

TaskServer for Win95/WindowsNT

Copyright Johann Weinzierl



What is TaskServer

TaskServer quickview

Installing TaskServer

Configure TaskServer

Security mode

The concept of task-locks

Logging options

The job-file format

Command line options

Praxis tips

The concept of shareware

Registration

What is TaskServer?

TaskServer is a tool that allows you to start any program, batch or command on a WIN95/ WindowsNT Workstation or Windows NT Server, from any computer in the net.

- * very simple to install and use.
- * works with very easy to create `job-files`
- * all machines that can access a shared drive on the Win95 / Windows NT computer can start tasks .
Examples of this are long print jobs, compile jobs,tosser job`s for Fido Sysops, backup of user-workstations and so on.
- * nearly zero cpu stress while running idle.
- * if you set up an FTP Server, even UNIX machines can start tasks in this way.
- * start tasks on the Server over a WAN (RAS) connection to ease remote administration.
- * optional security mode.
- * option to `lock` a task for a specified amount of time
- * multiple task-search directories.
- * various logging-level options.
- * GUI Interface for easy administration

[TaskServer sample screenshot](#)

Installing TaskServer

There is no special installation required to use TaskServer.
Simple copy all the files from the distribution to a directory of your choice.

For WindowsNT users only : If you have not already installed the CTL3D32.DLL on your computer, please move this file to the \winnt\system32 directory.

Configuring TaskServer

A quick look at TaskServer's main dialog box

To ease the understanding of the setup options, let's take a short look how TaskServer works:

In a configurable interval, TaskServer looks into one or more specific directories (called 'job-directories' for files with a special file-extension.

(Default is *.tsc, for 'task-server-command', to change this refer to chapter command line options)

If such a file is found (named a 'job-file'), the first line is read and started as a task.

An example: Let's assume the first line of the file: *test.tsc* looks like:

notepad.exe c:\tmp\mytext.txt

TaskServer starts up notepad with the file **mytext.txt** loaded.

After starting, TaskServer deletes the file *test.tsc*.

So, all you have to do is to write a simple ASCII-file and copy it to a job-directory where TaskServer looks for job-files.

All those setup's can be done by the TaskServer main dialog box

Once done, press the save button to save your settings to disk.

Command line options

There is only a single command-line option:

If you start TaskServer with the `-e<new job extension>` you can change the file extension that is used as a job-file specifier.

Example:

If you want files with the extension `*.abc` to be interpreted as job-files start TaskServer with the following command line:

TaskServer -eabc.

Registration

If you want to get registered please use the file `register.txt`

Please Note: If you are registered the startup `unregistered` dialog-box` does not appear.
Also, the unregistered version expires on 31.10.1995

Once you got the (KEY = Licence number) please open the registration dialog box:



The image shows a Windows-style dialog box titled "Register Dialog". The dialog has a blue title bar with the text "Register Dialog" in white. Below the title bar, the text "Please enter data for registering:" is displayed. There are four input fields, each with a label to its left: "Name", "Street:", "City:", and "Licence number:". Each input field is a simple rectangular box with a thin border. At the bottom of the dialog, there are three buttons: "OK", "Cancel", and "Help". The "Help" button has a small "H" underlined, indicating it is the help key.

Please fill up this dialog exact with the (case-sensitive) values you got from me.

