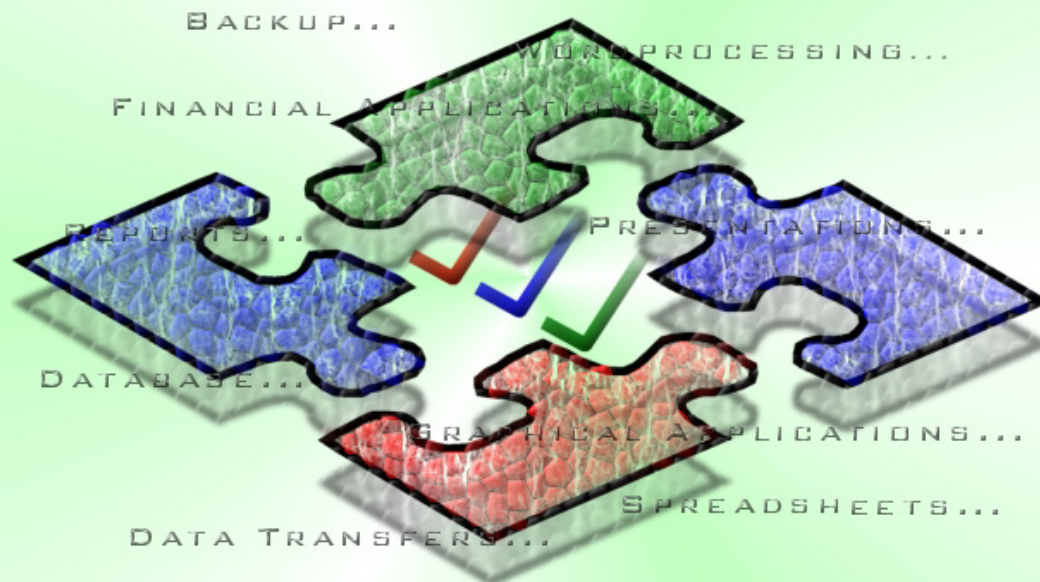


TASK3000



Batch Processing For The New Millennium

Contents

Contents	2
Task3000 Family of Products	3
Concepts and Planning (Version 1.3)	3
Running the Job (Introduction)	3
Figure 1: Job definition	3
Figure 2: Task Network	5
Task3000 Design	6
Figure 3: Task3000 users	8
Products	9
Task3000 Menu	9
Figure 4: Task3000 Menu	9
Figure 5: Task3000 Menu Schema	10
Task3000 Submitter	11
Figure 6: Task3000 Submitter	11
Task3000 Workstation	12
Figure 7: Task3000 Workstation	12
1. Command	13
2. File RC (Changed from Version 1.0)	14
Note:	16
Maximum RC is 8 for the given examples (general parameters)	16
Example 1:	16
Example 2:	16
3. Working directory	16
4. Environment Variables	16
Workstation usage Tips and Tricks	18
Task3000 Personal Server and Task3000 Server	20
(Full Version for Windows NT 4.0 and Windows XP/2000/2003 only, Limited Version for Windows 9X/ME)	20
Figure 8: Task3000 Server	20
Execution of the Tasks	22
Task Queue	22
User Parameters (Environment Variables)	23
Conclusions	24

Task3000 Family of Products

Concepts and Planning (Version 1.3)

Running the Job (Introduction)

Job is complex process, which user performs to achieve some business function. Job can be divided to one or more tasks, and the task can have one or more steps:

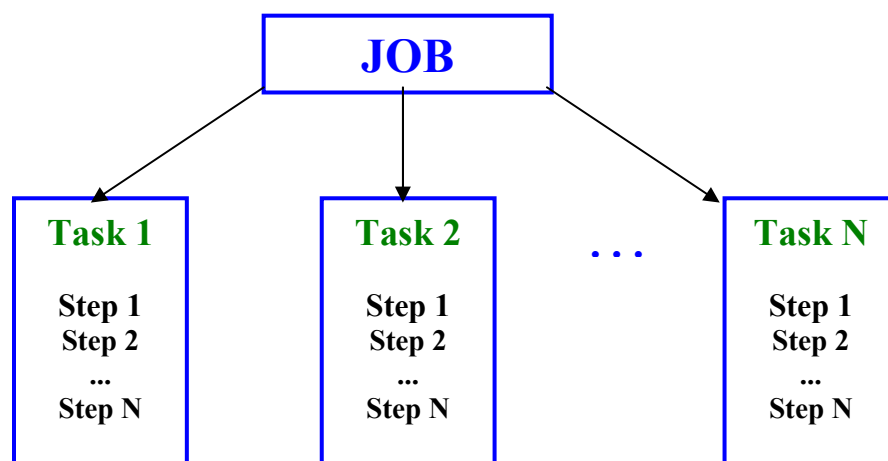


Figure 1: Job definition

Such situation is very much simplified; it does not show the time dependency between tasks. Between one task and another usually we have to perform some activities, like data collection. These activities are in large organizations performed by many different peoples.

The situation is also complicated with the fact, than many computer programs are involved to perform the tasks. Programs are not only different between tasks, but also between the steps in the task. Today's programs tend to be more and more universal, but in spite of that it is not possible to perform everything with one program.

Many computers are involved in the process, usually we have computer networks. Within computer networks we have many workstations, which are used to perform:

1. Data entry
2. Data analysis
3. General Processing (Graphic, Sound, Video, Data Compression/Decompression)
4. Printing
5. Data storage

Also we have usually one or more server to perform these typical functions:

1. Data storage
2. Data analysis
3. Printing
4. Dedicated processing (like fax, communication, call center, answering machine, fast data processing etc.)

