Automating FTP with the CP 443-1 IT

Contents **Page**

Introduction 2

FTP Basics with the SIMATIC NET CP 443-1 IT 3

- CONFIGURATION 3
- FTP SERVICES 6

FTP Server with the SIMATIC NET CP 443-1 IT 9

- OVERVIEW 9
- CONFIGURATION 9
- FILE_DB.TXT 9
- USER DATA BLOCK 10
- SENDING DATA 11

FTP Client with Windows 2000 13

- OVERVIEW 13
- BATCH PROGRAMMING 14
- FTP COMMAND FILE 17
- TRANSFER LOG 18
- WINDOWS 2000 ‘START’ COMMAND 19
- WINDOWS 2000 ‘FTP’ COMMAND 21
- AVAILABLE FTP COMMANDS ON THE CP 443-1 IT 22
- WINDOWS 2000 TASK SCHEDULER 23

FTP Client with UNIX 29

- OVERVIEW 29
- USER CONFIGURATION FOR FTP 30
- SHELL PROGRAMMING 31
- FTP COMMAND FILE 33
- UNIX CRON DAEMON 34

References 35

Technical Support – the SIMATIC Hotline Worldwide 36

The content of this document has been inspected for accuracy and reliability at the time of publication. However, deviations and omissions cannot be precluded as relevant hardware, software, and reference documentation cannot be guaranteed with respect to consistency, accuracy, reliability, or otherwise. This document is provided at no charge as a suggested guide. Use of the information in this document is entirely at the users own risk and the user assumes full liability when utilizing the information contained in this document.

Introduction

Automating file transfers from the IT module to a PC can be accomplished with a minimum of programming. The basics of the file transfer protocol command (ftp) will be discussed, as well as details concerning the 443-1 IT as an ftp server. Ftp client services on a PC with Windows 2000 is then detailed, including how to schedule a task such as a batch file for automatic processing.

This document is presented strictly as a guide and is not guaranteed for accuracy or reliability. The examples contained herein have been verified at the time of publication for the equipment and versions utilized.

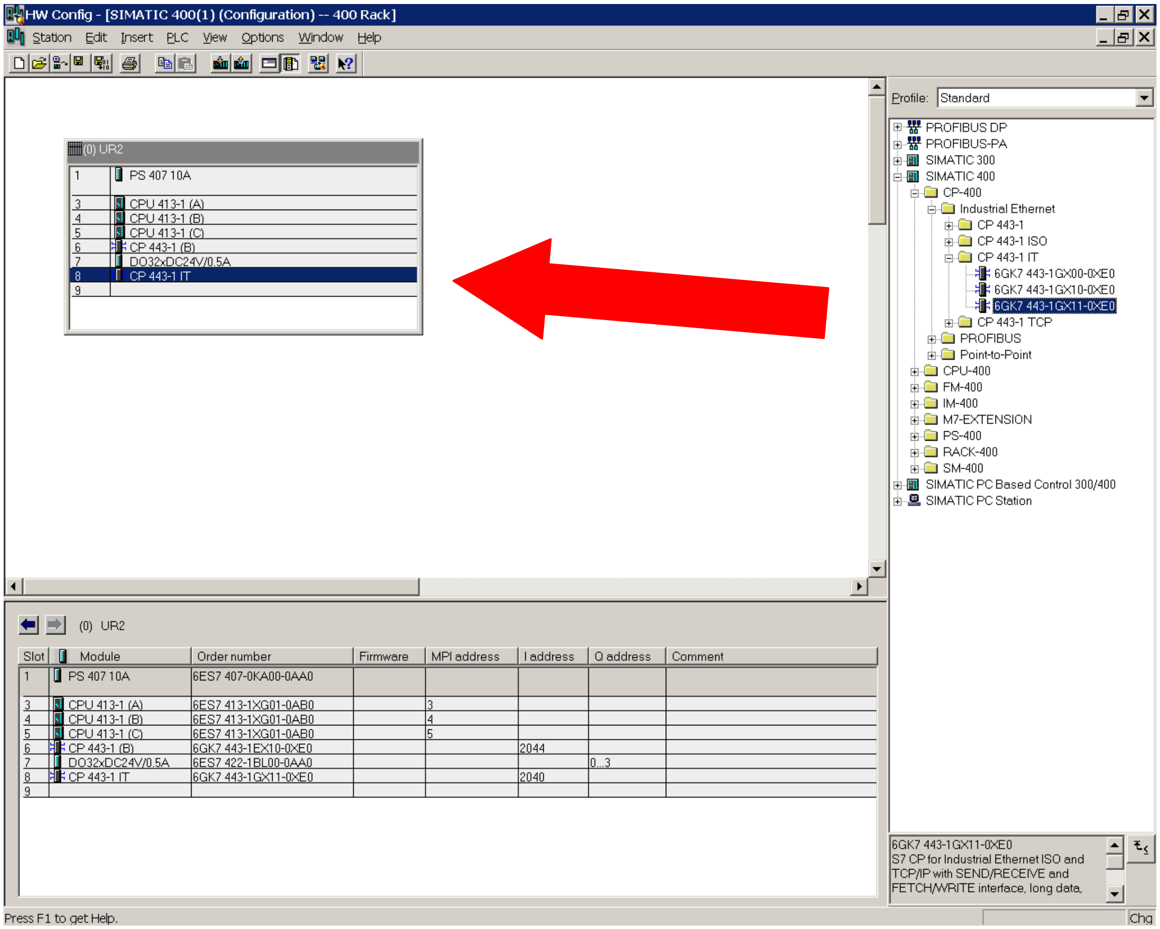
FTP Basics with the SIMATIC NET CP 443-1 IT

