ToolTalk Commands 5

This chapter provides reference information for the ToolTalk-enhanced operating system commands.

ttce2xdr

ttce2xdr [-xn] -d *user* | *system*
ttce2xdr [-xn] -d *network* [*OPENWINHOME-from* [*OPENWINHOME-to*]]
ttce2xdr [-h]
ttce2xdr [-v]

Converts the ToolTalk Classing Engine type tables to XDR format.

Description

The `ttce2xdr` command converts the ToolTalk types stored in the Classing Engine database used by ToolTalk Versions 1.0.x to the XDR-format database used in ToolTalk Version 1.1 and later. This command generally only needs to be used to update types databases common to systems or the network; it is run automatically for user type databases.

The format to convert user or system data bases for the current user or current system is:

```
ttce2xdr [ -xn ] -d user| system
```

The format to convert the network-wide data base is:

```
ttce2xdr [ -xn ] -d network [ OPENWINHOME-from [ OPENWINHOME-to ]]
```

This format provides additional options allowing types to be taken from one database and stored into another.

Options

-d

Specifies the database (use, system, or network) to be converted. The types are read from the Classing Engine database (shown int Table 5-1) and written to the XDR database (shown in Table 5-2).

If the network database is specified, the optional arguments `OPENWINHOME-from` and `OPENWINHOME-to` may be specified.

- If neither argument is specified, ToolTalk uses the current value of the environment variable `OPENWINHOME` to locate the data bases to be read from and written to.
- If only the `OPENWINHOME-from` argument is specified, ToolTalk reads from and writes to the databases under the directory named by `OPENWINHOME-from`.
- If both arguments are specified, ToolTalk reads from the database under `OPENWINHOME-from` and writes to the database under `OPENWINHOME-to`.

Table 5-1 Classing Engine Database

Database	Directory
user	~/.cetables/cetables
system	/etc/cetables/cetables
network	$OPENWINHOME/lib/cetables/cetables

Table 5-2 XDR Database

Name	Database
user	~/.tt/types.xdr
system	/etc/tt/types.xdr
network	$OPENWINHOME/etc/tt/types.xdr

-h
Describes the ttce2xdr options and exits.

-n
Displays the underlying commands that are to be executed by the `ttce2xdr` command.

-v
Prints version and exits.

-x
Displays the underlying commands executed by the `ttce2xdr` command.

Environment

CEPATH
In Classing Engine mode, `tt_type_comp` uses this variable for its definition of where the databases are located.

OPENWINHOME
Location of network databases.

Files

~/.tt/types.xdr
The user's ToolTalk XDR format types file.

/etc/tt/types.xdr
The system ToolTalk XDR format types file.

$OPENWINHOME/etc/tt/types.xdr
The network-wide ToolTalk XDR format types file.

$OPENWINHOME/lib/cetables/cetables
The Classing Engine database that contains the ToolTalk type definitions.

ttcp

ttcp [–pL] *filename1 filename2*
ttcp -rR [–pL] *directory1 directory2*
ttcp [–prRL] *filename directory*
ttcp -v
ttcp -h

Copies files and directories in a ToolTalk-safe way.

Description

The `ttcp` command invokes the standard operating system command `cp` to copy files and directories, and informs the ToolTalk service about its actions. This command ensures that the ToolTalk objects associated with the named files and directories are copied at the same time.

Options

-h
Prints the usage information for the `ttcp` command.

-i
Interactive copy option.

Note – This option of the standard operating command `cp` is not supported.

-L
Copies the ToolTalk objects of the specified files but does not invoke the standard operating system command `cp` to copy the actual files.

–p
Preserves the modification time and permission modes when the contents of the original file or directory are copied.

Note – To preserve the modification time of ToolTalk objects, you must copy the file or directory as *superuser*.

–r
–R
Recursively copies any files and directories (including any subdirectories and their files) associated with the specified source file.

Note – The destination must be a directory.

-v
Prints the version of the `ttcp` command.

Files

***/mountpoint*/TT_DB**
The directory used as a database for the ToolTalk objects of files in the filesystem mounted at */mountpoint*.

ttdbck

ttdbck [-v] [*selection*<*plainopts*] [*diagnosis*<*plainopts*] [*display*<*plainopts*] [*repair*<*plainopts*] [*data-base-directory*]...

Display, check, or repair ToolTalk databases.