More realistic situation is shown in the following picture:

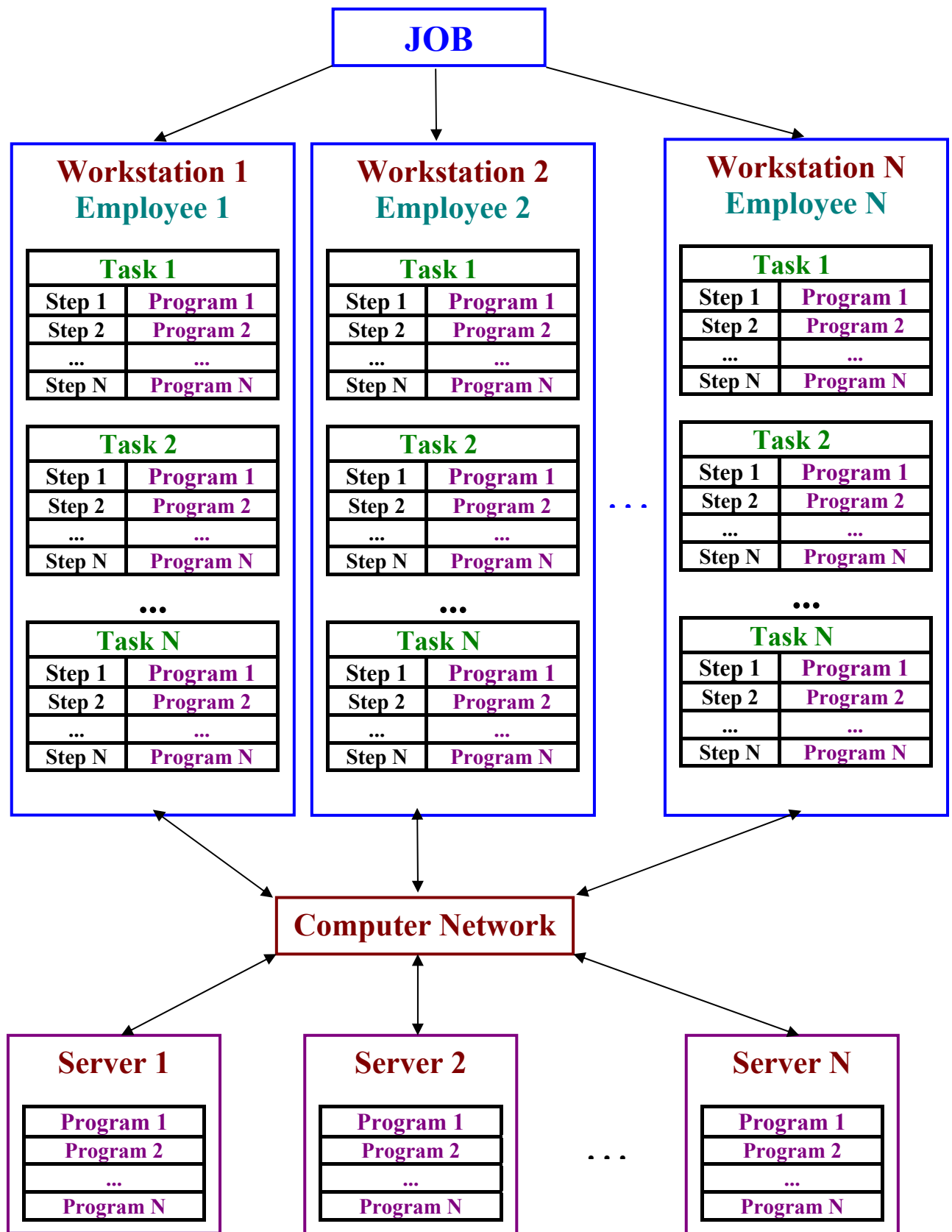


Figure 2: Task Network

The main purpose of the Task3000 program is to **integrate** all tasks into easy **manageable system**, which needs only **necessary parameters to do the job**. Also it enables the functionality to automate the process and employ computers to perform management of the whole computer network.

Task3000 Design

Task3000 package is designed to give the users of the Microsoft Windows operating systems (Windows 98 or higher, Windows NT 4.0 or higher) the ability to create and run the job by the functionality of the batch processing. Functionality of the Task3000 package is:

1. User can create and modify the desired task, which is the basic work unit
2. Desired task can consist of one or more steps
3. Task can have various number of parameters
4. Parameters can be set by the graphical user interface (menu system)
5. Parameters are checked by some simple checking routines for correctness before execution
6. Run the tasks on the same machine or on the server
7. Job can have time schedule and can be automatically rescheduled according to rescheduling plan
8. Task3000 Server creates log files about the task executions, which can be cleaned according to queue cleaning plan
9. Task step execution can be controlled according to the File RC (Return Code) which is set according to the file contents
10. Time limitations according to the duration of the task, task CPU time consumption, maximum memory usage and so called System Units, which consist of the used CPU time, used memory and with Windows XP/2000/2003 used Input/Output operations.
11. Manage some security issues on the server queue automatically

All that serves as the background to fully automated tasks, which can be run on the same machine or on dedicated server for batch processing. Reason for the usage of the dedicated server can be:

1. Releasing the workstations for other tasks. Reason for that could be long task execution or desire to have logs about the tasks.
2. Faster network access to for instance database servers in case that communication between servers is 100MB or 1GB, and between workstations and servers 10MB or 100MB etc.
3. Saving the money for the licenses. For instance dedicated Task3000server can ran on the windows NT workstation and use workstation licenses, which are generally cheaper than the server licenses.
4. Saving the money by having only one or more licenses for a given product, and enabling the users to submit the tasks for execution on the dedicated server. However, such approach is not possible if the users should get the answers interactively.
5. Maintenance of the networks by using smart task processing, manipulating the backup is possible etc.

There is a difference in the possibility of the usage Task3000 on the same machine (workstation) or on the server. On the server program cannot prompt the user for the reply, and only the server administrator can only do the reply to the question. Otherwise, task can be cancelled by the user or some limit (time, CPU etc.). However, on the workstation such situation is possible.

There are three different users of the Task3000 package:

