipt`

  This parameter specifies the absolute path to the Signon file. This file is required to perform PassTicket Signon.

  You can find more information in the IBM document *Planning and Repairing R/3: DB2 for OS/390*.

- `icli_trusted_connections`

  Operating system: AIX, Windows NT

  Value range: Text

  Default: Not set

  This parameter enables you to switch off the PassTicket Signon function. If the parameter is not set, the ICLI client assumes that the ICLI server demands PassTicket Signon and behaves accordingly. If the parameter is set to 1, PassTicket Signon is switched off.

  You can find more information in the IBM document *Planning and Repairing R/3: DB2 for OS/390*.

- `r3_db2_ssid`

  Operating system: AIX, Windows NT, OS/390 Unix

  Value range: Text

  Default: SAP system name (<SAPSID>)

  This parameter specifies the DB2 group attachment name if this name differs from the SAP System name (<SAPSID>).

  You can find more information on this topic in *R/3 Installation Guide for UNIX/Windows NT - DB2 for OS/390*.

- `r3_port`

  Operating system: AIX, Windows NT

  Value range: Text

  Default: Port number that is specified under the service name `sapdb2<SAPSID>` in the file `/etc/services`.

  This parameter specifies the ICLI connection port.

You can find more information in the IBM document *Planning and Repairing R/3: DB2 for OS/390*.

- `r3_servicename`

  Operating system: AIX, Windows NT

  Value range: Text

  Default: `sapdb2<SAPSID>`

  This parameter specifies the service name that defines the ICLI connection port in the file `/etc/services`. The transport profile parameter `r3_port` overwrites this parameter.

- `rsdb_iclilibrary`

  Operating system: AIX, Windows NT

  Value range: Text

  Default: `/usr/sap/<SAPSID>/SYS/exe/run/ibmiclic.o`

  This parameter specifies the absolute path to the shared library of the ICLI client.

  You can find more information on this topic in *R/3 Installation Guide for UNIX/Windows NT - DB2 for OS/390*.

- `dbs_db2_planname`

  Operating system: OS/390 Unix

  Value range: Text

  Default: Not set

  This parameter specifies the DB2/390 plan name of the dynamic SAP database interface.

- `rsdb_db2pslibrary`

  Operating system: OS/390 Unix

  Value range: Text

  Default: Not set

  This parameter must be set. It specifies the absolute path to the library of the DB2 Performance Monitor.

DB2 Universal Database

The two parameters `dbname` and `dbhost` are required for DB2 Universal Databases:

- `dbhost`

  Value range: Text

  Here, you can enter the host name where the database processes run.

  In Windows NT, it is the TCP/IP name of the host.

- `dbname`

  Value range: Text

Here, you can enter the name of the database instance.

You must also install the DB2 Universal Database Client Application Enabler software on the host where `tp` runs.

DB2/400

Only `dbhost` is required for AS/400 databases:

Value range: Text

Here, you can enter the system name of the host where the database runs.

If you use TCP/IP instead of `opticonnect`, you must also specify the following:

```
opticonnect = 0
```

INFORMIX

The two parameters `dbname` and `dbhost` are also needed for INFORMIX databases with a standard installation:

- `dbhost`

  Value range: Text

  Here you can enter the host name where the database runs.

- dbname

  Value range: Text

  Here, you can enter the name of the database instance. The name is case-sensitive.

Generally, it is sufficient to specify the parameters `dbname` and `dbhost`. Default values are defined for the following Informix-specific parameters. In this case, you do not have to specify them in the transport profile.

- `informixdir`

  Default: `/informix/<sapsid>`.

  Value range: Text

  Here, you can enter the directory name where the database software is located.

- `informixsqlhost`

  Default (UNIX): `$(informixdir)/etc/sqlhosts[.tli|.soc]`

  Value range: Text

  This parameter is used to specify the name of the SQL hosts file with a complete path.

- `informix_server`

  Default: `$(dbhost)$(dbname)shm`

  Value range: Text

  Here, you can enter the database server name for a local connection.

- `informix_serveralias`

  Default: `$(dbhost)$(dbname)tcp`

Value range: Text

Here, you can enter the database server name for a remote connection.

Microsoft SQL Server

The two parameters `dbname` and `dbhost` are also needed for SQL Server databases:

- `dbhost`

  Value range: Text

  Here, you can enter the TCP/IP host name where the database runs.

- `dbname`

  Value range: Text

  Here, you can enter the name of the database instance.

Microsoft SQL Server has 2 ways of connecting:

1. SQL Server authentication

   SQL Server user and password are defined (`sapr3`, `sap`)

2. Windows NT authentication (integrated security)

   No user or password is necessary. The NT security descriptor is used and the NT user connects itself.

As of Release 4.5A, Windows NT authentication is used by default. For `<SID>adm` and `SapService<SID>`, the corresponding logins are created during the upgrade and the installation (see Note 116735). The parameters `mssql_user` and `mssql_passwd` are **no** longer evaluated in normal circumstances.

However, by setting the environment variable `MSSQL_USEINTEGRATEDSECURITY` to zero, you can cause the process to log on via SQL Server authentication. If a `tp` user is running under a non-`<SID>adm`, the user can work with a SQL Server database by setting **`MSSQL_USEINTEGRATEDSECURITY=0`** via a user name and password. You can also define a Windows NT login for this user with the SQL Server Enterprise Manager. We recommend using this option.

- `mssql_user`

  Value range: Text

  Here, you can enter the connection to the database for a specific user.

- `mssql_passwd`

  Value range: Text

  The parameter `mssql_user` lets you enter the password for a specific user here.

ORACLE

Only the parameter `dbname` is required for Oracle databases, if your installation does not differ to the SAP standard. Other parameters relevant to the configuration of the database connection are `dbconfpath` and `dblogicalname`. Due to the Oracle software used, ORACLE-NET-V2 is used to connect to the database.

The NLS (National Language Support) and TNS (Transparent Network Substrat) files are installed in the directory ORACLE_HOME. Oracle only supports the installation with this configuration. (Oracle uses TNS to implement the SQL*NET V2 database connection.)

SAP's own database interface is in a dynamic library that is linked to the user only at runtime.

- `dbhost`

  Value range: Text

  Here, you can enter the host name where the database processes run. In Windows NT, it is the TCP/IP name of the host.

  This parameter is no longer required for Release 4.x SAP Systems.

- `dblogicalname`

  Default: `dbname`

  Value range: Text

  Here, you can enter the logical name that identified the instance in the Oracle network during configuration.

- `dbconfpath`

  Default: `$transdir`

  Value range: Text

  Possible values: Paths

  You can use this parameter to inform `tp` as to where the configuration files for ORACLE SQL*NET V2 are located. The default value `$transdir` mirrors the fact that SAP has already stored these files centrally in the transport directory. From this parameter, `tp` derives the value of the environment variables `TNS_ADMIN`. However, if the parameter `DBSWPATH` is set, `dbconfpath` is ignored.

  This parameter is not evaluated under Windows NT.

- `dbname`

  Value range: Text

  Here, you can enter the name of the database instance.

  In a standard SAP System installation, it is enough to specify the parameter `DBNAME` in the parameter file. The standard installation uses the name of the SAP System for the name of the database instance, and for the logical name of the database in the ORACLE network.

- `dbcodepage`

  Default: `american_america.us7ascii`

  Value range: Text

  Here, you can enter the code page of the database instance.

The Oracle databases are installed with the code page `we8dec`. In this case, you must enter the value `american_america.we8dec` in the transport profile.

When upgrading a Release from 3.x or a later Release to a 4.x, the code page `us7ascii` is retained.

- `dbswpath`

  Value range: Text

  Possible values: Paths

  This parameter lets you inform the transport control program of the value of the directory in which the Oracle client software is installed. If the parameter is set, `tp` derives the following environment variables:

  ```
  ORACLE_HOME = $(dbswpath)
  TNS_ADMIN = $(dbswpath)/network/admin (on UNIX)
  ORA_NLS = $(dbswpath)/ocommon/nls/admin/data (on UNIX)
  ORA_NLS32 = $(dbswpath)/ocommon/nls32/admin/data (on UNIX)
  ORA_NLS33 = $(dbswpath)/ocommon/nls33/admin/data (on UNIX)
  ```

  This parameter is not evaluated under Windows NT.

SAP DB

The parameters `dbname`, `dbhost`, and `dbuserkey` are needed for SAP DB databases with a standard installation:

- `dbhost`

  Value range: Text

  Here you can enter the host name where the database runs. In Windows NT, it is the TCP/IP name of the host.

- `dbname`

  Value range: Text

  Here, you can enter the name of the database instance.

- `dbuserkey`

  Here, you can enter the name of the SAP instance.

The relevant connection information is read from the `xuser` file after the `userkey` entry. The `xuser` file format is stored in file `$HOME/.XUSER62` for UNIX and in the Registry for Windows NT.

The database connection is established using the following strategies:

- `dbuserkey` is set

  The connection occurs using the corresponding `userkey` entry in the `xuser` information.

- `dbuserkey`, `dbname` , and `dbhost` are not set

  The database logon uses the default entry (`DEFAULT userkey`) from the `xuser` information. These are installed for the local system in a standard installation.

- `dbuserkey`, is not set; `dbname` and `dbhost` are set

  The database logon uses the default entry in the `xuser` information, whereby the database instance and host are overridden by the values entered for `dbname` and `dbhost`.

For the test import in a system group, you must define additional `userkeys` for all `xuser` information in the group.

1. Log on as user `<sidadm>`.
2. Enter the `xuser` password of the userkey DEFAULT: **`sap`**
3. Use function key `F4` to search for free userkeys and add the user parameters for `userkey`:

USERKEY:	<Instance name of the SAP System>
USERID:	sapr3
PASSWORD:	<Database password of sapr3>
repeat passw:	<Database password of sapr3>
SERVERDB:	<Name of the database instance>
SERVERNODE:	<Name of the host on which the database is running>
SQLMODE:	SAPR3
CACHELIMIT:	1
TIMEOUT:	0
ISOLATION:	0

4. Store the changes with the function key `F5`.

Global Parameters

- `alllog`

  Default: `ALOG$(syear)$(yweek)`

  Value range: Text

  Here, you specify the name of a file in which `tp` stores information about every transport step for each change request. The file is always stored in the transport directory `log`.