Description

The `ttdbck` command is the ToolTalk database maintenance tool. Use this command to directly inspect ToolTalk spec data, detect inconsistencies, and repair the problems found. Run this command on the same machine on which the ToolTalk database files that are being inspected and repaired physically exist; that is, do not attempt to access the ToolTalk database files using the Network File System (NFS).

Note – Since ToolTalk databases are typically accessible only to *root*, this command is normally run as *root*.

Options

data-base-directory
Names the directory that contains the ToolTalk database to be inspected or repaired.

- If no directory is named, the current directory is assumed.
- If a directory path does not end in `TT_DB`, the `TT_DB` is appended.

The user must have read access to the files in the directory in order to inspect the data and write access to the files in the directory in order to repair the data.

-v
Prints the version of the `ttdbck` command.

Selection Options

The selection options determine which specs in the database are displayed or modified. If no selection options are given, all specs in the database are displayed.

Note – To prevent accidental changes to the ToolTalk databases, only repair option `-I` is allowed unless a selection or diagnosis option is given.

-f ***filename***
Restricts the set of specs to be inspected or modified. When this option is specified, only specs whose files describe objects can be inspected or modified.

Note – If you use shell-style wildcards in *filename*, precede them with an escape (\ ) symbol to prevent the shell from expanding them.

-k ***objidkey***
Specifies a particular spec to be displayed or modified. The object id key can be obtained from a previous invocation of the `ttdbck` command; for example, you can display a set of specs, determine the spec that needs repair, and specify its key with this option.

-t ***type***
Restricts the set of specs to be inspected or modified. When this option is specified, only specs whose otype is *type* can be inspected or modified.

Note – If you use shell-style wildcards in *type*, precede them with an escape (\ ) symbol to prevent the shell from expanding them.

Diagnosis Options

The diagnosis options check for and report on inconsistencies in the selected specs. Only specs specified by the selection options are checked. If a diagnosis option is given, any display or repair option is applied only to specs which fail the diagnostic check.

Note – To prevent accidental changes to the ToolTalk databases, only repair option `-I` is allowed unless a selection or diagnosis option is given.

-b
Checks for badly formed specs; that is, specs that do not contain a filename or type, or specs that have types not defined in the ToolTalk Types Database.

–x
Checks for specs which reference files that no longer exist.

Display Options

The display options determine the data to be printed for each selected spec.

–a
Displays all data. This option is equivalent to specifying:

```
ttdbck -imp
```

–i
Displays the object identifier (including the object id key).

–m
Displays the mandatory data that must appear in every spec; that is, the otype of the object described by the spec and the file in which the spec is stored.

–p
Displays all the properties and values for each selected spec.

Repair Options

The repair options modify the selected specs.

Note – To prevent accidental changes to the ToolTalk databases, only repair option `-I` is allowed unless a selection or diagnosis option is given.

–F filename
Changes the filename for the selected specs to the filename specified by this option.

–I
Invokes the NetISAM `isrepair()` function for all files accessed. The NetISAM function is applied before any other inspection or repair action. Use this option when normal operations return error `EBADFILE`.

–T *obtypeid*
Changes the type of the selected specs to the otype specified by this option.

–Z
Removes the selected specs entirely.

Examples

The examples in this section illustrate three uses of the `ttdbck` command.

Example 1

In the directory `/home/TT_DB`, find all badly formed specs and all specs that reference non-existent files and prints their identifiers.

```
ttdbck -bxi /home
```

Example 2

In the directory `/home/TT_DB`, finds all specs that reference objects in file `/home/sample/data` and change the references to `/home/sample/data1`.

```
ttdbck -f /home/sample/data -F /home/sample/data1 /home
```

Example 3

In the directory `/export/TT_DB`, find all specs that reference objects of type *Sun_Chain_Link* and delete the specs.

```
ttdbck -t Sun_Chain_Link -Z /export/TT_DB
```

Files

/path/TT_DB
The ToolTalk database.

5

ttdbserverd, rpc.ttdbserverd

rpc.ttdbserverd[-n] [-v]

Remote Procedure Call (RPC)-based ToolTalk database server

Description

`rpc.ttdbserverd` is the ToolTalk database server daemon. This process, normally started by inetd, performs all database operations for databases stored on the host.