1. Task3000 Designer
2. Task3000 Menu user
3. Task3000 Server Administrator

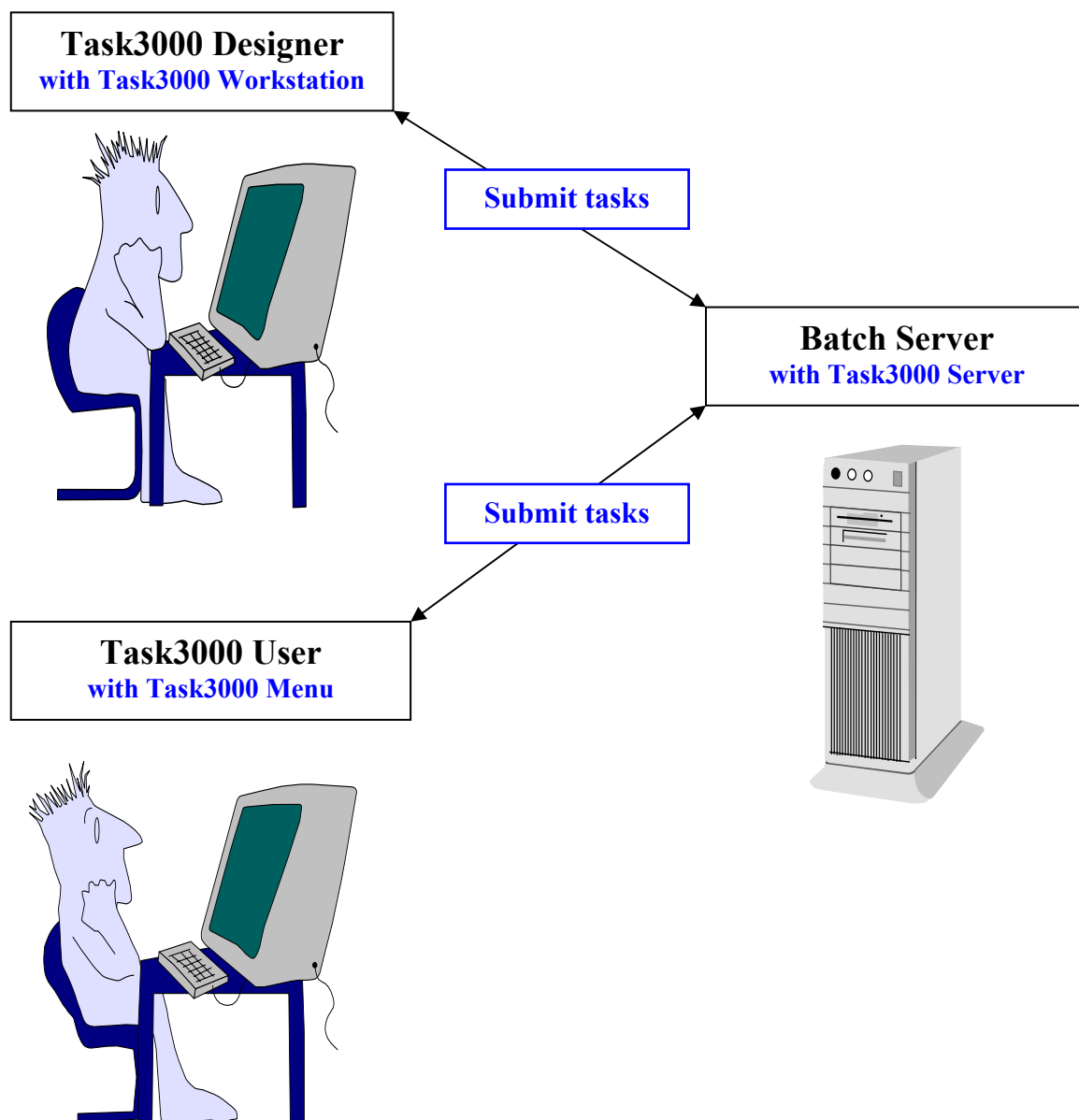


Figure 3: Task3000 users

Of course in the simplest case this can be the same person, but the advantage is by the saving money by scheduling the desired job among many people, which are Task3000 Menu users. Task3000 Menu users can be less educated peoples, which perform the job routinely.

Tasks are prepared by the Task3000 Designers, which must be aware of the technology of the task step execution on the network environment. Symbolic environment variables are used to control the process and the knowledge of the network architecture is elementary, therefore much higher education of the people is needed.

Task3000 Server Administrator can only be the person who runs the Task3000 Server (one or many) and is responsible for the Task3000 Queue processing and backup, but can also implement the Task3000 Menu program, with which enable the users to submit the tasks or Task3000 Workstation program which enable the Task3000 Designers to design and submit the task etc.

Products

Task3000 Menu

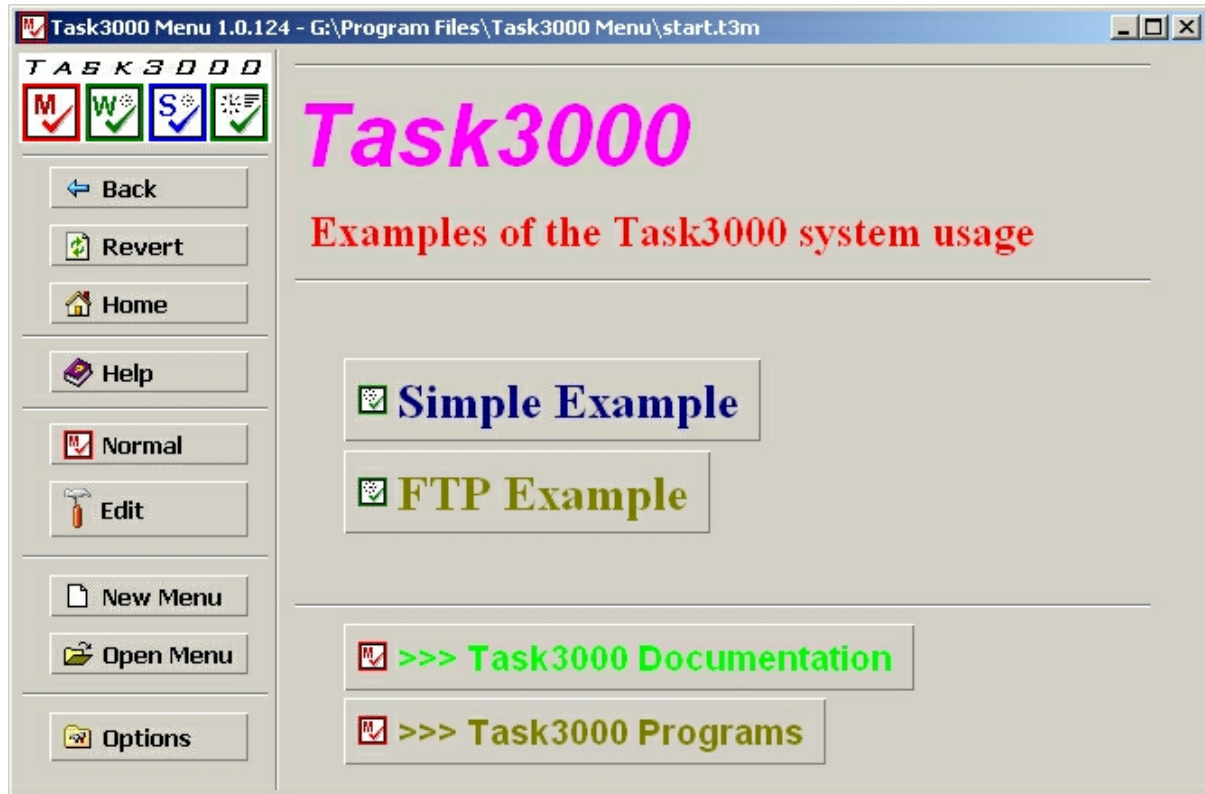


Figure 4: Task3000 Menu

Task3000 Menu is used to generate the tree level structures above tasks and other programs. Menu can be generated by Task3000 Menu, which enables you to generate the desired menu system.