- `ctc`

  Default: `0`

  Value range: Boolean

  `ctc` stands for *Client Transport Control* and represents a method of configuration for transport routes. This parameter lets you set specific transport routes using various clients in your system landscape. This method requires an extension of the import buffer. This new format is not understood by old `tp` programs. Therefore, this new function is switched off by default and the user must explicitly turn it on.

  At the same time, the profile generator from the Transport Management System is also delivered. Along with the transport route configuration in the Transport Management System, the transport profile parameter `ctc` is automatically switched on from the TMS if it is required for your transport landscape.

  ⚠

  The new format of the import buffer, activated by `ctc=1`, requires at least a `tp` from Release 4.5A. Older `tp` versions destroy this new import buffer.

  When parameter `ctc` is set, `tp` requires that you set the value for parameter `tp_version` to at least **`264`**. This prevents transport control programs with lower version numbers from working with this import buffer. As a result, you can only set the new parameter `ctc` if all the transport control programs that work with a specific SAP System have at least version 264.

  Client category definition enables new options during import, and the general handling of transport requests and target systems. There are also the command options [Page 37] `clires` and `defaultclient`.

  If you set the parameter `ctc`, this changes the output of the command `showbuffer`, because the corresponding target client must also be part of the output.

- `nbufform`

  Default: `0`

  Value range: Boolean

  Transport request names may have a length of 20 characters. However, these transport requests cannot be processed in the previous format of the import buffer. If you set the new parameter `nbufform`, you can switch this format so that transport requests with names between 10 and 20 characters can be processed. As with parameter ctc, you have to protect this change to the import buffer format from transport control programs that are too old.

> ⚠
>
> The new format of the import buffer, activated by `nbufform=true`, requires at least a transport control program from Release 4.5A. Older `tp` versions destroy this new import buffer.

When parameter `nbufform` is set, `tp` requires that you also set the value for parameter `tp_version` to at least **`264`**. This prevents transport control programs with lower version numbers from working with this import buffer. As a result, you can only set the parameter `nbufform` if all the transport control programs that work with a specific SAP System have at least version 264.

- `new_sapnames`

  Default: `0`

  Value range: Boolean

  A file is created for each user of an SAP System in the transport group in the `sapnames` subdirectory of the transport directory. The file is given the name of the user. However, non-valid user names in the SAP System may be used as file names in some of the operating systems. Special characters such as " ","."، etc. or the length of the file name permitted could cause problems.

  `tp` offers a solution. The user names are modified to create file names that are valid in all operating systems. The original user name is stored in the corresponding file.

  The prerequisite is that all the SAP Systems in the transport group have at least Release 3.0.

  > ⚠
  >
  > Once you have activated this option by setting `new_sapnames` to `true`, you may never again set this parameter to `false` (unless you also delete all the files in the `sapnames` subdirectory of the transport directory).

- `repeatonerror`

  Default: 9

  Value range: Numeric

  Parameter `repeatonerror` has an effect similar to parameter `stoponerror`, but `repeatonerror` specifies the return code up to which a change request is considered to be successfully processed. Return codes less than `repeatonerror` are still accepted. Change requests that were not completely processed successfully stay in the buffer.

- `sli`

  Since SAP Release 4.5A, you can use the option `selectivelanguageimport` (`sli`) to control the selective language import. The default for this option is `sli=no`. If `sli=yes`, you make sure that only the language-dependent data is imported whose languages exist in the system.

⚠

In this case, it makes sense if the number of languages selected (language vector at export) contain all the languages of the target systems during the import. If this is not the case, `R3trans` generates a corresponding warning at import.

➡

The languages in the system are the ones for which language imports were already performed. Languages that were **not imported** are not part of the system and are not overwritten.

This option is used as of Release 4.5B when importing Support Packages (Transaction SPAM).

- `stopimmediately`

  Default: `1`

  Possible values: Boolean

  This parameter is connected with the parameter `stoponerror`.

  If one of the tools (R3trans, DD Activation, etc.) called by `tp` outputs a return code that is equal or higher to `stoponerror`, `tp` sops immediately.

  If `stopimmediately = 0`, `tp` lets the current import phase be completed. If serious problems occur, this can lead to the same error messages reoccurring (for example, database problems). If temporary problems occur, this can lead to sequence problems. Therefore, we recommend **not** changing the default value.

- `stoponerror`

  Default: 9

  Value range: Numeric

  If `stoponerror` is set to 0, `tp` is never stopped in the middle of an `import` or `put` call. If `stoponerror` is set to a value greater than 0, `tp` stops as soon as a change request generates a return code that is equal to or greater than this value. Change requests that still have to be processed for the current step are first completed. However, a `syncmark` in the buffer of the relevant SAP System functions as a limit here.

  If you call `tp showparams <SID>`, you see that two variables have been set. `stoponerror` itself is treated as a Boolean variable that determines if `tp` is stopped if the return code is too high. The numeric value of the `stoponerror` parameter is stored in the variable `badrc`.

- `syslog`

  Default: `SLOG$(syear)$(yweek).$(system)`

  Value range: Text

  Here, you name a file in which `tp` stores information about the progress of import actions for a specific SAP System. The file does not store information for any particular change request. The file is always stored in the transport directory `log`.

- `tp_version`

Default: `0`

Value range: Version numbers

If this parameter is set to a value not equal to zero, a lower version of `tp` is not allowed work with this transport profile. When the default value (0) is set, the parameter has no effect.

- `vers_at_imp`

  Default value: `never`

  Possible values: `never`, `c_only`, `always`

  If you set the parameter to either `c_only` or `always`, additional steps are started during the import that generate versions of the imported objects in the target system. This concerns the commands `import` and `put`. For more information on these additional steps, see the `tp` Online Help for both commands. Also call `tp` with either the command `import` or `put` without specifying any other options.

  The additional steps are:

  - Before importing the ABAP Dictionary sources, the transport request is imported (command file Import).
  - Then a version maker is called in the batch to store the object versions before the import into the versions database. Versions are only then created if the current version of the object is not the same as the latest version in the database, or if a version does not yet exist for the object in the versions database.
  - The version maker is called at a point in the sequence of the import steps where the version maker was already called to set the flags on the imported objects in the versions database. If `vers_at_imp` is configured differently from `never`, real versions are written to the versions database at this point.

  The difference between the values `c_only` and `always` concerns the handling of various transport methods. If the parameter `vers_at_imp` has the value `always`, all transports are made into a version according to the new method. The value `c_only` limits the new behavior to transports of originals.

  Transports of originals are:

  – C transports: Relocation without changing development class (for a normal consolidation transport, the object must be transported back again).

  – O transports: Relocation with change of development class. (In future, the object can be transported from the original system with a consolidation transport. The development class must also be changed during the transport.)

  – E transports: Relocation of a complete development class (you can specify a new transport layer as an option).

Parameters for Distributing Transports

Logging on to the SAP System

An important `tp` parameter is the name of the SAP System which `tp` should log on to. This is not a parameter that can be set in the transport profile. You set it implicitly by making entries in the command line when `tp` is called. The name of the SAP System is included as one of the arguments in the command line, or it can be determined from the first three characters of the change request specified (for export calls).

Additional Parameters

- `dummy`

  Default value: 0

  Value range: Boolean

  You can use this parameter to make a dummy system known in the global parameter file. Imports are not performed for an SAP System with `dummy=1`.

- `testimport`

  Default value: 1

  Value range: Boolean

  The parameter `testimport = 0` prevents the transport control program `tp` from performing a test import after the export (for example, when the target system is “remote”). You must specify the parameter for the source system of the transport request.

- `testsystems`

  Value range: Text

  You can define a list of up to 50 SAP Systems with this parameter. The names must be separated by commas. If the export of a change request is successful, `tp` determines the test systems that have been specified for the target system of the change request and places the request in the buffers of these test systems.

Parameters for Controlling R3trans

- `filesplit`

  Default: `0`

  Strings, some of which may be defined or a number

  You can distribute, or split, the data from a transport request over several files. This may be necessary, for example, if you are exporting very large transport requests. To distribute the data, activate the parameter `filesplit`. The default value `0` deactivates the new function so that all files that are exported can also be processed with older transport tools.

  The values `off`, `false` and no have the same meaning as the default value `0`. If you want to set the parameter to a numerical value other than `0`, the export data is split into files of this specified maximum size.

  ➡

  Values between `0` and `1048576` (=1 MB) are handled as if they were `1048576`, since smaller data sizes are not useful.

  If you enter one of the values on, `true` or `yes` instead of a numerical value, `tp` assumes that a numerical value was specified that corresponds to 1 GB-1 (2147483647).

  The names of the generated files are determined and only depend on the name of the transport request. If the transport request is called C11K904711, for example, then the files are named in the sequence they were generated, as follows: R904711.C11, R904711_0.C11, R904711_1.C11,...,R904711_Z.C11,R904711_00.C11,...,R904711_ZZ.C11. A maximum of 1298 continuation files is possible.

- `language`

  Default: Empty text

  Value range: Text

  If this value is specified, it is transferred directly to `R3trans`. If `R3trans` does not receive a value for the languages to be exported, it exports language-dependent data from all the languages that it can find. If the parameter `language` is transferred, `R3trans` tries to export the languages specified in this text. Each language is identified by a single letter. You can specify several languages by entering a sequence of letters.

- `lsm`

  – `lsm = master`

    The **master language** is stored for each object in the object directory (table TADIR). This is the language in which the object was created or is maintained. In this mode, the language-specific data in the master language is transported, however, not translated texts. This master language does not have to be contained in the language vector (LANGUAGE).

  – `lsm = translation`

Only the translated data is transported in this mode, not however entries in the master language. Only languages contained in the language vector are considered.

– `lsm = nomaster`

  As for `translation`, languages not contained in the language vector are also considered.

– `lsm = vectorandmaster`

  All entries belonging to the original language or contained in the language vector are transported.

  All the functions mentioned are based on the original language of a text object being defined. This applies for all text parts of the Repository objects. There are exceptions for the remainder objects that are not listed here.