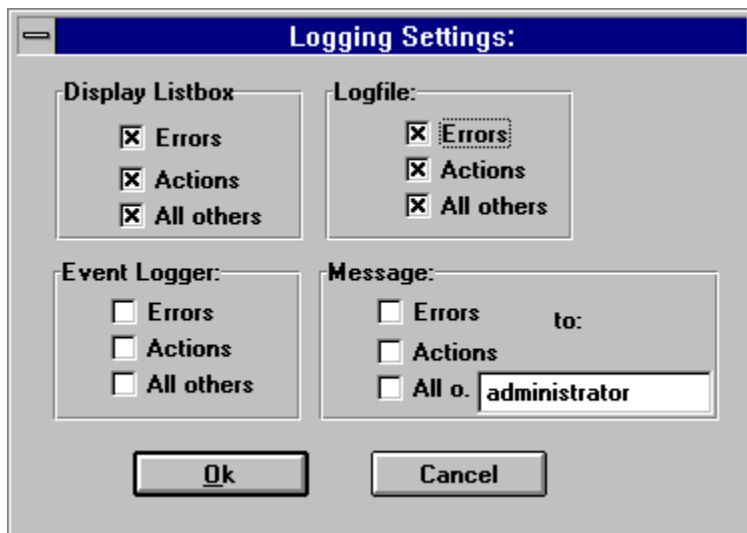
After you have completed this step, exit the dialog by pressing the OK button.
After restarting TaskServer your copy is registered.

Logging options

Each event that occurs in the TaskServer belongs to one of the following 3 categories:

- Action: TaskServer has found a job-file and has executed it.
- An Error: An error has occurred (for example a file specified in a job-file could not be started for some reason).
- Other events.

In the logging option dialog you can specify which of those event shall be reported to a logging-device.



Logging can be done to

The displayed listbox
the logfile `taskserv.log`
the WindowsNT event logger. (Not available in Win95)
as a message to someone in the net.

In case 4, you must specify the receiver of the message, for example `administrator`

Please note:

- If the receiver runs WFW 3.11 he must have `WINPOPUP.EXE` running active.
- Due to a known TaskServer Bug is each entry in the Windows NT event-logger is preceded with a message like:
The description for Event ID (...) in Source (TaskServer) could not be found. It contains the following insertion string(s):

please ignore this message.

Security mode

One basic disadvantage of TaskServer is, that you cannot set permissions on what a specific user is allowed to execute.

Since the task is always started with the rights of the user current logged in, nearly every action can be started by every user.

This basic lack of security can be avoided by enabling the `security mode`

With security mode enabled you must specify an additional security base directory.

Also in security mode the action specifier line is only allowed to be a batch or command file:

Let's make a simple example on how this works:

Let's assume TaskServer has found a task file in a job-directory `d:\users\mike\test.tsc`

Let's further assume the taskfile contains the following line: `makeall.bat`, and your security base directory is `c:\security`.

Now TaskServer extracts the last directory out of the path of those job-directory where the task-file was found (`mike\`) and adds it to the security-base-directory `c:\security`.

TaskServer now tries to start the batch file in this directory.

So in this example that would be:

```
security base dir + last directory from job-dir + batchfilename
= c:\security\ + mike\ + makeall.bat
= c:\security\mike\makeall.bat
```

So the TaskServer tries to start the file `c:\security\mike\makeall.bat`

So since in a typical environment the user mike has access to his home directory `d:\users\mike`, but has no access to `c:\security\mike`, he has no chance to start any tasks not allowed by you the administrator. In this mode you can create a set of batch files for each user, and the user has no chance to execute other commands.

With security mode enabled, the job-specifier line of the job-file **must** contain a `.bat` or `.cmd` file without any path.

Example for valid job-lines:

`makeall.bat`

`start makeall.bat`

Example for invalid job-lines:

`c:\batches\makeall.bat`

(must not contain a path)

`notepad.exe`

(direct executable call is not allowed in security mode)

A job-file is a file with a special extension (default is *.TSC) that is interpreted as a job definition file by TaskServer.

Job-file format

The job-file is a simple ASCII file and always consists of one action line and one or more optional job-description lines.

For example a very simple job file could consist of a single line:

[notepad.exe](#)

To specify more exactly how TaskServer has to start the program there are some optional description lines.