The CP 443-1 IT module is capable of providing FTP client and server functions. For this example, a PC with Windows 2000 (Service Pack 2) is the FTP client and the IT module is the FTP server. The CP version information is:

```
Part number          Hardware     Software
6GK7 443-1GX11-0XE0  1            V 1.1.0
```

CONFIGURATION

The IT module was configured in the following steps. First, the IT module was added to an S7-400 station and the Ethernet parameters assigned as follows.

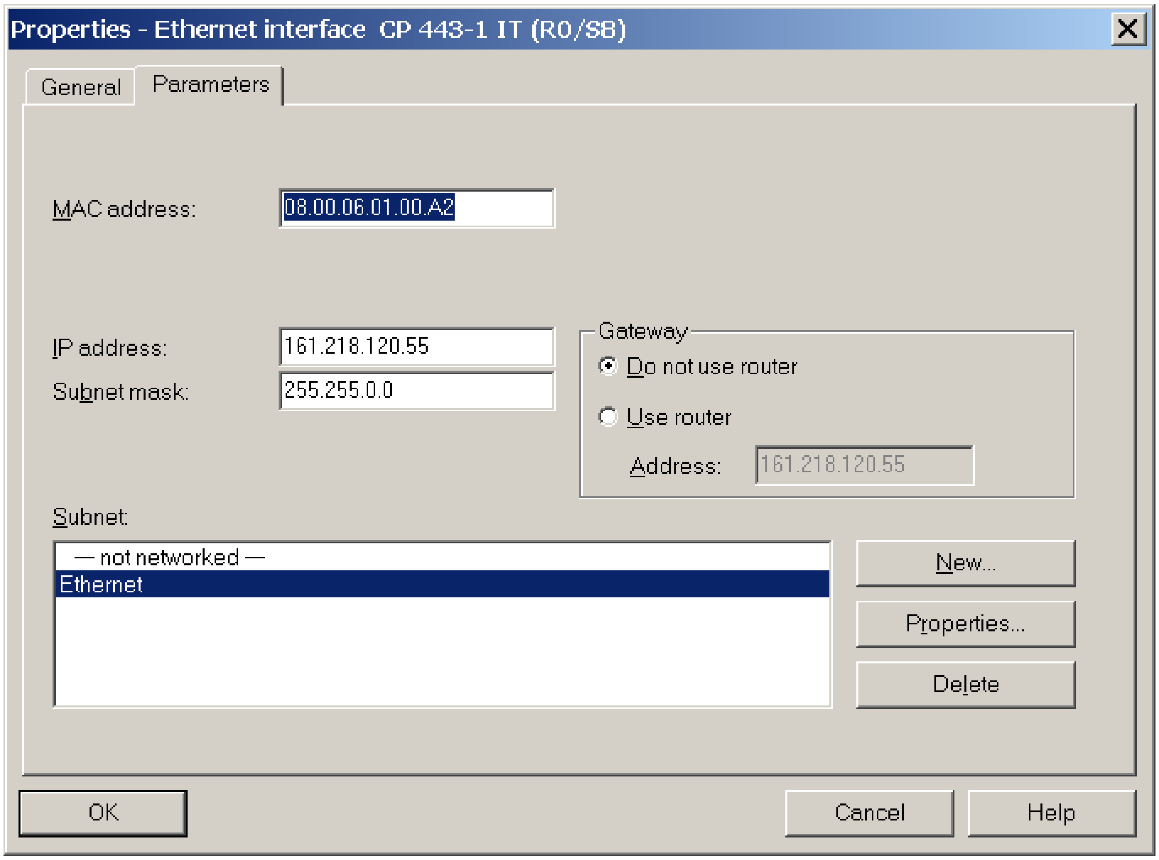


(0) UR2

Slot	Module	Order number	Firmware	MPI address	I address	Q address	Comment
1	PS 407 10A	6ES7 407-0KA00-0AA0					
3	CPU 413-1 (A)	6ES7 413-1XG01-0AB0		3			
4	CPU 413-1 (B)	6ES7 413-1XG01-0AB0		4			
5	CPU 413-1 (C)	6ES7 413-1XG01-0AB0		5			
6	CP 443-1 (B)	6GK7 443-1EX10-0XE0			2044		
7	DO32xDC24V/0.5A	6ES7 422-1BL00-0AA0				0...3	
8	CP 443-1 IT	6GK7 443-1GX11-0XE0			2040		
9							

6GK7 443-1GX11-0XE0
S7 CP for Industrial Ethernet ISO and TCP/IP with SEND/RECEIVE and FETCH/WRITE interface, long data,

After the IT module is inserted in the station, configure an Ethernet network and assign the IP address for the module. For this example, a router was not configured as the PC and the S7-400 station were on the same hub.



Properties - Ethernet interface CP 443-1 IT (R0/S8)