Design of the Task3000 Menu system is such that requires as little data as possible to generate the menu. Elements of the menu are:

1. Pointer to the file, which have an executable program. This can be also indirect, for instance file with .DOC extension Task3000 Menu will call MS Word etc.
2. Pointer to another menu file
3. Pointer to T3Sub.exe and Task3000 .T3F file, which enables you to submit the task with parameters. Parameters are actually Environment Variables for the task and are shown by using menu generated from the Task3000 Workstation .T3F file using Environment Variables for the steps. Special variable is used instead of step number to enable the selection of the step execution.
4. Label is any label
5. Space is used to add space between elements
6. Line is used to add horizontal line between elements

Note: pointer is full path to the element, and can contain Environment Variable like %SystemRoot% or %WinDir% etc.

The functionality of the Task3000 Menu can be seen from the following picture:

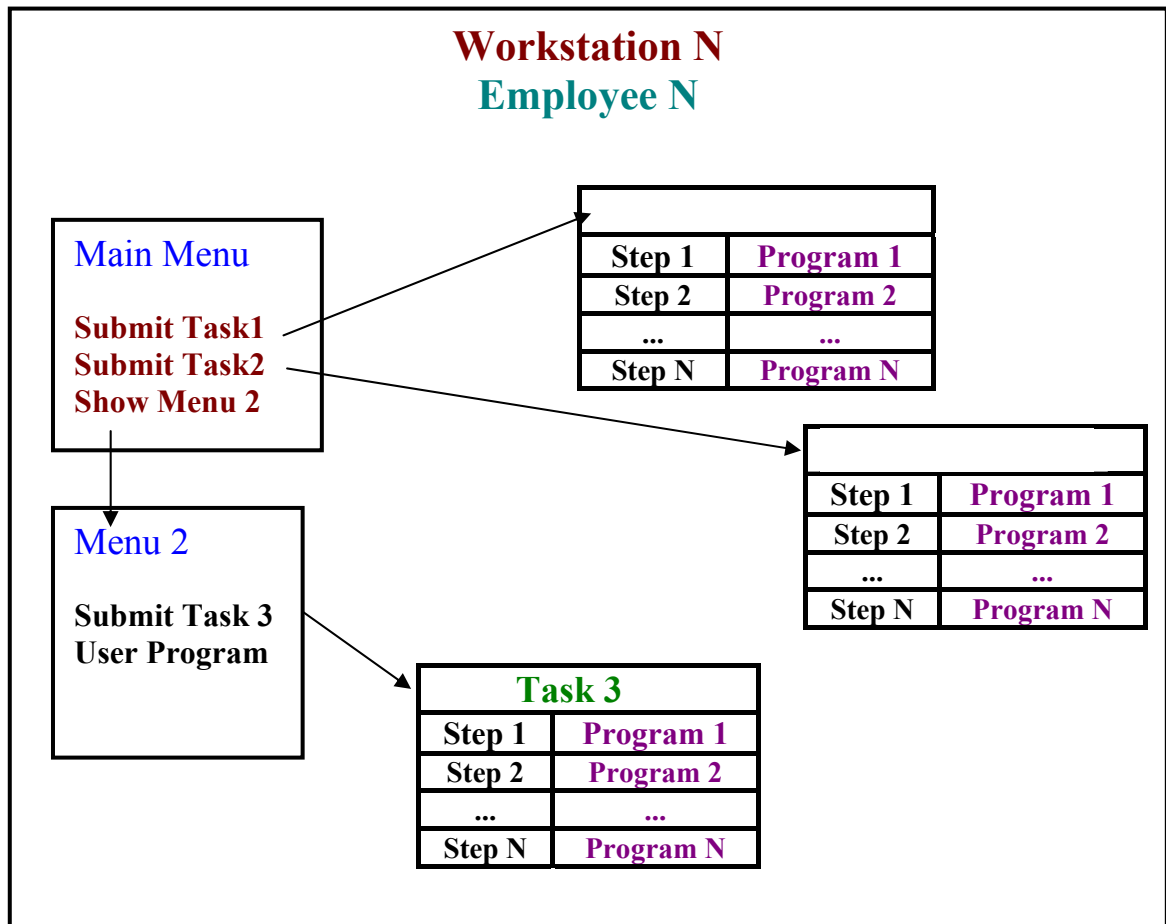


Figure 5: Task3000 Menu Schema

Task3000 Submitter

Task Name: GenTotal

☒ COPY ☒ Execute

Printer: Lexmark 7000 Series ColorFine: \\WIN2000\Lexmark7000

Report file: \\Program Files\Task2000 Examples\Report1.csv

Select Word Version: Office2000: Gentotal.doc

☒ Excel Cscript ☒ Execute

Visible: ☒ Check

☒ Excel Command Line ☐ Execute

☒ Word Cscript ☒ Execute

Number Of Environment Fields: 7

Figure 6: Task3000 Submitter

Task3000 Submitter is used to send the proper parameters to the task. Parameters for generation of the menu are generated by Task3000 Workstation program. Simple checking is performed, like test for numerical data, limiting the entered parameter by minimum and maximum, checking the date, selecting the files by menu and checking the existence of the file etc.

Following types of parameters are supported:

1. Text
2. Number
3. Date
4. Time
5. Combo box
6. Check box

Combo box can be used to enter the data from the lists, like selection of the desired printer, simple codebooks etc.