- `r3transoptions`

  Default: Empty string

  Value range: Text

  Possible values: `R3trans` options

  This parameter lets you use the transport control program `tp` to transfer any `R3trans` options to `R3trans`. Since you can specify several `R3trans` options using a transport profile parameter, note the following when using this parameter:

  Procedure for Transferring R3trans Option `op_name = op_wert`:

  a) First, check if an R3trans option is already set for the target system.

     Change to the subdirectory `bin` of the transport directory and enter the following command:

     ```
     tp showparams <TSID> parname=r3transoptions
     ```

     `<tsid>` is the target system name. The output from this command is normally empty. If your transport profile already contains `R3trans` options, this is the output. (You can ignore warnings or error messages about unknown or invalid parameters.)

  b) Add the output from the above-mentioned command to the new `R3trans` option (divided by blank space), and use this new value for the transport profile parameter `r3transoptions`.

     Example for the option `safeandslow=yes`

     Assuming the command above returns `rawdocu=no`, then the new value for the parameter would be `r3transoptions: rawdocu=no safeandslow=yes`.

- `r3transpath`

  Default: `R3trans`

  Value range: Text

This parameter is used to transfer the complete name of the program `R3trans` to `tp`. The default value is not a complete path specification. The operating system and the settings of the operating system are used to find the correct value.

Because all transports are performed by the SAP System user on the operating system level, the default value is correct.

Parameters for the Transport Daemon

- impdp_by_event

  Default value: true

  Value range: Boolean

  If the parameter is set to true, tp tries to activate the transport daemon in the SAP System whenever it is needed. Otherwise the transport daemon must run periodically and tp waits for the daemon to fulfill its communications task.

  The transport daemon is the ABAP report RDDIMPDP. Another ABAP report is provided for periodic scheduling: RDDPUTPP.

  If you set the parameter impdp_by_event to false, you have to execute the report RDDPUTPP once in each SAP System of your transport environment. Do this in client 000 as user DDIC. If it is set to true, another ABAP report is provided. RDDNEWPP schedules the transport daemon in the background processing, enabling tp to start it by generating a specific event. To do this tp needs further parameters. Either parameter system_nr and system_pf, or sapevtpath must be set. Under Windows NT, the parameters system_pf and sapevtpath must always be set.

- sapevtpath

  Default value: sapevt

  Value range: Text

  If you set impdp_by_event to true, this parameter contains the complete program path for the specific SAP System that can send an event to the SAP System. The program is normally called sapevt and is in the directory where all the programs belonging to an SAP System are stored. When you transport with tp, work as the owner of the SAP System (from the operating system point of view) because you do not have to change the default value.

- system_nr

  Value range: Text

  This parameter gives tp the instance identification number of the relevant SAP System. Therefore, you must specify the value for each SAP System specifically. The value is needed if you set impdp_by_event to true so that tp can send the event to the transport daemon, which triggers its execution.

- system_pf

  Value range: Text

  This parameter gives tp the complete path for the profile of the relevant SAP System. The value may be needed, if you set impdp_by_event to true. tp can then send the event to the transport daemon, which triggers its execution.

Parameters for Buffer Synchronization

- `buffreset`

  Default value: `true`

  Value range: Boolean

  For performance reasons, the SAP System buffers special data on the database in the main memory of the respective application server. If this data is changed on the database, this must be communicated to the application servers. This also applies for changes that are made by the transport tools `tp` and `R3trans`. However, for special requirements, you can switch off the synchronization mechanism.

  Normally, the synchronization mechanism is used for the data changed on the database. `tp` resets all the buffers of the SAP System for some steps during a `tp put` call so that all the application servers can react to changes in the database. Buffer synchronization takes several seconds until all the application servers have read the request to reset all buffers. You can set the repetition period for reading the synchronization requests with parameters in the SAP System (`rdisp/bufreftime`). You can therefore set the parameters for the time `tp` waits until it can assume that all the buffers have been reset. See the description of the parameter `bufreftime`.

- `bufreftime`

  Default value: `180`

  Value range: Numerical

  The value of this parameter specifies how long `tp` should wait in seconds until it can assume that all the buffers of the SAP System have been synchronized. The value of this parameter must correspond to the `rdisp/bufreftime` profile parameter of the relevant SAP System.

Parameters for the tp Function PUT

- lockuser

  Default: 1

  Value range: Boolean

  During a tp put call, various users can log on to the SAP System at three different points. If the parameter lockuser is set to 0, nothing happens at these points. Otherwise, an indicator is set in the database at the beginning and in the middle of a tp put call, which only permits the users DDIC and SAP* to log on to the SAP System. Users that have already logged on are not affected (this would be the only reason for activating the parameters startsap and stopsap). The indicator is deleted at the end of the call, allowing all users to log on to the SAP System again. To find out the exact locking and unlocking locations, enter the command tp put without any arguments.

- lock_eu

  Default: 1

  Value range: Boolean

  When tp put is called, the system change option may be changed at two locations. If the parameter lock_eu is set to 0, the system change option remains unchanged. Otherwise, the system change option is set to *Objects cannot be changed* at the beginning of the call, and is reset to its previous value at the end of the call. To find out the exact locking and unlocking locations, enter the command tp put without any arguments.

- startdb

  Default: Space

  Value range: Text

  tp uses the value of this parameter to start the database of an SAP System. It interprets the value as the complete path specification of a program that is able to start the database of an SAP System. However, it is not necessary for tp to start the database of an SAP System at the customer. Therefore, SAP has defined the default value of the parameter so that the SAP System is not started at the customer.

  This parameter is not active under Windows NT.

- startsap

  Default: Space (Windows NT)

  Value range: Text

  tp uses the value of this parameter to start an SAP System. It interprets the value as a complete path specification for a program that is able to start an SAP System. For the customer, however, it is not necessary for tp to start (or stop) an SAP System. Therefore, SAP has defined the default value of the parameter so that the SAP System is not started at the customer.

- stopdb

Parameters for the tp Function PUT

Default: Space

Value range: Text

`tp` uses the value of this parameter to stop the database of an SAP System. It interprets the value as the complete path specification of a program that is able to stop the database of an SAP System. However, it is not necessary for `tp` to stop the database of an SAP System at the customer. Therefore, SAP has defined the default value of the parameter so that the SAP System is not stopped at the customer.

This parameter is not active under Windows NT.

- `stopsap`

  Default: Space (Windows NT)

  Value range: Text

  `tp` uses the value of this parameter to stop an SAP System. It interprets the value as a complete path specification for a program that is able to stop an SAP System. However, it is not necessary for `tp` to stop an SAP System at the customer. Therefore, SAP has defined the default value of the parameter so that the SAP System is not stopped at the customer.

If you want to activate the parameters `startsap`, `stopsap`, `startdb`, and `stopdb`, set them as follows:

- UNIX

```
STARTSAP = startsap R3

STOPSAP = stopsap R3

STARTDB = startsap db

STOPDB = stopsap db
```

- Windows NT

```
STARTSAP =
\\$(SAPGLOBALHOST)\sapmnt\$(system)\sys\exe\run\startsap.exe
name=<SID> nr=<SYSTEM NR>

STOPSAP =
\\$(SAPGLOBALHOST)\sapmnt\$(system)\sys\exe\run\stopsap.exe
name=<SID> nr=<SYSTEM NR>
```

If you want to start/stop a system on a remote machine, you must also set the parameter `SAPDIAHOST`. In this case, extend the transferred parameter for `startsap` and `stopdb` as follows:

```
name=<SID> nr=<SYSTEM NO> SAPDIAHOST=<HOST>
```

⇨

The parameters `startdb` and `stopdb` are not active under Windows NT.

Parameters for the tp Function CLEAROLD

- `cofilelifetime`

  Default: `365`

  Value range: Numeric

  If a file in the `cofiles` subdirectory is no longer needed for further actions in the transport system and has reached a minimum age, you can delete it using the call combination **`tp check.all`**, **`tp clearold all`**. You can set the minimum age in days using this parameter.

- `datalifetime`

  Default: `200`

  Value range: Numeric

  If a file in the subdirectory `data` is not needed for any further actions of the transport system and has reached a minimum age, it is moved to the subdirectory **`olddata`** using call combination **`tp check.all`**, **`tp clearold all`**. You can set the minimum age in days using this parameter.

- `loglifetime`

  Default: `200`

  Value range: Numeric

  If a file in the `log` subdirectory is no longer needed for further actions in the transport system and has reached a minimum age, you can delete if using the call combination **`tp check.all`**, **`tp clearold all.`** You can set the minimum age in days using this parameter.

- `olddatalifetime`

  Default: `365`

  Value range: Numeric

  If a file located in the `olddata` subdirectory is no longer needed for further actions in the transport system and has reached a minimum age, you can delete it using the call combination **`tp check.all`**, **`tp clearold all`**. You can set the minimum age in days using this parameter.

  ➡

  The minimum age refers to the date the file was created and not to the date on which the file was copied to the directory `olddata`.

Example of a Parameter File

The following is an example parameter file, showing how to use variables and set parameters.

UNIX

```
##############################################################
#   Template for UNIX                                  #
##############################################################
# First we specify global values for some parameters,              #
# later the system-specific incarnation of special parameters       #
##############################################################
##############################################################
#                Global Parameters                             #
##############################################################
transdir                = /usr/sap/trans/
dbname                  = $(system)
 ##############################################################
#                  System-specific Parameters                    #
##############################################################
##############################################################
#              T11                                            #
##############################################################
T11/dbhost          = Tabc123
##############################################################
#              P11                                            #
##############################################################
P11/dbhost          = Pabc123
```

Windows NT

```
##############################################################
#            Template for Windows NT                   #
##############################################################
# First we specify global values for some parameters,              #
# later the system specific incarnation of special parameters       #
##############################################################
##############################################################
#                Global Parameters                             #
##############################################################
transdir = \\Pabc123\sapmnt\trans\ or
transdir = \\$(SAPTRANSHOST)\sapmnt\trans\
dbname = $(system)

##############################################################
#                  System-specific Parameters                    #
##############################################################
##############################################################
#              T11                                            #
##############################################################
T11/dbhost          = Tabc123
##############################################################
```

```
#                  P11                                                          #
###############################################################
P11/dbhost          = Pabc123
```

Ensure that the name of the database host entered (parameter `<SAPSID>/dbhost`) is not the computer name, but the TCP/IP name of your host.