Options

–n
Turns off permission checking. The protection of the file that contains the spec determines who may read and write that particular spec; however, this option disables the checking for permission and allows anyone to read and write the spec.

Caution – This option allows any file to be over-written. Use with caution.

–v
Prints out the version number for this program and then exits.

Files

TT_DB/*
The NetISAM database files are kept in the this directory under each disk partition mount point.

ttmv

ttmv [–] [–fL] *pathname1 pathname2*
ttmv [–] [–fL] *pathname directory*
ttmv -v
ttmv -h

Move or rename files in a ToolTalk-safe way.

Description

The `ttmv` command invokes the standard operating system command `mv` to move files and directories, and informs the ToolTalk service about its actions. This command ensures that the ToolTalk objects associated with the named files and directories are moved at the same time.

Note – The `ttmv` command moves the ToolTalk objects before it moves the files; however, it does not check whether the object move operation is successful before it moves the files.

Options

-
Treats the arguments that follow as filenames. This option allows you to specify filenames that begin with a minus sign.

-f
Forces a move operation. This option does not report errors and passes the force option to the standard operating system command `mv`.

-h
Prints usage information for the `ttmv` command.

-i
Interactive copy option.

Note – This option of the standard operating command `cp` is not supported.

-L
Moves the ToolTalk objects of the specified files but does not invoke the standard operating system command `mv` to move the actual files.

-v
Prints the version of the `ttmv` command.

Files

***/mountpoint*/TT_DB**
The directory used as a database for the ToolTalk objects of files in the filesystem mounted at */mountpoint*.

ttrm, ttrmdir

ttrm [–] [–frL] *pathname*
rmdir *directory*
ttrm[dir] -v
ttrm[dir] -h

Remove files or directories in a ToolTalk-safe way.

Description

The `ttrm` command invokes the standard operating system command `rm`; the `ttrm`dir command invokes the standard operating system command `rmdir`. The specified files and directories are removed, and the ToolTalk service is informed about the actions. These commands ensure that the ToolTalk objects associated with the deleted files and directories are removed at the same time.

Note – The `ttrm` and `ttrmdir` commands removes the ToolTalk objects before they remove the files; however, these commands perform only a minimal check to verify whether the object remove operation is successful before they remove the files.

Options

-
Treats the arguments that follow as filenames. This option allows you to specify filenames that begin with a minus sign.

-f
Forces a move operation. This option does not report errors and passes the force option to the standard operating system command `rm` or `rmdir`.

-h
Print usage information for the `ttrm` or `ttrmdir` command.

-i
Interactive copy option.

Note – This option of the standard operating command `cp` is not supported.

-L
Removes the ToolTalk objects of the specified files but does not invoke the standard operating system command `rm` or `rmdir` to remove the actual files.

-r
Recursively deletes the ToolTalk objects of any directories specified, and pass the recursive option to the standard operating system command `rm` or `rmdir`.

-v
Prints the version of the `ttrm` or `ttrmdir` command.

Files

***/mountpoint*/TT_DB**
The directory used as a database for the ToolTalk objects of files in the filesystem mounted at */mountpoint*.

ttsession

ttsession [-a *level*] [-d *display*] [-spStvh] [-{E|X}] [-c *[command]*]

The ToolTalk message server.

Description

The `ttsession` command invokes the ToolTalk message server. Each message server defines a *session*. A session is a group of processes that have an instance of the ToolTalk message server in common.

The message server does not have a user interface and typically runs in the background. It is started either by the user's `.xinitrc` file, or automatically by any program which needs to send a message. The message server must be running before any ToolTalk messages can be sent or received.

The message server reacts to two signals.

- If it receives the *USR1* signal, it toggles trace mode on or off.
- If it receives the *USR2* signal, it rereads the ToolTalk Types Database.

Table 5-3 describes the ttsession exit codes.

Table 5-3 ttsession Exit Codes

Code	Description
0	Normal termination. If the `-c` or `-S` option has not been specified, a zero exit status means that `ttsession` has successfully forked an instance of itself that has begun serving the session.
1	Abnormal termination. `ttsession` was given invalid command line options, was interrupted by SIGINT, or encountered an internal error.
2	Collision. Another `ttsession` is already serving the session.