Task3000 Workstation

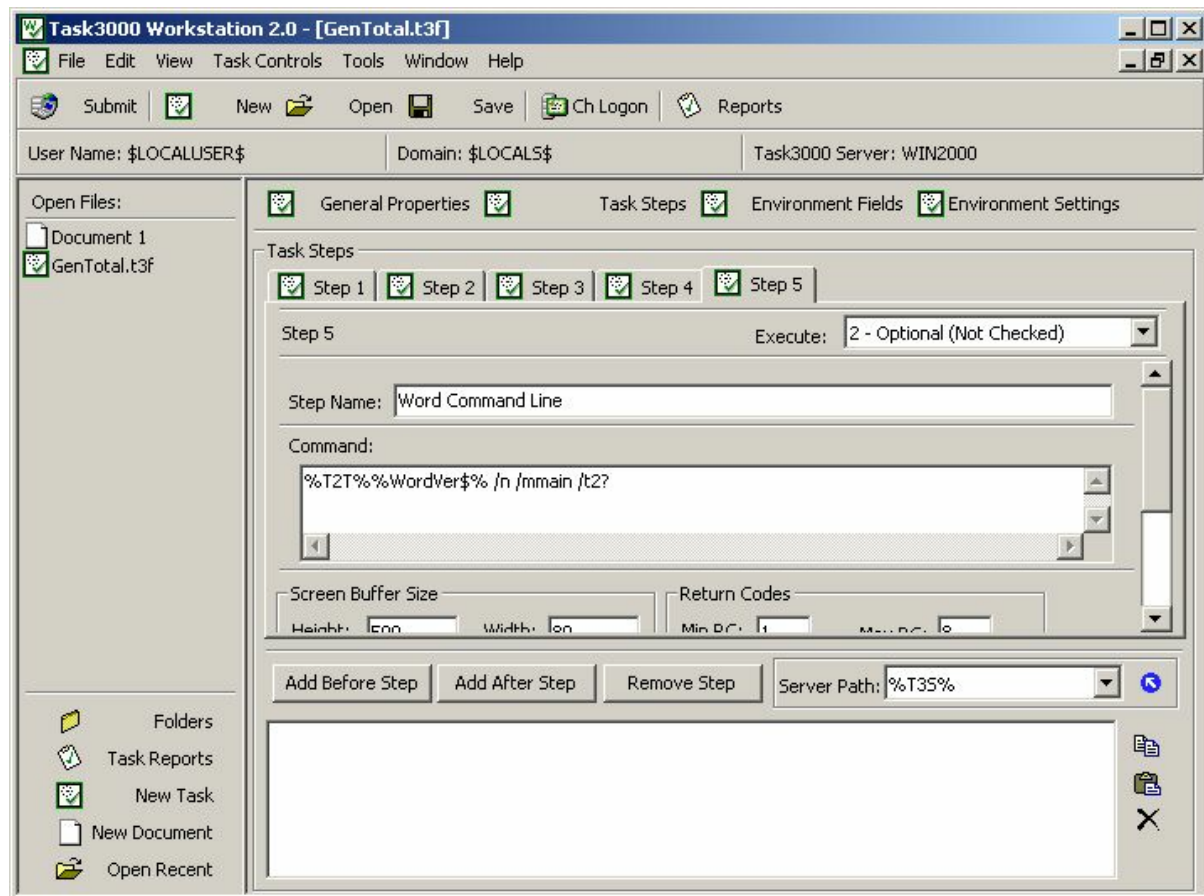


Figure 7: Task3000 Workstation

Task3000 Workstation is used to generate and submit the task by task designer. Task3000 Personal Server is used to execute the submitted task on the same machine; Task3000 Server is needed to execute the task on the same or different server.

Task is basic unit of the execution and is executed by the Task3000 Server. It can contain one or more steps. Steps are executed sequentially.

Every task have general parameter, like task name, execution time, number and conditions for repetition of the task, task dependency etc.

Every step has four basic elements:

- 1. Command**
- 2. FileRC**
- 3. Working directory**
- 4. Environment Variables**

Example:

Note: FileRC in Step 2; Find text "COPY SUCCESSFUL" in file C:\temp\test, execute before step, search file from the beginning forward, set RC to 9 if text not found

Step No	Command	FileRC	Working directory	Environment Variables
1	%COMSPEC% /C COPY C:\A.bat C:\TEMP\TEST		C:\TEMP	SET Name=USER; SET File=C:\TEST;
2	DIR >C:\TEMP\TEST	-> F=(R,"C:\temp\test">=0<>) S=(0,"COPY SUCCESSFUL") V=(0,0>=" "<"") R=9 N=0 P=0 Q=0		

1. Command

Command can be executable program of programs like shell. Windows NT has various shell programs like CMD.EXE, COMMAND.COM, WSCRIPT.EXE and CSCRIPT.EXE to mention just few most popular.

CMD.EXE is Windows NT version of the COMMAND.COM, which is DOS version of the command processor. Please refer to the Windows NT help for detailed explanation of this shell's. Normal usage with Task3000 is with parameter /c and command like:

CMD /c dir (Windows NT) or COMMAND.COM (Windows 98) /c dir

to show the current directory contents. Recommended usage is:

%COMSPEC% /c dir

which works on all platforms.

WARNING! Usage of the /k parameter instead of /c lead to execution of the command, but the program does not close the current window. User should close the windows with the **EXIT**, but if the task is executed on the different machine (server), than user cannot do that. User can only issue **CANCEL** command and the task will be terminated.

WSCRIPT.EXE and **CSCRIPT.EXE** are the same version of the programs; only first is Windows version and second is DOS version. DOS version is recommended to use with Task3000, because it can run on the server with minimum changes.

2. File RC (Changed from Version 1.0)

With **Return Codes (RC)** we can control the execution of each step and task.

FileRC table of possible values

0	Stop the task after this step
1-Max RC	Task normal codes
2	Null Command
>Max RC - 99	Task abend codes
99	System Cancel task code, Invalid program

From the previous table we can see that **MaxRC** (Maximum Task Return Code) divide the task execution into normal part and abnormal part.

Whether step will be executed or not is determined by Step MinRC and Step MaxRC. Possible values are in the range of **Task normal** and **Task abnormal codes** that is from **1-99** with condition that **MaxStepRC** \geq **MinStepRC**. By that step can be executed:

always if

MinStepRC=1 and MaxStepRC=99

during normal execution if

MinStepRC and MaxStepRC \leq MaxRC

during abnormal execution if

MinStepRC and MaxStepRC $>$ MaxRC

or any other given range from **MinStepRC=1.99 \leq MaxStepRC=1.99**

But how the **actual TaskRC** is set? Searching given file for some conditions, which can be seen from the following table, does this:

Condition	Value	Condition is
File Exist	True/False	Mandatory
File Attribute is in range	RSHA	Optional
File Size is in range	0-2GB	Optional
Text found on the file	Text	Optional
Text is on the given column	0 or 1-Length of the line	Optional
Text Occurrence is correct	0 or 1st, 2nd, 3rd etc.	Optional
Value1 after Text + Offset* is in range	Text or Number	Optional
Value2 after Text + Offset* is in range	Text or Number	Optional

* **Offset** is from the last **Text** character position and can be 0

Additionally file can be **searched** from **beginning to the end** or **reverse**. In case of **backward** search also **Text Occurrence** is from the **end of the file**.