You can find the TCP/IP name of your host under

> *Start → Settings → Control Panel → Network → Protocols → TCP/IP Protocol → Properties → DNS.*

The parameter `system_pf` is only needed for an event-controlled start of the transport daemon RDDIMPDP.

AS/400

```
###############################################################
#              Template for AS/400                        #
###############################################################
transdir                  = /usr/sap/trans/
R3transpath               = /sapmnt/exe/R3trans
sapevtpath                = /sapmnt/exe/sapevt

###############################################################
#              F3C                                                          #
###############################################################
F3C/dbhost               =as0009
F3C/opticonnect          =1
```

How tp Works

`tp` is a C program that runs independently of the SAP System. It reads and writes various files. “File” means a file in the operating system sense, not a database table.

All the files `tp` works with are located in a special transport directory. `tp` starts specific programs, some of which are C programs, some of which are special operating system commands, and some of which are ABAP programs.

File Structure for tp

On all operating systems, `tp` requires a special file structure for transport data, logs, temporary data, and transport control data. The file structure is operating system-dependent, but SAP has tried to keep these differences as small as possible.

`tp` needs a transport directory, which is called `/usr/sap/trans` under UNIX. You can also use any other directory in the file system, but it must exist on every machine in your network where an SAP instance is running (UNIX: mounted, Windows NT: Share). If this is not the case, you cannot install a transport system that runs automatically.

Under Windows NT, the default path of the transport directory is `\\$(SAPTRANSHOST)\sapmnt\trans`.

You can find more information on how to address and rename the transport directory in the section Preparing the File Systems [Page 13].

The transport directory must have the following subdirectories:

- `bin`

  Contains the global transport parameter file

- `data`

  Contains the transport data

- `log`:

  Contains transport logs, trace files, and statistics

- `buffer`:

  Contains control information on which transports are to be imported into which SAP Systems and the order of the imports

- `cofiles`:

  Contains information on change requests (different steps of a change request and their exit codes)

- `sapnames`:

  Contains information for SAP users (the status of each change request)

- `tmp`:

  Contains temporary files (logs, semaphores on files)

- `actlog`:

  Contains action logs for all tasks and requests These files are only changed by the SAP System.

- `olddata`:

  Contains “old” files from other directories that are saved here with the **`clearold`** command before being deleted by a later execution of this command

File Naming Conventions for tp

SAP designed `tp` taking into consideration that it has to run on many different operating systems. For this reason, `tp` requires restrictive file naming conventions. The naming restrictions ensure that file names are the same in different systems, allowing you to navigate more easily in heterogeneous networks.

Buffer Names

The buffer for the SAP System T11 is, for example, represented by a T11 file in the buffer directory.

Log File Names

All log files are located in the log directory. The name of a log file consists of the name of the change request, the executed step, and the SAP System in which the step was executed:

```
<source system><action><6 digits>.<target system>
```

Possible values for `<action>` are:

- **A:** Activate Dictionary
- **D:** Import application-defined objects
- **E:** `R3trans` export
- **G:** Generate reports and screens
- **H:** `R3trans` Dictionary import
- **I:** `R3trans` main import
- **L:** `R3trans` Import the command file
- **M:** Activate enqueue modules
- **P:** Test import
- **R:** Execute XPRA programs after PUT (XPRA)
- **T:** `R3trans` import of table entries
- **V:** Set version flag
- **X:** Export application-defined objects

There are also logs for the transport steps that are not executed for a special request. In this case, the log names follow another convention:

```
<action> <date>.<target system>
```

The date has the format YYMMDD. There are two positions each to define the year, month, and day. The following actions are affected by this:

- **C:** Copying the SAP Kernel
- **DS:** Distribution program flow
- **N:** Structure conversion

- **O:** Starting/stopping the database
- **P:** Moving the nametabs
- **S:** Starting/stopping the SAP System

Data File Names

All data files are located in the transport directory `data`. The name of a data file consists of the name of the change request, and a code letter. The code letter distinguishes between data files generated by `R3trans` or by application programs for application-specific development environment objects (ADOs):

```
R<6 digits>.<source system>  R3trans
D<6 digits>.<source system>  application programs
```

Change Request Information File

The change request information file contains information about a change request, including the transport type and the class of the objects to be transported. It also has information on the steps required for the change request, exit codes, and the time of execution.

The information file is located in the `cofiles` directory. The name is derived from the name of the change request:

```
K<6 digits>.<source system>
```

Example of File Names for tp

For the change request T11K123456 with target system P11, the transport file names are as follows:

- Data files:

  UNIX / AS/400:

  `$(transdir)/data/R123456.T11` and
  `$(transdir)/data/D123456.T11`, if necessary

  Windows NT:

  `$(transdir)\data\R123456.T11` and
  `$(transdir)\data\D123456.T11`, if necessary

- Buffer:

  UNIX / AS/400: `$(transdir)/buffer/P11`

  Windows NT: `$(transdir)\buffer\P11`

- Information file:

  UNIX / AS/400: `$(transdir)/cofiles/K123456.T11`

  Windows NT: `$(transdir)\cofiles\K123456.T11`

- Log files:

  UNIX / AS/400: `$(transdir)/log/T11E123456.T11` (export)

  `$(transdir)/log/T11P123456.P11` (test import to P11)

  `$(transdir)/log/T11I123456.P11` (import to P11)

  `$(transdir)/log/T11A123456.P11` (activation in P11), ...

  Windows NT: `$(transdir)\log\T11E123456.T11` (export)

  `$(transdir)\log\T11P123456.P11` (test import to P11)

  `$(transdir)\log\T11I123456.P11` (import to P11)

  `$(transdir)\log\T11A123456.P11` (activation in P11), ...

tp: Return Codes

The transport control program `tp` collects all return codes that occur when a `tp` command is executed. From all of these single return codes, a combined return code is calculated that `tp` outputs as a result at the end the command. During an import, for example, all return codes for individual import steps are handled. If a special return code does not occur, then the combined return code at the end is the maximum of individual return codes that occur. In addition to the value of the return code, `tp` also gives a short description of this return code. You can display this short description using the command **`explainrc`**.

Normally, all return codes that occur originate from individual transport steps. Since these individual return codes can only have the following values 0, 4, 6, 8, 12, 13, 14, 16, the combined return code also has one of these values. However, if a general error occurs, a single return code over 200 is generated. For example, this could arise if `tp` cannot establish a connection to the database. The maximum return code for all single transport steps is output in a separate message before specifying the combined returned code. In this case, the combined return code is always the return code of the general error message.

Logs of the Individual Transport Steps

There are transport steps that cannot be assigned to a single transport request. The logs of this transport step could not be displayed previously from the SAP System. (Except for using Transaction *AL11*, if you know the name of the log file.) However, the return codes of this transport step are considered when calculating the combined return code.

This can lead to `tp` returning a combined return code of 8, although all the transport request logs displayed by the SAP System have a lower return code.

The names of the affected files are described in File Naming Conventions for tp [Page 75].

R3trans Calls

R3trans is called with the following command:

- UNIX

  fork()

- AS/400

  spawn()

- Windows NT

  CreateProcess()

The global parameter file determines which R3trans is used for each SAP System. If the parameter r3transpath is incorrectly specified, no exports or imports can be performed. The implemented default value r3transpath = R3trans or for Windows NT r3transpath = R3trans.exe is sufficient.

The complete path must be specified for AS/400. It is a symbolic link pointing to the program R3transdb4 in the SAP kernel library, for example, R346COPT, which has to be installed locally. The value must be /usr/sap/$(SYSTEM)/SYS/exe/run/R3trans.

For more information, see Parameter TRANSDIR [Page 47].

Utility Calls

The operating system-specific differences are very large for several utility functions (for example, starting and stopping the SAP System and databases). Sometimes differences even exist between various SAP Systems.

You can enter the utility functions as strings in the global parameter file (parameters `startsap`, `stopsap`, `startdb`, `stopdb`). These strings are executed by `tp` with `system()`.

Log Hierarchy

You can monitor the transport actions carried out in an SAP System using the log hierarchy, which has three levels.

- Level 1 File SLOG

  This file gives an overall view of all imports running in the SAP System.

- Level 2 File ALOG

  This file contains individual return codes for all the transport steps performed in the transport directory.

  A short status report is also entered for each export or import carried out in the transport directory.

- Level 3 Logs of the individual transport steps

Log File SLOG

Definition

This file gives an overall view of all imports running in the SAP System.

The file SLOG is in the `log` subdirectory of the transport directory. You can define the name in the transport profile (file in the `bin` subdirectory of the transport directory).

The default value is `SLOG$(syear)$(yweek).$(system)`. Two characters represent the current year and two characters the current week.

For more information on naming conventions, see File Naming Conventions for tp [Page 75].

Structure

The entries in the file are represented in the form of a table. All lines are divided into nine columns separated by blank characters. Lines that do not fit into the table structure may also appear in the general overview log.

- Line type (5 characters):

  This entry defines the meaning of the columns. The following values are possible:

  – START

    The following columns describe an action that was started. Actions can be nested in other actions.

    START lines of component actions can appear before the STOP line of the current action. The actions involved are then described in the following column. The actions `put`, `imp single`, and `imp all` only appear at the highest nesting level. The only action that can appear at the third nesting level is `tp_getprot`.

  – STOP

    The action that began in the corresponding START line has ended.

  – ERROR

    An error occurred executing an individual step. In this case the relevant line from the file ALOG is duplicated in the following eight columns (see Log File ALOG [Page 84]).

  – WARN

    Warnings that cannot be assigned to a special, complete log are written to the file SLOG.

    The line after the first column consists of unformatted text. A typical example of such warnings is described in Note 12746.

  – LIST

    This line can only appear when the command **`put`** is used. Here, **`tp`** enters the steps that are necessary for executing the **`put`** command (unformatted, from the 5th column onwards).

  – HALT

`tp` has encountered a situation that makes it impossible to continue executing the current command. This line has its own formatting: `HALT (time stamp)`. The subsequent line states the reason for the HALT message as free text and is followed by a line that specifies the process identification number.

- (Free text)

  All other text that appears is unformatted output that is self-explanatory.