First line is always job-job

All following lines are optional:

A job-description line can be one of the following, job-description lines can be in any order:

Option	Param1	Param2	Description
LOCK	LOCKNAME	TIMEOUT in seconds	<i>Locks a task for the specified time, so that this task cannot be started again for the specified time. 0 means: lock forever</i>
TITLE	"Tasktitle"		<i>Set a new Window title for the started task</i>
SEPARATESESSION			<i>Start a 16 Bit session in a separate VDM (Only valid for Windows NT)</i>
NEWCONSOLE			<i>Creates a new console instead of using the current console</i>
SHOWWINDOW	SHOWNORMAL		<i>Show Window normal</i>
SHOWWINDOW	SHOWMINIMIZED		<i>Show Windows minimized</i>
SHOWWINDOW	SHOWMAXIMIZED		<i>Show Windows maximized</i>
SHOWWINDOW	SHOWHIDDEN		<i>Show Window hidden (Note: Since the task is hidden it is not visible in the TaskManager)</i>
POSITION	left top right bottom		<i>Start window at the specified position</i>

Example 1:

The following example starts the batch `makeall.bat` with the title "test batch file" as a minimized window and lock the task with the name `MAINOPERATION`, with timeout 600 seconds:

```
c:\batch\makeall.bat
LOCK MAINOPERATION 600
TITLE ``test batch file``
SHOWWINDOW SHOWMINIMIZED
```

Example 2:

If you want to start the 16 Bit write.exe as a separate session with a specified window position at startup:

```
write.exe
SEPARATESESSION
POSITION 30 30 350 400
```

Support for the start command available on WindowsNT

You can also use the WindowsNT `start` command (and all it's options) to start a task. Please note that the locking mechanism even works if you use the start command.

The syntax of the start command is like:

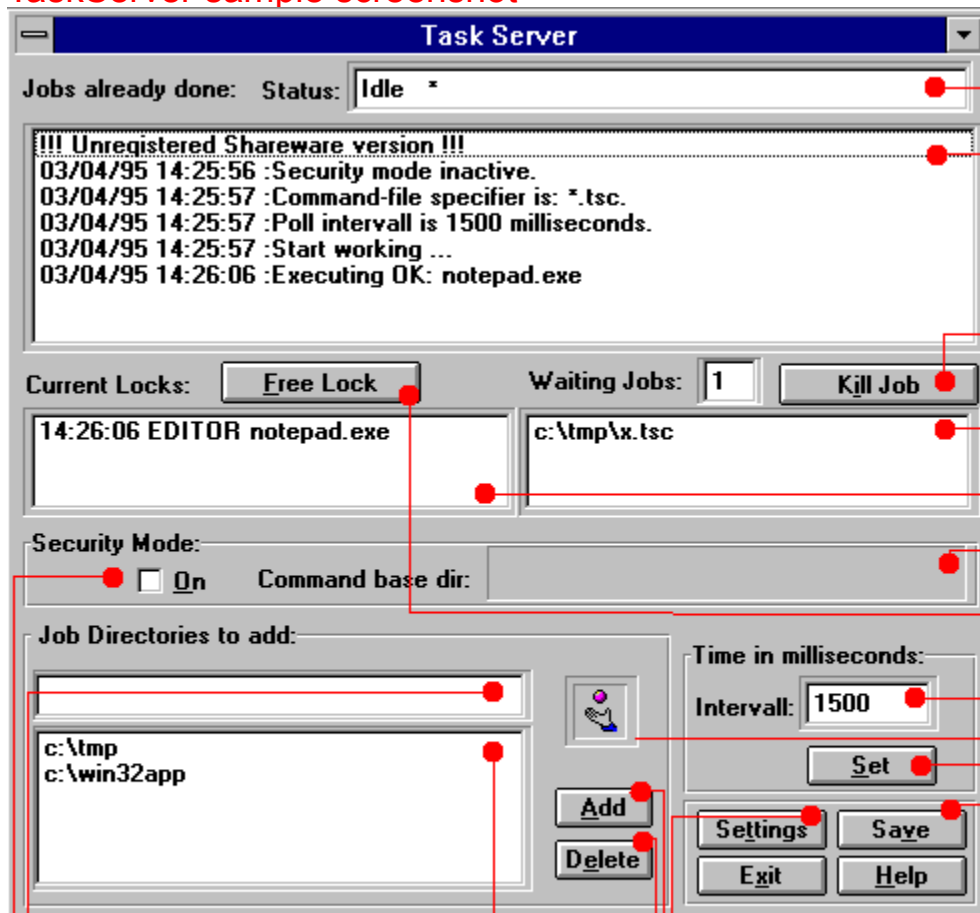
```
START ["title"] [/Dpath] [/I] [/MIN] [/MAX] [/SEPARATE] [/LOW | /NORMAL | /HIGH | /REALTIME] [/WAIT]
[/B] [command/program] [parameters]
```