Quantity determine how much of the file will actually **be searched**.

Caution: Default is usually set to 1MB. Increase the number in Quantity value field if necessary.

Position can be used to **set starting position** within the file.

File attributes (A-archive, R-read only, S-system, H-hidden) can be checked to have **exact value (=)** or **only one of the attributes (<>)**.

All of the **conditions** are checked in **increasing order**. At the **beginning** they are all set to **neutral values** and do not influence the checking. If we select some other value, then value checking take place.

Text is therefore treated as a **key value** in the file (if we use the feature) and we are searching to find the **Value1 and/or Value2** at the **Offset** position from the **key** (also optional).

User can select appropriate **Return Code**. **TaskRC** code will be set to **RC** in case of:

Positive Logic if **all** of the selected conditions are **True (conditions are met)**

or

Negative Logic if **any** of the condition **False (conditions aren't met)**

Note1: FileRC can be called **before step** execution **and/or** at the **end of the step** execution.

Warning: FileRC File directory is not by default step Working directory!

Note2: To be able to test if file exist File Size must be set to >0 or <>0. If file does not exist it is treated as 0. However, this is different as if the file with length=0 exist!

Note3: TaskRC=0 means the termination of the task immediate after current step (also this mean no further repetition of the job if set by appropriate parameter).

Note4: By deleting the Task3000.ctf files in %T3S% in %T3T% system will reassign the full rights to administrator and specific users (reset to default)

Note:

Maximum RC is 8 for the given examples (general parameters)

Example 1:

Step	Command	StepMin RC	StepMax RC	File RC	Task RC	File Value	Condition	Step Executed
1	Program 1	1	8	-	1			Yes
2	Program 2	1	8	2	2	X=P	X=C	Yes
3	Program 3	5	8	-	2			No
4	Program 4	1	8	7	7	Z=6	Z>=7 Z<9	Yes
5	Program 5	7	7	9	9	NO	OK	Yes
6	Program 6	5	5	-	9			No
7	Program 7	6	6	-	9			No
8	Program 8	9	99	-	9			Yes (but abend, TaskRC>MaxRC)

Example 2:

Step	Command	StepMin RC	StepMax RC	File RC	Task RC	File Value	Condition	Step Executed
1	Program 1	1	8	3	3	DO	DONOT	Yes
2	Program 2	1	8	-	3			Yes
3	Program 3	5	8	5	5	FALSE	TRUE	Yes
4	Program 4	1	8	-	5			Yes
5	Program 5	7	7	-	5			No
6	Program 6	5	5	-	5			Yes
7	Program 7	6	6	-	5			No
8	Program 8	9	99	-	5			No

3. Working directory

Working directory is the directory, which is searched first for the desired program. Users should be aware of the fact that directory is relative to the machine on which Task3000 Server run the tasks. Because this can differ very much from server to server, user can use Environment Variables set by Task3000 Server instead of true path (%T3T% or %T3S%).

4. Environment Variables

Environment Variables are part of the MS Windows system, which enables the users to set their variables, which can be then used by various programs, which can access the variables. Task3000 can set the Environment variables for the given step according to the user specification in the form:

SET Variable=Value;

Example:

SET X=5;

These variables can be set by Task3000 Environment Fields Menu system easily.

As a part of all **value fields (Command, Working Directory, FilesRC or Environment Variables)** we can use

%T3% Task Variables

which are automatically generated during the step execution:

%T3T%	path to the Task Name directory
%T3S%	path to the Step Number directory
%T3D%	Directory number - text format
%T3N%	Directory number - number format
%T3Max%	Maximum Return Code
%T3RC%	Current Maximum Return Code
%T3RepC%	Current Repeat Counter (0 if repeat always or once)
%T3NAME%	Task name -project name
%T3Step%	Current Step Number
%T3User%	Task User
%T3W%	Step Working Directory

Examples with the Set command of CMD or COMMAND shell:

```
T3T=Q:\$JQ\LOCALS\MIRAN\MyProject\  
T3S=Q:\$JQ\LOCALS\MIRAN\MyProject\000000153\  
T3D=000000153  
T3N=153  
T3Max=8  
T3RC=1  
T3Name=MyProject  
T3RepC=0  
T3Step=2  
T3User=MIRAN  
T3W=Q:\TEMP
```

Note:

It is possible to use variables together:

Example 1:

%T3T%%T3D% give result similar to %T3S% but without backslash:

```
Q:\$JQ\LOCALS\MIRAN\MyProject\000000153
```

Example 2:

%T3S%Myfile%T3N%.txt and step number is 2:

```
Q:\$JQ\LOCALS\MIRAN\MyProject\000000153\Myfile2.txt
```

Task is saved with the **.T3f** extension. The file contains all necessary data collected by the appropriate menu system. The data can be divided to the following items:

1. **Security data (Username, Password, Domain, Active Directory Service etc.)**
2. **Job data (Job Name, Job Schedule Data, Maximum RC etc.)**
3. **Step data (Command, FileRC, Environment Variables, Working Directory, Step RC etc.)**
4. **Information about necessary files to copy to the server**

Workstation usage Tips and Tricks

1. Use of Right Click Mouse Button is similar to that of Explorer, however **drag and drop** is limited to:
 - a) Drag and drop from Folders TreeView to the Open Files will always open the files with .rtf local editor. Use Right click+Open to use file with associations.
 - b) Drag and drop the file from Folders TreeView to the Folders Address field. Use Copy + Paste to copy the path to other parts of the system
 - c) Drag and drop from (New) Task + Task Steps Open files to the Step File field
 - d) Use buttons under Open Files to fast switch between documents and Task3000 Buttons (Folders, Reports etc.)
 - e) In the Reports Windows use View Task, View Task Folder on Server and Up One Level button to switch between Project, Project %T3S% and Project %T3T%. Very useful is Right Click Button on the mouse. You can for instance Open the file, delete the file etc. If this is not sufficient use Tool+Map Network Drive and connect to the \$JQ share on the server. This is only way to delete Project folder.
 - f) Microsoft allows only one connection with given user name to individual server. It is necessary drop the connection and connects with different user name instead. On Win9X and ME systems it is necessary to do Logoff+Logon with different user name. Sometimes it is necessary to use DOS prompt and command:
Net use \\server /delete
where server is Task3000 server Name.
 - g) Use Folders ListView+Right Click to copy the files and (New) Task + Step File Button to paste the files into Step File Fields.
 - e) Use Folders ListView+Right Click+select files. Use Right click+to copy the file(s) to Clipboard. Open New Document and use Paste. Selected File names will appear in the document.
 - f) At least Internet Explorer 4.0 is mandatory! Also latest Service Pack's of the system is recommended (at least after it is verified by users after 2-3 month). Use SP4 and later for Windows NT 4.0!
 - g) **Task3000 is reusing Microsoft WebBrowser technology. Because of that it is necessary to set the Options for Internet Explorer to:**

Tools+
Internet Options+
Security(Local Intranet)+
Custom Level+
Launching Programs and Files in an IFRAME set to **Enable**

If this option is not set then there is problem with the right click mouse functions, especially on Folders and Reports menu.