- Change request/action (20 characters):

  If **`START`**, **`STOP`**, or **`LIST`** is entered in the first column, this column contains a short description of the action.

  If **`ERROR`** is the value in the first column, this column contains the name of the change request. This and the following columns are then formatted in the same way as the ALOG file.

  The name of the change request can also be **`ALL`**.

- Name of the SAP System (3 characters):

  The name of the SAP System in which the transport step was performed is in this column.

- Description of the transport step (1 character):

  Each transport step is uniquely identified by a character. This character is also included in the name of the complete log.

- Return code (4 characters) interpreted as a number
- Time stamp (14 characters) with the format YYYYMMDDHHmmss
  - four characters for the year (YYYY),
  - two for the month (MM),
  - two for the day (DD),
  - two for the hour (HH),
  - two for the minutes (mm) and
  - two for the seconds (ss).
- SAPUSER (12 characters), the owner of the change request
- Operating system user (12 characters) who called up the transport tool `tp` Normally, this should be `<sapsid>adm`.
- Name of the machine (indefinite length) on which the transport tool ran.

Log File ALOG

Definition

This file contains individual return codes for all the transport steps performed in the transport directory.

The file ALOG resides in the `log` subdirectory of the transport directory. You can define the name in the transport profile (file in the `bin` subdirectory of the transport directory). The default value is `ALOG$(syear)$(yweek)`. Two characters refer to the current year and two characters to the current week.

For more information on naming conventions, see File Naming Conventions for tp [Page 75].

Structure

The entries in the file are represented in the form of a table. All lines are subdivided into eight columns that are separated by a blank character.

- Change request (20 characters):

  Most transport steps are assigned to a particular change request. In this case, the change request is in this column. If a transport step cannot be assigned to a change request, **`ALL`** is always entered in this column.

  Whenever the import command is called, the conversion program checks if one of the imported change requests contains a modification to the ABAP Dictionary that requires a table conversion. The table involved is then converted.

- Name of the SAP System (7 characters):

  The name of the SAP System in which the transport step was performed is in this column.

  With extended transport control (see parameter `ctc` in Global Parameters [Page 56]), the target of a transport step is a client in an SAP System and no longer just an SAP System. The combination of system and client is described here as a logical SAP System.

  The value of this column can have 3 characters. In this case, the value specifies **only** the SAP System. If the value has 7 characters, a logical system in the form `<SID>.<CLI>` is given. `<SID>` is the system name and `<CLI>` the client name. If the transport step affected all the clients, the client can also have the value **`ALL`**.

  Another possible form is `<SID>:<CLI>`. In this case ('**:**' instead of '**.**'as separator), the transport request contains client-specific objects that were handled in the specified client, and cross-client objects whose import affected all the clients.

- Description of the transport step (1 character):

  Each transport step is uniquely identified by a character. This character is also included in the name of the complete log. For more information on naming conventions, see File Naming Conventions for tp [Page 75].

- Return code (4 characters) interpreted as a number

- Time stamp (14 characters) with the format YYYYMMDDHHmmss

- four characters for the year (YYYY),
- two for the month (MM),
- two for the day (DD),
- two for the hour (HH),
- two for the minutes (mm) and
- two for the seconds (ss).

- SAPUSER (12 characters), the owner of the change request
- Operating system user (12 characters) who called the transport tool `tp`. Normally `<sapsid>adm` is entered here.
- Name of the machine (indefinite length) on which the transport tool ran

Communication Between tp and ABAP

Transport steps that are implemented in ABAP include activating ABAP Dictionary objects, structure conversion, and generating reports and screens. Executing these steps is complicated. `tp` uses the tables TRBAT and TRJOB to communicate with the ABAP transport programs.

To execute these steps, the SAP System must be running with at least two background work processes. Furthermore, the batch job RDDIMPDP must be scheduled with the repeat-start option. We recommend scheduling this job event on a regular basis. If the job is not running, then start the program RDDNEWPP once in every SAP System. Do this as user DDIC in client 000 and in all clients that are used as the source or target for a transport.

Once `tp` has sent the corresponding event to the SAP System, RDDIMPDP checks table TRBAT to see if there is anything to do. If there is, it starts the required programs (for example, mass activation) in the background, schedules itself again, and then stops.

RDDIMPDP automatically recognizes if a previous step was terminated and restarts the action. However, this can only be done with two background work processes.

RDDIMPDP recognizes terminated import steps by checking table TRJOB.

Table TRBAT

An example is the best way to understand how `tp` and ABAP communicate and the task of table TRBAT.

Assume there are three change requests which are to be activated. For each change request, `tp` adds an entry into table TRBAT.

TRKORR	Function	RETCODE	TIMESTMP
T11K004711	J	9999	00000001
T11K004720	J	9999	00000002
T11K003456	J	9999	00000003

The table columns have the following meaning:

- TRKORR

  Contains the names of the change requests

- FUNCTION

  Contains a key for the requested action. Possible values are:

  - **A:**

    Activate all ABAP Dictionary objects in the change request with mass activation (old activation program).

  - **B:**

    Activate all ABAP Dictionary objects in table TACOB with mass activation.

  - **D:**

    Import application-defined objects.

  - **G:**

    Generate reports and screens.

  - **J:**

    Activate all ABAP Dictionary objects other than the enqueue modules in the change request with mass activation (new activation program).

  - **M:**

    Activate enqueue modules in the change request with mass activation.

  - **N:**

    Convert all structure changes generated by the import and registered in the table TBATG, other than those of matchcode objects.

  - **O:**

    Convert all structure changes that were generated by actions in the online system and **not** by the import, and that are registered in the table TBATG.

  - **R:**

Table TRBAT

Execute programs after PUT (XPRA).

- **S:**

  Necessary actions to transfer the new Dictionary structures into the runtime environment are distributed to different steps.

- **X:**

  Export application-defined objects.

- **Y:**

  Convert structure changes to matchcode objects that were generated by the import and are registered in table TBATG.

- RETCODE

  Contains return codes for finished steps

  Possible values are:

  - **9999: 9999:** This step is waiting for execution.
  - **8888: 8888:** This step is active.
  - <= **12**: This step is finished.

- TIMESTMP

  Contains the order in which the actions should be executed Contains the completion time for finished steps

In addition, a header entry is inserted to tell RDDIMPDP to start (the RETCODE column is set to B(egin)).

TRKORR	FUNCTION	RETCODE	TIMESTMP
T11K004711	J	9999	00000000000001
T11K004720	J	9999	00000000000002
T11K003456	J	9999	00000000000003
HEADER	J	B	19921207130147

If RDDIMPDP is activated, it sets the header entry to R(unning) and starts the corresponding program (mass activation in this case), which sets the first transport to the active status.

TRBAT now looks like this:

TRKORR	FUNCTION	RETCODE	TIMESTMP
T11K004711	J	8888	00000000000001
T11K004720	J	9999	00000000000002
T11K003456	J	9999	00000000000003
HEADER	J	R	19921207130650

8888 in the RETCODE column means that this change request is active. After it is finished, the column contains the return code of the step, and the TIMESTMP column contains the completion time.

When all actions are performed, the header entry is set to F(inished). In the meantime, `tp` copies the logs of finished steps from the `tp` directory to the log directory and deletes the corresponding TRBAT entries.

TRBAT then looks like this (the log of the first change request was already copied to the log directory):

TRKORR	FUNCTION	RETCODE	TIMESTMP
T11K004720	J	4	19921207130912
T11K003456	J	0	19921207130945
HEADER	J	F	19921207139945

As soon as the last log has been edited, `tp` deletes the header entry and TRBAT is empty again.

Note that actions **B** (TACOB activation), **N**, **O**, **Y** (structure conversion), and **S** (distribution) only have header entries. They work independently of change requests.

Table TRJOB

As soon as RDDIMPDP recognizes that there is work to do, the job number (generated by SAP background processing) is inserted into the table TRJOB before the corresponding actions are started.

If RDDIMPDP has become active again and has found a running job in TRBAT and an entry in TRJOB, it checks if the job is still running.

If not, the corresponding step (`retcode = 8888`) is set to 12, and the next step is executed. As soon as the header entry in TRBAT is set to F(inished), the TRJOB entry is deleted.

The R3trans Program

Use

The SAP transport program `R3trans` is used to transport data between SAP Systems and for the migration between different SAP releases.

However, note that `R3trans` is usually called from other programs, in particular from `tp` (Transport Control Program tp [Page 7]) and `R3up` (upgrade control program).

Therefore, direct use of `R3trans` is only necessary in exceptions and is not supported other than in these cases. This `R3trans` documentation does not describe the normal transport procedure, but offers help in cases where the regular tools and methods are insufficient.

Versions of R3trans

Use the version of `R3trans` which corresponds to the version of your SAP System. However, different versions of `R3trans` are fully compatible with each other, because the data is transported in a standardized format. You can therefore use different versions of `R3trans` for export and import.

Transports between different databases or operating systems are not a problem either.

When you transport between different SAP releases, this usually does not present any technical problems due to the upward and downward compatibility of `R3trans`. However, these transports can lead to logical inconsistencies.

Calling R3trans

Call `R3trans` as follows:

`R3trans [<options>][<control file>]`

AS/400: **`R3trans '[<options>][<control file>]'`**

The actions to be performed are specified in the control file.

Options for Calling R3trans

You can specify the following options for calling `R3trans`:

- **`-c <f1> [<f2>]`** Conversion

  Copies file `<f1>` to file `<f2>` by converting the character set to the local character set. If `<f2>` is omitted, `R3trans` copies to `trans.dat` (see Automatic Character Set Conversion with R3trans [Page 103]).

- **`-d`** DB connect

  Check if connect to database is possible and if the ABAP Dictionary is available.

- **`-i <file>`** direct import from data file without specifying a control file (see Import [Page 99])

- **`-l <file>`** Output of a table of contents to the log file

- **`-n`** Brief information about new features of `R3trans` not yet documented

- **`-t`** Test mode All modifications in the database are rolled back.

- **`-u <int>`** Unconditional mode This option invalidates the rules of the correction and transport system. Each mode can be concatenated (for example **`-u 12`**).

- **`-v`** Verbose mode

  Writes dots to `<stdout>` and writes additional details to the log file.