For further details of the Windows NT start command please refer to the WindowsNT online reference.

Example 3:

```
start "test"/MIN /LOW c:\batches\test.bat param1 param2
```

TaskServer sample screenshot



Input line for job-directories

Current job-directories

Security Mode Button

Logging options dialog

Add job directory

Delete job directory

Status Line

Display Listbox

Button to kill a waiting job

List of currently waiting jobs

Current locked jobs

Security base dir (inactive)

Free a specific lock

Poll time interval

About and registration dialog

Set new intervall time

Save current configuration

The concept of task-locks

The problem: Let's assume you have some task that must not run more than one instance at a time. (Backup scripts, Database updates and so on).

Solution: Specify a additional description-line in your job-files to LOCK the task for a speciefied amount of time (or even forever with 0).

Example: job-file `update.tsc`:
UPDATE.BAT
LOCK DATABASEUPDATE 600

Now every new job-file that specifies the same lockname (here: DATABASEUPDATE) will not execute until:

a, The lock is freed by the task that owns the lock.
This is done automatically when this task ends.

or:

b, The timeout occurs (here 600 sec)
After this time the next waiting job is started, regardless whether the first task has freed the lock or not. The new task now owns the lock.

or:

c, The user has freed the lock manually by clicking the appropriate button in the main dialog box.

A task that cannot run immediately because it has to wait for a lock will be scheduled as a `waiting` task. You can see the current waiting tasks in the main dialog box.

Praxis tips

*** Do not try to create a job-file directly in a job-directory. Dependant on the behaviour of your ASCII editor it is likely that the editor creates an empty or incomplete file at startup or while saving a temporary copy.**

TaskServer will generate an error if such a file is found.

Tip:

*** create a set of files containing the job you want to execute and save them with for example `doit.tsj` extension. Now you can use a simple batch ``copy doit.tsj doit.tsc` to start the job.**

*** Alternativ you can save them under `*.tsc` extension, but into a directory that is not specified as a job-directory. If you want to execute, simply copy the file to a job-directory.**

The concept of shareware

TaskServer is distributed and marketed via a method known as shareware.

The main characteristic of shareware is that the vendor of software allows a user to try a product before they actually buy it. The amount of time a user is allowed to try the software is known as the evaluation period.

At the end of the evaluation period (31.10.1995 for this release of TaskServer) the user of the software is required either to buy the software or to stop using the software.

A side effect of the shareware method is that shareware vendors actually encourage you to copy and share the software with your friends and associates. In this manner, they too can decide whether or not the software fulfills their needs in a way that justifies its price.

Users that have paid are called "registered" users and receive a "Licence code" that allows them to use the software free of payment reminder screens and other forms of reminders.

I have designed my registration code with an additional feature:

Later, if you acquire a new update of the product through shareware distribution channels then you will be able to use the same registration code with the new version.

If after trying the TaskServer you decide they meet your needs, the file register.txt provides the information necessary to purchase the product.

Thank you very much for using my products, and feel free to contact me whenever you want.

Johann Weinzierl
Eichstätt 29
84098 Hohenthann

Fido: 2:2494/17.1
email: weinzij@kontron.de
Compuserve: 100531,401 Johann Weinzierl