Options

-a level
Sets the server authentication level. The level must be either *unix, xauth*, or *des.*

-c [command]
Starts a process tree session and runs the specified command. The special environment variable `_SUN_TT_SESSION` is set to the name of this session. The default session of any process started with this special environment variable will be in this session. If *command* is omitted, the value of `$SHELL` is used. When this process tree session exits, `ttsession` exits with its exit code.

Note – The `-c` option must be the last option on the command line; any options or arguments that follow the `-c` option are read as the command to be executed.

-d [display]
Directs `ttsession` to start an X session for the specified display. (The `ttsession` command normally uses the `DISPLAY` environment variable.)

–E
Reads in the types from the Classing Engine database.

–h
Prints help on how to invoke `ttsession` and exits.

–p
Prints the name of a new process tree session to stdout, then directs `ttsession` to fork a background instance to manage this new session.

–S
Directs `ttsession` to not fork a background instance to manage its session.

–s
Suppresses the printing of warning messages.

–t

Turns on trace mode. When trace mode is on, the state of a message when it is first seen by `ttsession` is displayed, and then the lifetime of the message is shown as follows:

- Dispatch stage: The result of matching the message against type signatures.
- Delivery stage: The result of matching the message against any registered message patterns.
- Any attempt to send the message to a given process and the success or failure of that attempt.

–v

Prints out the version number of `ttsession` and exits.

–X

Reads in the types from the ToolTalk Types Database. This option is the default.

Environment Variables

Table 5-4 describes the environment variables that can be set.

Table 5-4 Environment Variables for ttsession

Environment Variable	Description
SUN_TTSESSION_CMD	If set, all ToolTalk clients use this variable as the command to auto-start `ttsession`. Although this variable can be set manually, it is typically set by `ttsession` when running in process tree mode.
TTPATH	Tells the ToolTalk service where to find the ToolTalk Types Databases.The format of this variable is: userDB[:systemDB[:networkDB]]
CEPATH	If the `-E` option is specified, tells the Classing Engine where to find the databases that contain ToolTalk types.
_SUN_TT_ARG_TRACE_WIDTH	Specifies how many characters of argument and context values to print when in trace mode. The default is to print the first 40 characters.
_SUN_TT_SESSION	`ttsession` uses this variable to communicate its session ID to the tools that it starts. If set, the ToolTalk client library uses its value as the default session ID.

Table 5-4 Environment Variables for ttsession (Continued)

Environment Variable	Description
DISPLAY	If this variable is set and the `_SUN_TT_SESSION` variable is not set, all ToolTalk clients will use this variable as the command to auto-start `ttsession`. This variable is typically used when ttsession is auto-started while running under OpenWindows.
_SUN_TT_TOKEN	Notifies the ToolTalk client library that it has been invoked by `ttsession` so that the client can confirm to `ttsession` that it started successfully.
_SUN_TT_FILE	`ttsession` places a pathname in this variable when it invokes a tool as a consequence of a message scoped to that file.

tt_type_comp

tt_type_comp [E] [-d {user | system | network}]
-{p | O | P | h | v | {r type1..typeN} | {m | M | x} [-o ofile] file}

The ToolTalk otype and ptype compiler.

Description

The `tt_type_comp` command invokes the ToolTalk types compiler, which compiles the otypes and ptypes in the specified typefile. The *typefile* is first run through the standard operating system function `cpp`, and then parsed and checked for correctness. The types compiler produces files for either an XDR format or the OpenWindows Classing Engine.

By default, the tt_type_comp function merges the types of the given file into the existing user ToolTalk Types Database file.

Options

–d
Specifies the database (*user*, *system*, or *network*) which contains the file to be compiled. The default is *user*.

Note – The three databases form a hierarchy where the definition of a type in the user database overrides the definition of the type in the system database, which overrides the definition of the type in the network database.

Table 5-5 ToolTalk Types Database Definitions

Database	Definition
user	~/.tt/types.xdr
system	/etc/tt/types.xdr
network	$TTHOME/tt/types.xdr

–E
Resets the default format XDR to Classing Engine.

–h
Prints help when the `tt_type_comp` function is invoked and exits.

–M
Merges types into the specified database only if they do not already exist in that database.

–m
Merges types into the specified database and updates any existing type with the new definition. This option is the default.

Note – The -m option is no longer supported for Classing Engine mode.

–O
Prints a list of known otypes.

–o *outputfile*
Outputs to the specified file name.

–P
Prints a list of known ptypes.