- **`-w <file>`**

  The transport log is written to the specified file. The default is `trans.log` in the current directory.

- **`-x`** Check if connection to database can be made

R3trans Return Codes

`R3trans` sets a return code that shows whether or not the transport has succeeded. You can view details on transports in the log file. Here are the following return codes:

- `0`: No errors or problems have occurred.
- `4`*:* Warnings have occurred but they can be ignored.
- `8`*:* Transport could not be finished completely. Problems occurred with certain objects.
- `12`: Fatal errors have occurred, such as errors while reading or writing a file or unexpected errors within the database interface, in particular database problems.
- `16`*:* Situations have occurred that should not have.

  Examples:

  – Constants in the program are too small

  – Not enough available memory

  – Internal `R3trans` error

Other return codes are not set by `R3trans` itself but point to errors, such as segmentation faults.

Unconditional Modes

By using unconditional modes, you can override the rules of the transport system. The following modes are available:

Export:

- **1**: Ignore incorrect status of a change request.

  Normally only change requests with the status **A** (all locked), **O** (open) or **R** (released) are exported. If the export is successful, the change request is released by `R3trans`.

- **8**: Allow direct selection of forbidden tables.

  Some tables are part of complex objects and generally cannot be exported alone.

Import:

- **1**: Ignore that this change request was already imported into this system and import everything again. Only objects which have not already been imported successfully are imported again.
- **2**: Overwrite originals
- **6**: Overwrite repaired objects
- **8**: Ignore the restriction resulting from table classifications and import all table entries into specified clients.

Export

Exports are usually started within the SAP System (Transport Organizer) or from `tp` (transport control program), where `R3trans` is called indirectly.

The control file for an export has the following structure:

```
export
# < options >
# < selections >
```

This entry shows a single step. A control file can consist of several steps.

Export Options

You can use the following parameters in control files:

- **CLIENT = DEFAULT | ALL | <client>**

  The default is 0. The export is performed from client 0 by default.

  ➡

  Cross-client data is also exported.

  The option allows you to perform the export from clients other than 000 or from all clients.

- **COMPRESS = YES | NO | C | D | P**

  The default is **C**.

  You can export compressed data. The data is automatically unpacked at import. There are the following options:

  - **D**: Simple compression, where successive bytes with the same value are grouped together
  - **P**: Compression of common prefixes of successive data entries
  - **C**: Combination of **D** and **P** (default)

- **FILE = <file name>**

  The default is `trans.dat` in the current working directory. The data is exported to this file.

- **LANGUAGE = <languages>**

  Objects belonging to the class TEXT or entries from language-dependent tables are only exported if they exist in one of the languages specified here.

  ➡

  With this parameter, not the differences in upper and lowercase spelling. Lowercase letters in the language vector are **not** converted automatically into uppercase letters.

  See also: Transporting Language-Dependent Data [Page 105]

- **RECCLIENT=XXX**

You can activate table logging with this `R3trans` option.

XXX can have the following values:

- nnn: Logging is only performed in client *nnn* for client-specific tables.
- mmm,nnn,ppp,...: Logging occurs for the clients listed here for cross-client tables.
- ALL: Logging always occurs for all clients for client-specific tables.
- OFF: No logging (default)

➡

Only the tables marked appropriately in the ABAP Dictionary are logged (particularly Customizing tables).

For further information, particularly on evaluating these logs, see Note 1916.

- **SQLTRACE = YES | NO**

  The default is NO.

  With this option, you activate a trace for an `R3trans` run for a database statement. You can use this trace to determine the causes of performance problems during transports.

  This database trace is only implemented for Oracle (`tkprof`).

Selections for Export

There are various methods for specifying which data is to be exported. Normally this is done using change requests (command files). If this is not possible for any reason, you can also specify selections directly.

Parameters for Control Files

- **USE COMMANDFILE<change request>: <change request>**

  Change request that was maintained by the Transport Organizer

  Several entries of this type may also be contained in a single step.

- **USE DEVCLASS <devclass>**

  `R3trans` creates an object list `DEVC<devclass>` with all objects belonging to this development class and exports these objects.

  Several entries of this type may also be contained in a single step. This type of selection is especially suited for transporting complete development classes.

Direct Selection

For special purposes (such as repairs), you can export without command files by specifying direct selections of objects in a control file:

- **SELECT <object> [<whereclause>]**

  You can specify the following objects directly: `CMDFILE`, `CUA`, `DATAELEMENT`, `DOCUMENTATION`, `DOMAIN`, `SCREEN`, `REPORT`, `SQLTABLE`, `TABLE`

  You have the following options in `<whereclause>`:

  - `WHERE NAME = <name>`, where `<name>` can also end with “*” (for generic names)
  - `WHERE NAME BETWEEN <name1> AND <name2>` for selecting an interval

- **SELECT * FROM <tabname> WITH KEY <generic key>**

  This command lets you select records from a table with a generic key.

  ```
  SELECT * FROM THLPF WITH KEY SAPMS01*
  ```

Import

Imports are generally started by `tp` (transport control program), where `R3trans` is called indirectly. The control file for an import has the following structure:

```
import
# < options >
# < selections >
```

If you use the default options for the import, you do not need a control file. You can perform the import directly with the following command:

```
R3trans -i <file>
```

Import Options

By using the following parameters, you can specify in the control file how an import should be performed:

- **BUFFERSYNC = YES | NO**: The default is NO.

  This parameter controls whether `R3trans` synchronizes the buffer for the imported data. This synchronization only affects the data actually imported. Synchronization is for single objects (individual tables, reports, etc.). No complete buffer classes are synchronized.

- **CHARSETADAPTION/CHARSET**

  For all language-specific objects, `R3trans` automatically selects the appropriate conversion table for the language concerned. If you, for example, want to transport a Polish or Czech text, a conversion table according to LATIN-2 is used.

  To deactivate the automatic character set conversion and to convert everything according to LATIN-1, use the following `R3trans` option:

  ```
  charsetadaption=no
  ```

  Some texts cannot be converted correctly because they are not stored in language-dependent objects (for example, comments in ABAP programs). If you also want to convert these types of texts correctly, you can specify explicitly which character set is to be used for conversion:

  ```
  charset = XXX
  ```

  Replace XXX by one of the following values:

  ```
  WEUROPEAN EUROPEAN CYRILLIC TURKISH GREEK HEBREW
  ```

  You can also make these settings in the transport profile.

  ```
  <SID>/r3transoptions =
  charsetadaption=no;charset=eeuropean
  ```

- **CLIENT = DEFAULT | ALL | <client>:** The default is ALL.

  By default, the import always takes place in the same client as the export. This option allows you to import into other clients. See R3trans and Clients [Page 108].

- **SAFEANDSLOW = YES | NO**: The default is YES.

This parameter enables you to switch off the R3TRANS "Arrayinsert". This enables R3trans to better manage data inconsistencies. The disadvantage is that R3trans works much more slowly. Use this parameter only if it is specifically recommended by SAP for resolving a particular problem.

- **SELECTIVE_LANGUAGE IMPORT = YES | NO**: The default is NO.

  See also: Transporting Language-Dependent Data [Page 105]

- **SQLTRACE = YES | NO**: The default is NO.

  This parameter enables you to activate a trace for a database statement for an R3trans run. You can use this trace to determine the causes of performance problems during transports. This database trace is currently only implemented for Oracle (tkprof).

- **SUBSTITUTE <hex1> <hex2>**

  This parameter activates the special character set conversion for single characters, if the default conversion is not sufficient (for example, if problems occur with umlauts, see Automatic Character Set Conversion with R3trans [Page 103]).

- **TABLECACHESIZE**

  R3trans automatically adapts its main memory requirements during import to the main memory available. This means the main memory requirements are independent of the volume of tables to be transported.

  You can limit the size of main memory area to be used for merging the tables by using the option tablecachesize.

  Default values for this parameter are:

  - 16000000 for AIX and OSF1 (DEC-UNIX) due to the undefined behavior for the internally used function realloc(). If there is enough main memory, you can increase the value.
  - 16000000 for OS400. Do not increase this value.
  - -1 (= infinite) for all other platforms.

Options for Selective Imports

You can perform a selective import of certain data in an import file. SAP recommends importing complete files. Selective imports are only useful for restoring backups.

The data file is imported completely by default. You can import specific parts of data by specifying include or exclude lists in a control file.

- EXCLUDING <string1>, <string2>, ...:

  With this specification, you can exclude certain objects (reverse logic). During the upgrade, for example, you can exclude table entries the first time in order to import them after the activation.

- INCLUDING <string1>, <string2>, ...:

  Only data whose object directory entry (TADIR) starts with <string> is imported. Typical strings are R3TRTABL, R3TRPROG, etc.

The control file for a selective import appears as follows:

```
import
file = ...
excluding “R3TRCNTD”, “R3TRTABU”
```

Test Import

Before an import, you can check if it will succeed or not. The test import checks the rules of the Transport Organizer and transport system.

This option lets you identify problems before you perform the actual import and can solve problems in advance.

The control file for a test import has the following structure:

```
testimport
file = ...
```

File Contents

If you want to create a table of contents for a file, use the following control file or the `R3trans` command:

-
  ```
  listonly
  file = ...
  ```
- `R3trans -l <file>`

Automatic Conversion with R3trans

If the structure of a table in the target system is different from the structure in the source system, `R3trans` performs a conversion which is similar to the `MOVE-CORRESPONDING` in ABAP:

- Character fields with different lengths are shortened or lengthened.
- Missing fields are not converted.
- Initial values are entered in new fields.
- Packed and binary fields with different lengths are shortened or lengthened.
- Conversion from `NUMC` to `CHAR`.
- Conversion from `HEX` to `SHORT/LONG`.

Automatic Character Set Conversion with R3trans

The character set conversion is automatically converted for transports between SAP Systems with different code pages.

You can also use this `R3trans` function separately. For example, this is a good supplement to your own migration and conversion programs because they are independent of different character sets in the different R/3 Systems.

The only prerequisite is that the file you create corresponds to the R/3 transport format. You can then convert the file to another character set with `R3trans` and continue processing the file with your own program. As a side effect, packed fields are converted into a valid R/3 presentation.

