NAME

ttsession – the ToolTalk message server

SYNOPSIS

ttsession [–hNpsStv] [–E| –X] [–a *level***] [–d** *display***] [–c [***command***]]**

DESCRIPTION

The **ttsession** utility is the ToolTalk message server. This background process must be running before any messages can be sent or received. Each message server defines a *session*.

The message server has no user interface and typically runs in the background, started either by the user's **.xinitrc** file or automatically by any program that needs to send or receive a message.

OPTIONS

The following options are available:

–a *level*
: Set the server authentication level. The following *level* string values are supported:

  unix The sender and receiver must have the same user ID.

  des The underlying RPC calls use **AUTH_DES**.

–c [*command***]**
: Start a process tree session and run the given command. The **ttsession** utility sets the environment variable *TT_SESSION* to the name of this session. Any process started with this variable in the environment defaults to being in this session. If *command* is omitted, **ttsession** invokes the shell named by the *SHELL* environment variable. Everything after **–c** on the command line is used as the command to be executed.

–d *display*
: Specify an X Windows display. The ToolTalk session will consist of those applications displaying on the named display. The default display is identified by the *DISPLAY* environment variable.

–E Read in the types from the Classing Engine database. If neither **–E** nor **–X** is given, **–X** is assumed.

–h Write a help message to standard error that describes the command syntax of **ttsession**, and exit.

–N Maximize the number of clients allowed to connect to (in other words, open procids in) this session by attempting to raise the limit of open file descriptors. The precise number of clients is system-dependent; on some systems this option may have no effect.

–p Write the name of a new process tree session to standard output, and then fork a background instance of **ttsession** to manage this new session.

–s Silent. Do not write any warning messages to standard error.

–S Do not fork a background instance to manage the **ttsession** session.

–t Turn on trace mode. See **ASYNCHRONOUS EVENTS** for how to turn tracing on and off during execution. Tracing displays the state of a message when it is first seen by **ttsession**. The lifetime of the message is then shown by showing the result of matching the message against type signatures (dispatch stage) and then showing the result of matching the message against any registered message patterns (delivery stage). Any attempt to send the message to a given process is also shown together with the success of that attempt.

–v Write the version number to standard output and exit.

–X Read in the types from the XDR format databases. (Default)

OPERANDS

None.

STDIN

Not used.

INPUT FILES

The XDR format databases listed by the –**X** option are serialized ToolTalk data structures of an unspecified format, except that it is the same as the format of **tt_type_comp**(1) output files.

ENVIRONMENT VARIABLES

The following environment variables affect the execution of **ttsession**:

CEPATH In Classing Engine mode, this variable tells the Classing Engine where to find the databases that contain ToolTalk types. See **ce_db_build**(1).

DISPLAY If *TT_SESSION* is not set and *DISPLAY* is set, then the value of *DISPLAY* will be used by all ToolTalk clients to identify the **ttsession** process serving their X display. If no such process is running, the ToolTalk service will auto-start one.

If **ttsession** is run with the –**d** option and *DISPLAY* is not set, **ttsession** sets *DISPLAY* to be the value of the –**d** option for itself and all processes it forks. This helps ToolTalk clients to find the right X display when they are auto-started by **ttsession**.

LANG Provide a default value for the internationalization variables that are unset or null. If *LANG* is unset or null, the corresponding value from the implementation-specific default locale will be used. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.

LC_ALL If set to a non-empty string value, override the values of all the other internationalization variables.

LC_MESSAGES Determine the locale that is used to affect the format and contents of diagnostic messages written to standard error and informative messages written to standard output.

NLSPATH Determine the location of message catalogues for the processing of *LC_MESSAGES*.

TT_ARG_TRACE_WIDTH
Specify the number of bytes of argument and context values to write

when in trace mode. The default is to print the first 40 bytes.

TTPATH In XDR mode, a colon-separated list of directories that tells ToolTalk where to find the ToolTalk types databases. See **tt_type_comp**(1).

TTSESSION_CMD
Specify the shell command to be used by all ToolTalk clients for auto-starting **ttsession**.

The **ttsession** utility creates the following variable when it invokes another process:

TT_FILE When **ttsession** invokes a tool to receive a message, it copies the file attribute (if any) of the message into this variable, formatted in the same manner as returned by the **tt_message_file**(3) function.

TT_SESSION The **ttsession** utility uses this variable to communicate its session ID to the tools that it starts. The format of the variable is implementation specific. If this variable is set, the ToolTalk client library uses its value as the default session ID.

TT_TOKEN Inform the ToolTalk client library that it has been invoked by **ttsession**, so that the client can confirm to **ttsession** that it started successfully. The format of the variable is implementation specific.

A tool started by **ttsession** must ensure that the *TT_SESSION* and *TT_TOKEN* are present in the environment of any processes it invokes.

RESOURCES

None.

ASYNCHRONOUS EVENTS

The **ttsession** utility reacts to two signals. If it receives the **SIGUSR1** signal, it toggles trace mode on or off (see the –**t** option). If it receives the **SIGUSR2** signal, it rereads the types file. The **ttsession** utility takes the standard action for all other signals.

STDOUT

When the –**v** option is used, **ttsession** writes the version number in an unspecified format. When –**p** is used, **ttsession** writes the name of a new process tree session.

STDERR

Used only for diagnostic messages and the help message written by the –**h** option.

OUTPUT FILES

None.

EXTENDED DESCRIPTION

None.

EXIT STATUS

When the –**c** child process exits, **ttsession** exits with the status of the exited child. Otherwise, the following exit values are returned:

0 Normal termination. Without the –**c** or –**S** options, a zero exit status means **ttsession** has successfully forked an instance of itself that has begun serving the session.

1 Abnormal termination. The **ttsession** utility was given invalid command line options, was interrupted by **SIGINT**, or encountered some internal error.

2 Collision. Another **ttsession** was found to be serving the session already.

CONSEQUENCES OF ERRORS

The **ttsession** utility takes the standard action for all signals.

APPLICATION USAGE

Since everything after −**c** on the command line is used as the command to be executed, −**c** should be the last option.

Tracing is helpful for seeing how messages are dispatched and delivered, but the output may be voluminous.

EXAMPLES

None.

SEE ALSO

tt_type_comp(1), **tt_message_file**(3).