Task3000 Personal Server and Task3000 Server

(Full Version for Windows NT 4.0 and Windows XP/2000/2003 only, Limited Version for Windows 9X/ME)

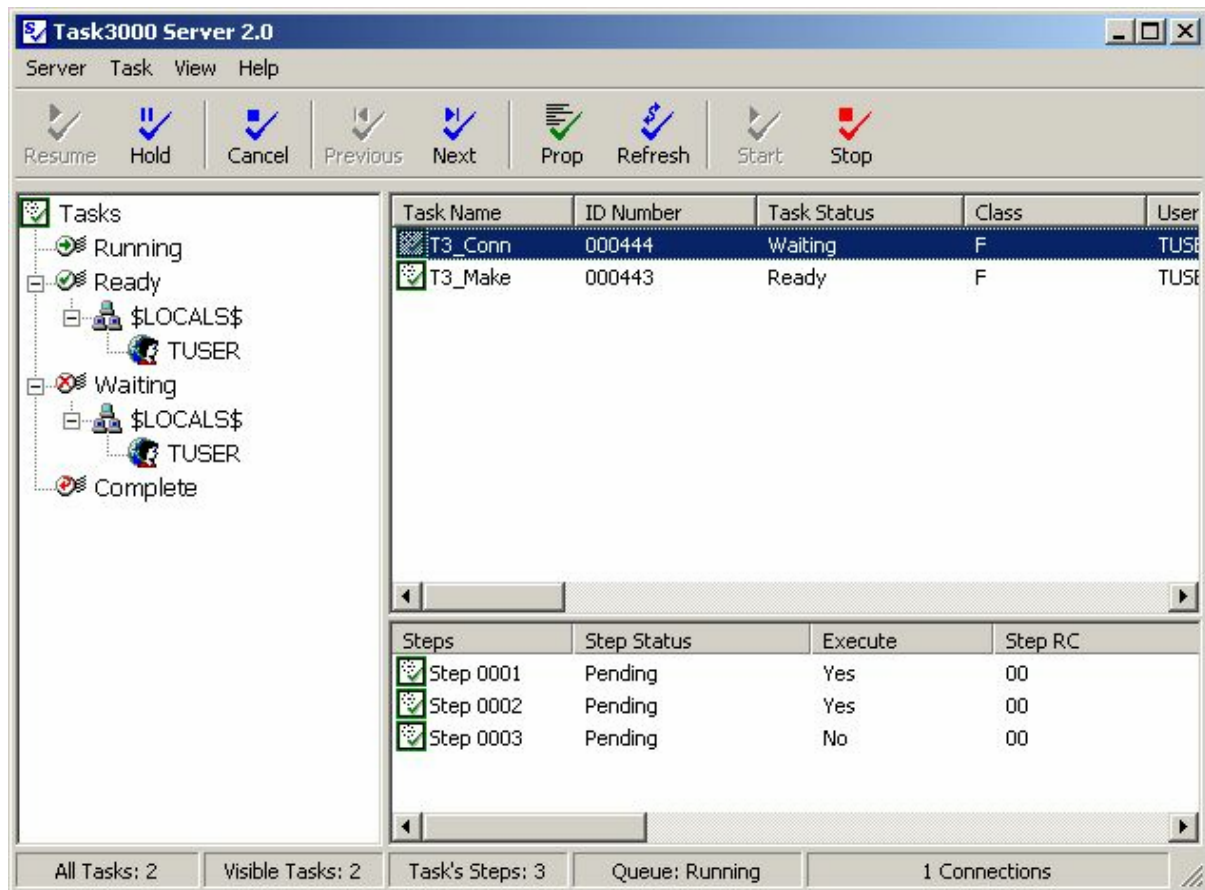


Figure 8: Task3000 Server

Task3000 Personal Server on Windows 9X/ME has limited functionality due to operating system restrictions.

Known limitations on Task3000 Personal Server on Windows 9X/ME has limited functionality due to operating system restrictions:

- 1) Console size can be only 80x25, 80x43 or 80x50 to run similar to normal usage on Windows NT/ Windows XP/2000/2003. With larger size you should redirect output to the file. E.g. to use `COMMAND.COM /C Your_Command` it is recommended to use `%COMSPEC% /C Your_Command >%T2S%Your_file`
- 2) Cancel, Hold and Resume task works on the next step (current step is not canceled, hold or resumed)

Task3000 Server and Personal Server on Windows NT4.0 cannot collect system input-output (IO) data. System IO is collected only during FileRC operations.

Task3000 Personal server is limited comparing with Task3000 Server on such way that only one Task at the time can be executed and also security is not enabled (UserID, Password, Domain, ADS) There is also no security on the Task Queue. However, it can run in the context of the user while administrative rights are needed to run Task3000 Server. Task3000 Server by default set Full rights on all \$JQ directories and Full Rights to User Assign Directories (Project, JobNO) is assigned to the User (User_Name directory RXD,X).

Warning: Task3000 server need beside all other standard Administrators rights:

- 1) **Act as a Part of the Operating System**
- 2) **Replace a Process Level Token**

Example:

Windows NT 4.0 domain: Start + Programs + Administrative Tools + User Manager for Domains + User Rights Policy (check Show Advanced User Rights!)

Win XP/2000/2003 local system account: Start + Setting + Administrative Tools + Local Security Settings + Local Policies + User Rights Assignments

Task3000 Personal Server is used to execute the submitted task **on the same machine** and **current user**.

Task3000 Server can execute the task on the same or different server, however logged user **must have administrator user rights**.

Basic job of the Task3000 Server is to execute the prepared tasks according to the parameters given and maintaining the Task Queue.