–p
Prints the ToolTalk types in the specified database to a file. The types are output in a source format that can be recompiled with the `tt_type_comp` command.

–r type1 ... typeN
Removes the given ptypes or otypes from the database.

–s
Suppresses printing.

–v
Prints the version number and exits.

–x
Produces a compiled XDR format type file.

tttar

tttar [EfhpSv] [*tarfile*] *pathname1pathname2*
tttar [EhpRSv] *tttarfil*e [[-rename *oldname newname*] . . .]
pathname1pathname2 . . .
tttar -v
tttar -h[elp]

Archives or de-archives files and ToolTalk objects.

The `tttar` command has two fundamentally different modes.

1. If the `L` function modifier is not specified, the `tttar` command invokes the standard operating system command `tar` to archive or extract multiple files and their ToolTalk objects onto or from, respectively, a single archive (called a *tarfile*) in a ToolTalk-safe way.
2. If the `L` function modifier is specified, the `tttar` command does *not* invoke the standard operating system command `tar` to archive or extract actual files. Instead, this command archives or extracts only ToolTalk objects onto or from, respectively, a single archive (called a *tttarfile*).

Note – This section discusses the `ttar` command with the `L` function modifier specified; that is, it references *tttarfiles* instead of *tarfiles* and discusses archiving and de-archiving only the ToolTalk objects of the named file rather than archiving and de-archiving both the named file and its ToolTalk objects.

The first (or *key*) argument controls the actions of the `ttar` command. The key argument is a string of characters that contain one *function letter* and one or more *function modifiers*. Other arguments are file or directory names that specify from which files ToolTalk objects are to be archived or extracted. By default, a directory name recursively references the files and subdirectories of that directory.

Note – A file does not need to exist in order for a ToolTalk object to be associated with its pathname. The `tttar` command does not attempt to archive the objects associated with any files that do not exist in the directory.

Note – When you extract a *tttarfile* from a tar archive, the current working directory must be writable so that the *tttarfile* can be placed in it temporarily.

Function Letters

c

Creates a new *tttarfile* and writes the ToolTalk objects of the specified files onto it.

r

This function letter of the standard operating system command `tar` is not supported.

t

Lists the table of contents of the *tttarfile*.

u

This function letter of the standard operating system command `tar` is not supported.

x

Extracts the ToolTalk objects of the specified files from the *tttarfile*. If a specified file matches a directory with contents written onto the tape, this directory is (recursively) extracted. The owner and modification time of the ToolTalk objects are restored. If a *filename* is not specified, the ToolTalk objects of all files named in the archive are extracted.

Function Modifiers

F

This function modifier of the standard operating system command `tar` is not supported.

f

Uses the next argument as the name of the *tttarfile*. If *tttarfile* is specified as a minus (–) sign, the `ttar` command either writes to the standard output or reads from the standard input, whichever is appropriate.

h
Treats symbolic links as normal files or directories. (The `tttar` command normally does not follow symbolic links.)

L
Do not invoke the standard operating system command `tar`.

Note – This function must be used with the `f` function modifier because reading and writing an archived *tttarfile* directly to or from magnetic tape is unimplemented.

---

p
Preserves the original mode of the specified files when used with the `x` function letter. You can also extract setUID and sticky information if you are the superuser.

Note – If the `L` function letter is also specified, this function modifier is disabled.

R
Do not recurse into directories. You must specify the `L` function modifier with this function modifier.

v
Verbose mode. This function modifier displays the name of each file, preceded by the function letter. (The `ttar` command normally does not display this information.)

w
This function modifier of the standard operating system command `tar` is not supported.

X
This function modifier of the standard operating system command `tar` is not supported.

Options

-rename
Reads the next two arguments as *oldname* and *newname*, respectively, and renames any entry archived as *oldname* to *newname*.

- If *oldname* is a directory, its entries are recursively renamed.
- If more than one `-rename` option applies to an entry (for example, because one or more parent directories are being renamed), the most specific `-rename` option applies.

Note – You must use the `L` function modifier with the `-rename` option.

-C
This option of the standard operating system command `tar` is not supported.

-h
Prints usage information for the `ttar` command.

-I
This option of the standard operating system command `tar` is not supported.

-v
Prints the version of the `ttar` command.

Files

***/mountpoint*/TT_DB**
The directory used as a database for the ToolTalk objects of files in the filesystem mounted at */mountpoint*.