Use the following control file for this task:

```
convert
input file = ...
output file = ...
```

If you omit one or both of the file specifications, `trans.dat` is used by default at the relevant point. If your operating system allows reading and writing of the same file, input and output files can be identical. This is the case in UNIX and Windows NT systems.

The call `R3trans -c <file1> <file2>` does the same job.

The character set conversion is currently implemented so that data is always converted into the character set that is valid on the operating system in which `R3trans` is running. The conversion must always run on the target system.

For programs for migrating application data from R/2 to R/3 Systems, the character set conversion has been extended with an exit for special characters.

This extension was required because characters used in R/2 were not always converted into the correct code page (for example, for Danish characters) and this resulted in incorrect character representations in R/3.

You want to change the character Ö (hex d6) to ∅ (hex d8) by converting into ASCII ISO 8859-1. Use the following control file:

```
convert
input file = ...
output file = ...
substitute 0xd6 0xd8
```

The hexcodes must always refer to the **character set**.

You can also use the option `SUBSTITUTE` for normal imports.

Transporting Language-Dependent Data

During the import and export , you can select language-dependent data.

Export: Selecting language-dependent data

During the export, the data is always searched in all languages.

If you want to specify a behavior that deviates from this, you need to set both `R3trans` options `language` and `language_selection_mode` (`lsm`). In the transport profile of the transport control program `tp`, you can specify these `R3trans` options using the parameters `language` and `r3transoptions` (up to 3.0F) or `lsm` (as of 3.1G).

Option `language`

The default value for this option is "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz".

As long as this value is not changed, language-dependent data is exported in all languages. By specifying only a few languages, you can restrict the languages to be selected. This reduces the export times for tables where the key begins with the language key. A disadvantage is that you have to adjust the parameter `language` in the transport profile if you install languages in the system.

Option `language_selection_mode` (`lsm`)

The default value for this option or `lsm` is `VECTOR` (the value of the option `language` is often described as a language vector). This means that `R3trans` is selected in the languages specified with the option `language`. For specific object types, you can specify a different behavior according to different object types:

– Development environment objects, such as ABAP programs, screens, GUI statuses, Data Dictionary objects or T100 messages, but **no** long documentation:

  – `lsm = MASTER`

    As of Release 3.0B, the **master language** is stored for each object in the object directory (table TADIR). This is the language in which the object was created or is maintained. In this mode, the language-specific data in the master language is transported, however, not translated texts. This master language does not have to be contained in the language vector.

  – `lsm = NOMASTER`

    As for `TRANSLATION` however, languages not contained in the language vector are also considered.

  – `lsm = TRANSLATION`

    Only the translated data is transported in this mode, not however entries in the master language. Only languages contained in the language vector are considered.

  – `lsm = VECTORANDMASTER`

    All entries belonging to the master language or contained in the language vector are transported.

- Long documentation

The long documentation for the object types above is handled in the same way as the objects, and you can also control it using the options `language` and `lsm`.

⚠

For the long documentation, the master language is not read from the object directory (TADIR). Instead it is directly with the object. This mechanism enables a different master language for the long documentation from the object to which it belongs.

This affects `lsm = MASTER`, `TRANSLATION`, `NOMASTER`, or `VECTORANDMASTER`. There is no effect on `lsm = VECTOR` (default).

- Table entries

  There is no entry in the object directory and no master language for general table entries in a transport request (R3TR TABU …). `R3trans` always behaves according to `lsm=VECTOR` independent of the value set for option `lsm`. This means all the languages specified in the language vector are searched.

  This does not apply if a table key is specified in the transport request in which the language field is explicitly evaluated. In this case, the specified entry is selected even if the language specified there is not in the language vector.

  These types of selections are a particular result of recording Customizing changes automatically in transport requests. This often involves entering the logon language directly in the selection. You can use **LSM=ALL** to overwrite the specified language with the language vector.

  If a table has the language field at the beginning of the key and you want to transport in all languages, but you want to specify individual keys or key areas, you can enter ***** in the language field. (This applies only for the language field in the key).

Up to Release 3.0F, option `lsm` is set in the transport profile using the parameter `r3transoptions`, for example, `r3transoptions = lsm=MASTER`. As of Release 3.1G, there is a unique parameter `lsm`, for example `lsm=MASTER`.

Import: Selecting language-dependent data

You can restrict the data set during the import using the SAP transport tools `tp` and `R3trans` so that language-dependent data is imported in the languages that also exist in the system. This is useful, for example, if an export data set is created for several different installations. Since these imports were always completely imported into the target systems, there were language-specific table entries in non-installed languages.

Option `selectivelanguageimport (sli)`

Since `R3trans` Release 4.5A (usable as of SAP Release 4.5A), you can use the option `selectivelanguageimport` (`sli`) to control the selective language import. The default value for this option is `sli=no`. If `sli=yes`, you ensure that only the language-dependent data is imported whose languages exist in the system. For this purpose, `R3trans` analyzes a log table of the language import.

⚠

In this case, it makes sense if the number of languages selected (language vector at export) contain all the languages of the target systems during the import. If this is not the case, `R3trans` generates a corresponding warning at import.

➡

The languages in the system are the ones for which language imports were already performed. Languages that were **not imported** are not part of the system and are not overwritten.

This option is used as of Release 4.5B when importing Support Packages (Transaction SPAM).

R3trans and Clients

Both the R/2 and R/3 Systems use the client concept to allow legally and financially independent entities to coexist in a system. `R3trans` provides special functions for working with clients. Note that client names are always numeric.

Transporting Client-Specific Tables

`R3trans` exports from client 0 by default. However, you can also export from a different client (for example, at the customers) or from all clients (for example, for table conversions).

By default `R3trans` imports into the client that the export came from. During the import, you can specify other target clients.

If the transport requests are released regularly, exports always occur automatically from the current client.

Creating a Client

The R/3 System provides a wide range of tools for creating clients. Copy tables directly using `R3trans` **only in exceptional situations**. Otherwise this is function is not supported.

Creating clients only with `R3trans` requires a lot of subsequent manual processing. In this case, a fee is charged if support is provided by a consultant.

The `R3trans` function CLIENTCOPY enables you to create a new client from an existing one. This is also possible for single tables.

The control file appears as follows:

```
clientcopy
source client = 0
target client = 23
use commandfile xyz /* all tables from change request */
# or
select * from x /* single tables*/
select * from y  ...
```

Deleting a Client

With the `R3trans` function CLIENTREMOVE, you can remove these table entries again.

The control file appears as follows:

```
clientremove
client = 23
use commandfile xyz /* all tables from change request */
# or
select * from x /* single tables*/
select * from y   ...
```

Recommendations for R3trans

The following recommendations help you if you must use `R3trans` directly.

Export

Use **transport requests**. They have significant advantages for the import.

Import

Exporting objects using a transport request gives you the following advantages:

- `R3trans` restarts automatically if the first import aborts or if single objects have not been imported (provided that you did **not** use unconditional mode 1 for a complete reimport).
- The transport request is also used by the mass activation program, which you **must always** call if you also transport ABAP Dictionary objects. The mass activation program establishes whether or not a certain object has been imported successfully by checking the transport request.
- The transport is also visible in the SAP System.

Logs

Note that the log file is always called `trans.log`, unless you specify otherwise. To redirect logging, use the option **-w** so that you can keep track of all errors that have occurred.

Transporting Between Different SAP Releases

`R3trans` is **upward compatible**. `R3trans` writes data using a standard SAP transport format. This enables you to export data with an old `R3trans` version and import data with a new version of `R3trans`. If you have any doubts on which version to use, contact the SAP hotline.

Transports between different databases or operating systems are not a problem either. When you transport between different SAP releases, this usually does not present any technical problems due to the upward and downward compatibility of `R3trans`, but logical inconsistencies can occur in such transports.

Unconditional Modes

Use unconditional modes very carefully. Transport according to the rules of the Transport Organizer and Transport Management System.

⚠

Using `R3trans` unconditional modes, it is easy to destroy work that has taken days or even weeks to perform.

Problems

If you have any problems with `R3trans`, refer to this documentation to check if the function you have used is exactly what you need. All known errors are stored in the SAP information system (Notes). In addition, you can contact the SAP hotline.

Transport Control under Windows NT

Use

This chapter describes how you can control transport between UNIX and Windows NT systems.

Contents

Transporting Between UNIX and Windows NT Systems

When you use the Change and Transport System (CTS), various files described by the kernel, `tp.exe`, and `R3trans.exe` are created in the transport directory. The files of the subdirectories `log`, `buffer`, and `cofiles` are opened in default mode (=text mode), while the files of the transport subdirectory `data` are described in binary mode.

When the files are written in text mode, there are the following differences between the two operating systems:

- UNIX writes a line feed at the end of each line.
- Windows NT writes a carriage return/line feed at the end of each line.

When the files are written in binary mode, a line feed is written under UNIX and under NT at the end of each line. The files that are written in text mode under Windows NT cannot therefore be read under UNIX, and vice versa.

To set up a **common transport directory for Windows NT and UNIX**, you must choose the binary mode as the default mode to describe the files. You can set this using profile parameters or environment variables. These parameters are not required for transports between pure Windows NT systems performed in the conventional text mode.

Adjusting Modes to Open a File

To use a common transport directory for both operating systems, you must adjust the mode to open a file under Windows NT in the UNIX text mode.

- For the transport tools, you can set this using the transport profile parameter `ABAPNTFMODE` or the environment variable `abap/NTfmode`.
- For the SAP Kernel, set the mode to open a file using the profile parameter `abap/NTfmode`.

Required Software and Tools

UNIX

At present, Windows NT does not have a hierarchical file system like UNIX. There is also no way to create soft links. As a result, the file systems on NT machines cannot be included in the local directory tree, or vice versa.

To enable both systems to access the file systems, you need to install additional software.

This software must meet the following requirements:

- Distinguish between uppercase and lowercase within file names, since file names are case-sensitive under UNIX
- Authentication on the respective system with the LanManager API
- Display contents of directories, independent of how these directories are accessed or referenced

➡

Only use software that can export soft links.

Since the LanManager has a large share of the PC network market, software was developed for UNIX systems to enable them to access LanManager networks. There is now appropriate server software for almost all UNIX dialects. This software allows the UNIX system to appear as the LanManager to the PC.