Task3000 Server hasn't own remote administration and rely on the programs like SMS, VNC or similar to do the job. Windows NT administrative tools can do many administrative jobs about security, and remote administration is necessary only in cases when administrator would like to reply to task messages, which is not recommended practice. Task3000 Server should normally run only such application, which does not need human intervention.

Under such conditions every program can be executed as usually. Designers should test the tasks to such extent that they are sure that no human intervention is needed. If designed properly every task execution is documented in the appropriate log files on the server, for normal and abnormal execution of the task, and can be analyzed at any time.

Standard is recommended on that how to use network drives. If UNC (example: \\Server\MyDir) is used, there is no problem, but usage of the NET USE Drive_letter command could lead to the situation where several tasks can use the one Drive_letter as a different drive. Defining the standard for such situation is highly recommended. The best practice could be to define Drive_Letter for permanent connection to network drives.

Note: usage of too much network drives simultaneously is not recommended and task should release all network drives at the end of the run (normally or abnormally).

Recommended is also standard for network printers, which are used to print the reports.

Task3000 Server Administrator Note: By deleting the **Task3000.ctf** files anywhere in the \$JQ directory system will reassign the full rights to administrator. Depend on the directory Task3000 Users group is assign RX,X(Directory,File) rights or specific user RXD,X rights on the User_Name directory and Full access to %T3T% and %T3S% directory (reset to default)

Execution of the Tasks

Task3000 Server run as an application. In some cases it could be possible to run as service (tested with SRVANY from Windows NT Resource Kit), however not all applications could run on that way. Recommended practice is to run only as application and NT User. Auto Logon feature enable the server to logon the User with Administrator rights to logon automatically.

Task3000 Server executes the task step by step. Each step is totally separated from previous step, because it runs as a separate process. If data should be passed from one step to another, the data should be written to the file. Files can typically be stored in Task File Directory, Task Step Directory or some other server (see Task Queue later). File RC mechanism is used to search the files for the text of our interest and setting the Return Code, which can be used to decide whether some future step will be executed or not. If the program is able to write to the console and console is not cleared, this information is kept under Task_Step_Directory\Step_Number.TXT. File can be searched in the subsequent steps with FileRC mechanism to set the RC. Only few programs for DOS are written on such way that can use %errorlevel% variable. In such cases it is necessary to write Command procedure (CMD or BAT), run the program within the procedure and write %errorlevel% to the file with:

```
Program
ECHO %errorlevel% >%T3S%\File1
```

Program is desired program and File1 is the result file written to the Task_Step_Directory and can be searched in the same or next step by FileRC.

Task Queue

Task Queue serves as a storage media for the tasks. It is actually directory structure in the form:

Drive_Letter:\\$JQ\Domain\User\Task_Name\Task_Number

Example:

C:\\$JQ\UserDomain\TaskUser\Demo\00000033

is the case where user TaskUser from NT domain UserDomain has task with the name Demo and task number is 33.

There are number of possibilities about first part. It is dependant on the operating system in use. Task3000 server can run:

1. As Standalone server with full file security and user security
2. Server, which is part of the Windows NT 4.0 domain
3. Server, which is part of the Active Directory (Windows 2000/2003)

Domain can be **\$LOCALS\$** for standalone servers or Domain Name for NT 4.0 or **\$AD\$** or ADS Name for Windows 2000/2003.

Also **User** can be simple **User Name** or User@ADS. Personal Server User is always **\$LOCALUSERS**.

When user submits the task, directory structure is automatically adapted to contain new data for the user. Usually user has the access only to his part of the tree (only with Task3000 Server Professional

Windows NT4.0 or higher), and to easy access the files user can use %T3T% and %T3S% variable. **%T3T% (Task File Directory)** is defined as

Drive_Letter:\\$JQ\Domain\User\Task_Name

And **%T3S% (Step File Directory)** as

Drive_Letter:\\$JQ\Domain\User\Task_Name\Task_Number

Note: backslash at the end are included as shown. Please see the whole set of %T3% variables in the appropriate chapter.

Example:

User want to write file MyFile to Step File Directory:

Copy c:\Myfile %T3S%Myfile

Task_Number is assigned **automatically** by the Task3000 Server and is different for each repetition of the task

Task3000 automatically assign **full access right** to **Task File Directory** and down to **User** and **Administrator**. However, all other security administration should be done with appropriate Windows NT tools (security is very limited on Windows 98, there is no file security). Also **disk quotas** are limited to **Windows XP/2000/2003**.

Task3000 can **delete Step File Directory** directories and files, which **exceeds desired number**. This eliminates the need to clean the queue manually.

Files are copied to the Task File Directory and should be cleaned by user. When task is submitted user is asked to confirm the replacement of the files. The idea is that Task_Name is project name and project data are saved on the Task3000 server, where backup of the data is performed regularly. Data in the Step File Directory are cleaned to contain only desired number of data from few latest run. These data should be archived to permanent media if necessary. Backup and Archiving can be part of the tasks, can be done manually or automatically scheduled on the daily basis, or performed as the dependent tasks which is repeated automatically if the tasks with the given name is executed.

User Parameters (Environment Variables)

In order to perform the job task step should usually receive the parameters. Setting the Environment Variables before task step execution does this. Environment Variables are easy visible within Command Procedures (CMD or BAT) with SET command. Every program, which can read Environment Variables, can accept the parameters like text processor, spreadsheet, database etc. User should read the documentation about that in the appropriate help file for the MS operating system and application program. Included is the example, which can run on the MS Windows with MS Office installed.

Conclusions

From the explanation before, it is possible to see that development of fully automated tasks is complicated and the knowledge about MS operating systems is essential. However, the **user** can do the job with a **click of the button** and following the instruction prepared by **Task3000 designer**. This is because the Task3000 system is very opened and the desire of the authors is that every application, which can run automatically, can be connected with other such application in a way that performs user job.

Running the job as batch tasks have great advantages, because it forces the standardization of working procedures. Less skill workers can perform the job, and outsourced designers can be used during the design of the tasks. Task3000 is also very suitable for data entry procedures, which stores the data from workstations on the server, and further backup/archive of the data automatically, therefore reduce the human effort to do regular system maintenance.

It is at first hard to understand, why to implement such package. Maybe the simplest answer possible is to be able to connect the work of many different people together to work as a single unit with minimum necessary parameters. Ideally implemented Task3000 system has all the logic to perform the job **automatically without human intervention**.

More information on www.task3000.com or www.shareit.com.

Copyright by Task3000 Ltd.