Many hardware manufacturers offer this type of LanManager software. However, SAP cannot recommend which software you should use. When deciding on a product, make sure that the three requirements listed above are met.

SAP uses the LanManager server SAMBA from Andrew Tridgell, which you can obtain free of charge from the Internet.

`http://samba.anu.edu.au/samba`

➡

Note that SAP does not support SAMBA.

The LanManager SAMBA provides the following functions:

- File services are made available. You can access NT files in exactly the same way as from a UNIX shell.
- Uppercase and lowercase are distinguished within file names.
- You can configure the SAMBA server so that it relies on the authentication of a Windows NT server at logon.

  The assignment of the UNIX user to the corresponding NT user is stored in a SAMBA server table. The user is checked from NT.

  For more information, see Example of a SAMBA Configuration File [Page 120].

Configuring with TMS Under Windows NT

Procedure

To configure the TMS under Windows NT, proceed as follows:

1. To make your Windows NT system known in the transport profile:

   a. Log on to the transport domain controller and call Transaction STMS. The initial screen of the TMS appears.

   b. Choose *Overview* → *Systems*. The system overview appears. Choose a system by double-clicking one.

   c. Choose Configuration → Display/Change. Click on the tab *Transport tool*, and choose *Edit* → *Insert row*.

   d. Enter the following information:

Global	Parameter	Value	OpSys
x	ABAPNTFMODE	b	wnt
x	transdir	<path to UNIX TransDir>	wnt

Global	Parameter	Value	OpSys
x	ABAPNTFMODE	b	wnt
x	transdir	\\trans01\trans4	wnt
	A99/r3transpath	<path>\R3trans.exe	wnt
	A99/sapevtpath	<path>\sapevt.exe	wnt
	A99/system_pf	<path>\default.pfl	wnt

   The parameter `ABAPNTFMODE` determines the mode for opening the files used by `tp` and all the tools. These tools are called by `tp`.

   e. Save the configuration by choosing *Configuration* → *Save* and then go back to the system overview.

   f. Choose *Extras* → *Distribute TMS Configuration*.

2. Include the following entries in the system-specific instance profiles of the SAP Systems under Windows NT:

   a. Set the parameter `abap/NTfmode=b`.

      - This parameter sets the default mode for opening a file for the kernel.

   b. Set the parameter `DIR_TRANS` on the UNIX transport directory.

```
DIR_TRANS=\\trans01\trans4
```

c. Set the parameter `DIR_EPS_ROOT` to a subdirectory of the transport directory for Transaction SPAM.

```
DIR_EPS_ROOT=\\trans01\trans4\EPS
```

d. Stop the relevant SAP System under Windows NT and then start it again.

3. You must convert the profiles into binary format of all systems running under windows NT that are supposed to use the central transport directory.

   To adjust the Windows NT profile to the binary format, log onto the SAP System and call Transaction RZ10. Choose *Utilities* → *Import profiles* → *Of active servers.* This saves the profiles in UNIX format.

Result

The TMS is now configured for transport between UNIX and Windows NT systems.

Configuring Without TMS Under Windows NT

Prerequisites

To access a common transport directory, you need the editor SAPPAD.EXE. The editor is in the executable directory of your system, and it can also interpret UNIX and Windows NT file formats.

Procedure

1. Set the binary mode for the SAP transport tools.

   As of Release 4.5A:

   Enter the following parameter in your transport profile:

   ```
   ABAPNTFMODE=b
   ```

2. Enter the following parameter in the transport profile:

   ```
   wnt|transdir=<path to UNIX transport directory>
   ```

   ```
   wnt|transdir = \\trans01\trans4\
   wnt|A99/r3transpath=\\<path to R3trans>\R3trans.exe
   wnt|A99/sapevtpath=\\<path to sapevt>\sapevt.exe
   wnt|A99/system_pf=\\<NTHOST>\sapmnt\<SID\sys\profile\default.pfl
   ```

   You must save the transport profile using SAPPAD in the UNIX format. To do this, choose *Options* → *Save as UNIX File*

3. Change the instance profile of the SAP systems under Windows NT.

   You must make the following entries in the system-specific instance profile for each system under Windows NT that is supposed to use the central transport directory:

   a. Set the parameter **`abap/NTfmode=b`**.

      This parameter sets the default mode for opening a file for the kernel.

   b. Set the parameter `DIR_TRANS` on the UNIX transport directory.

      ```
      DIR_TRANS=\\trans01\trans4
      ```

   c. Set the parameter `DIR_EPS_ROOT` to a subdirectory of the transport directory for Transaction *SPAM*.

      ```
      DIR_EPS_ROOT=\\trans01\trans4\EPS
      ```

   d. Stop the relevant SAP system and the service `SAP <SAPSID>_<instance no.>` by choosing *Control Panel* → *Services*.

   e. Then restart the SAP system and the service.

4. Adjust the SAP profile to the UNIX format.

   You must convert the profiles into binary format of all systems running under windows NT that are supposed to use the central transport directory.

   To do this, log on to each SAP System and call Transaction **RZ10**. Choose *Utilities* → *Import profiles* → *Of active servers.*

   This saves the profiles in UNIX format.

Example of a SAMBA Configuration File

The following is an example of a SAMBA Configuration File with NT Authentication.

```
----------> NT Server Authentication
;Configuration example file for samba using computer pswdf004 as
password server
[global]
; SECURITY OPTION guest account: the guest has the file access
; rights of this UNIX-user
guest account = nobody
debug level = 0
security = server
password server = pswdf004
getwd cache = yes
wide links = no
password level = 8
case sig names = yes
preserve case = yes
case sensitive = yes
read prediction = yes
[transdir]
comment = us0011:/usr/sap/trans
path = /usr/sap/trans
; SECURITY OPTION public: all users can access the share as a
; guest, no password is checked
public = yes
; SECURITY OPTION writable: files + directories can
; be changed if writable = yes
writable = yes
; SECURITY OPTION browsable: Share is visible if browsable = yes
browsable = yes
create mask = 0664
map hidden = no
map system = no
preserve case = yes
wide links = yes
; SECURITY OPTION allow hosts: only specified hosts can access
the
; share
allow hosts = us0011
```

Example of a Heterogeneous Transport Profile

```
ABAPNTFMODE  = b
transdir     = /usr/sap/trans/
wnt|transdir = \\trans01\trans4\
syslog       = SLOG$(syear)$(yweek).$(system)
alllog       = ALOG$(syear)$(yweek).$(system)
r3transstat  = $(transdir)log/STATLOG.$(system).#.$(yweek)

BIN/dbhost   = hs0055
BIN/dbname   = BIN

BIN/r3transpath     = /bas/$(system)/exe/dbg/$(cpu)/R3trans
wnt|BIN/r3transpath = \\NTPC\sapmnt\bin\sys\exe\run\R3trans.exe

sapevtpath         = /bas/$(system)/exe/dbg/$(cpu)/sapevt
wnt|BIN/sapevtpath = \\NTPC\sapmnt\bin\sys\exe\run\sapevt.exe

BIN/system_pf     = /bas/$(system)/profile/DEFAULT.PFL
wnt|BIN/system_pf = \\NTPC\sapmnt\bin\sys\profile\default.pfl
BIN/impdp_by_event = yes
```

Specifying the Central Transport Host

Use

You have set up a common transport directory for (almost) all the SAP Systems in which you want to centrally manage your transports. Furthermore, you can simplify the maintenance of the transport directory by maintaining the directory at a central point.

The advantage of a central transport host is that you only have to maintain one central entry when moving the transport directory to another host, and not the parameters of each system.

To do this, specify the name of the central transport host, where your central transport directory is located, in the **domain name server (DNS)** for all Windows NT systems.

> If you did not specify the `ALIAS SAPTRANSHOST`, the default value is valid for `DIR_TRANS` (your central instance):
>
> ```
> DIR_TRANS = \\$(SAPGLOBALHOST)\sapmnt\trans
> ```

Procedure

1. Adjusting the general configuration:
   a. Enter the `ALIAS SAPTRANSHOST` on the domain name server for the transport host you have chosen.
   b. Maintain the transport directory in the transport profile.

      The transport control program `tp` works with its own profile and cannot access the default SAP profiles. The parameter `transdir`, which is located in this profile, contains the path of the transport directory. You must set this parameter. For a central transport directory, define the parameter using the macro `$(SAPTRANSHOST)`.

      ```
      transdir = \\$(SAPTRANSHOST)\sapmnt\trans\
      ```

2. Adjusting the SAP-specific configuration:
   a. Give each transport user that wants to access the central transport directory the write-authorization for the transport directory and its subdirectories.
   b. Parameter `DIR_TRANS`:

      The parameter contains the path of the transport directory and is maintained in the instance-specific profile (if necessary in the default `default.pfl`). This parameter is evaluated by all programs that can evaluate profiles. These are the SAP Kernel and various additional programs, such as `sapntchk`, `sappfpar`, etc.

      Normally you should not set this parameter explicitly, since the default value, which points to `\\$(SAPTRANSHOST)\sapmnt\trans\`, usually is correct.

If you do not want to store the path under the share `\sapmnt\trans` within the transport directory, you can convert parameter `DIR_TRANS` using the macro `$(SAPTRANSHOST)`.

```
DIR_TRANS = \\$(SAPTRANSHOST)\transport
```

If you do not set the parameter `DIR_TRANS` in any profiles, the default value is activated.

If a system for which a central transport host is configured is **not** supposed to transport a central transport directory, then override the default values [Page 124].

Transporting to a Private Transport Directory

Use

If a system for which a central transport host is configured is **not** supposed to transport a central transport directory, then override the default values for this system.

Procedure

- Configuring a private transport directory for **all** systems of a host:

  If all the systems of a host access a common transport directory, you can override the transport host specified in the domain name server locally.

  Enter the following in the local file `%WINDIR%\system32\drivers\etc\hosts`:

<IP address of private> <transport host>	<TCP/IP name of private> <transport host>	SAPTRANSHOST

  155.56.1.74 trans01 SAPTRANSHOST

  End the file `hosts` with a **blank line**.

- Configuring a private transport directory for **one** system of a host:

  Set the parameters `DIR_TRANS` and `transdir` to the new transport directory for the single system.