MAC address: 08.00.06.01.00.A2

IP address: 161.218.120.55

Subnet mask: 255.255.0.0

Gateway

● Do not use router

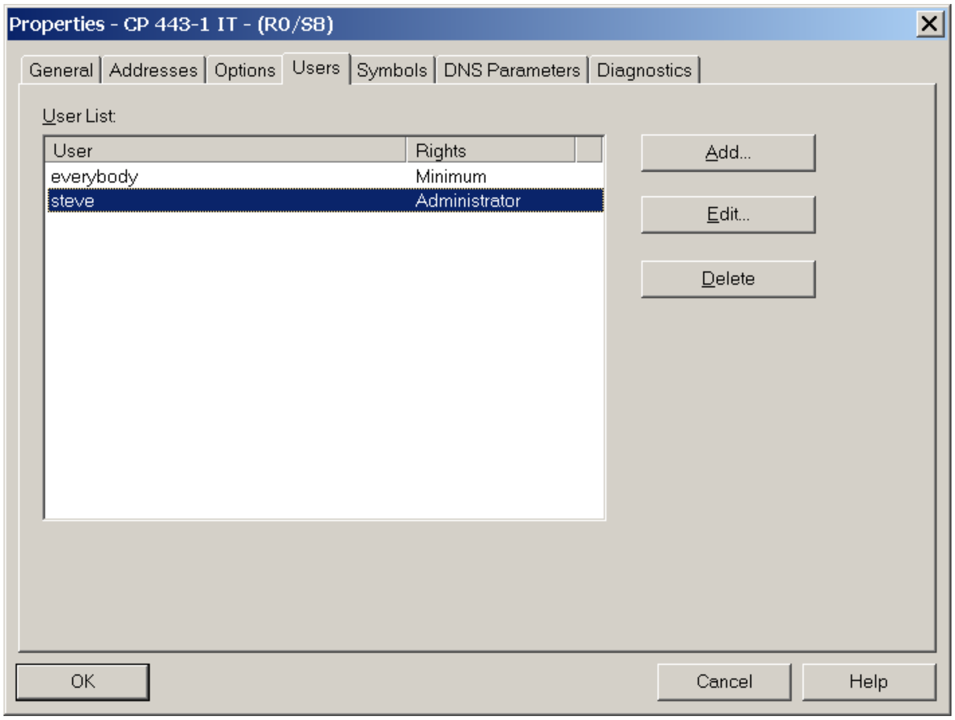
○ Use router

Address: 161.218.120.55

Subnet:

— not networked —
Ethernet

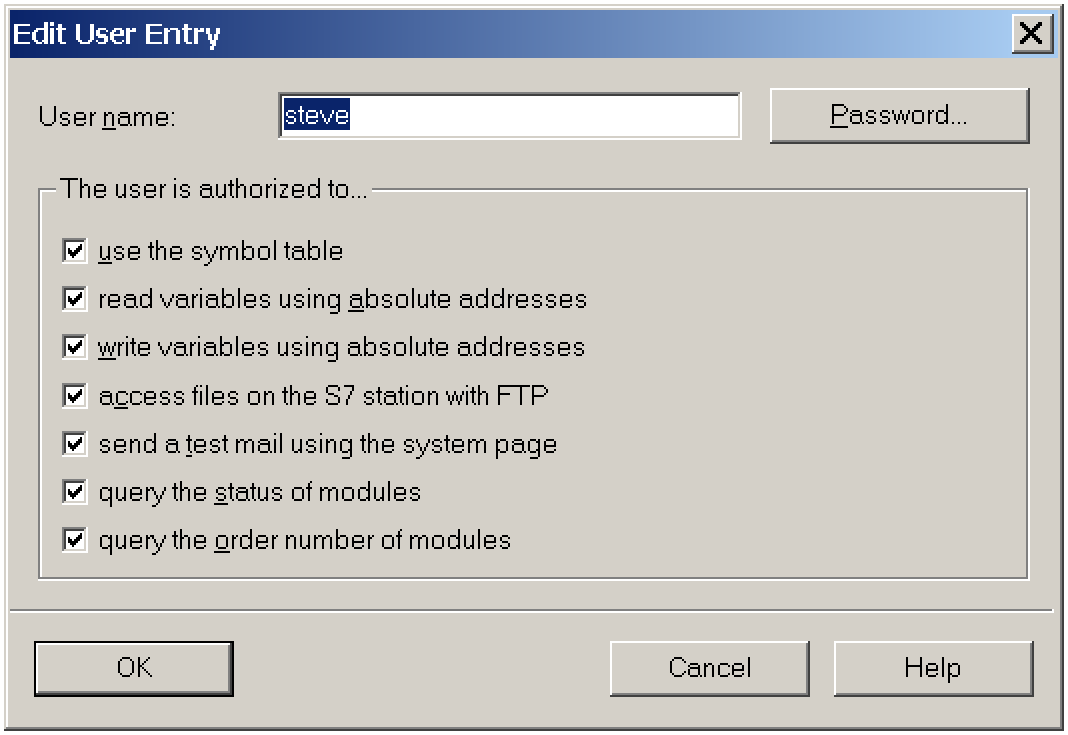
After the Ethernet parameters are assigned, open the object properties of the IT module and select the 'Users' tab. For this example, a new user named 'steve' was added, full access granted, and a password was assigned.



Properties - CP 443-1 IT - (R0/S8)

User List:

User	Rights
everybody	Minimum
steve	Administrator



Edit User Entry

User name: steve

The user is authorized to...

☑ use the symbol table

☑ read variables using absolute addresses

☑ write variables using absolute addresses

☑ access files on the S7 station with FTP

☑ send a test mail using the system page

☑ query the status of modules

☑ query the order number of modules

Once these steps are complete, the configuration is compiled and downloaded. After the download is complete, the S7-400 station power should be cycled.

FTP SERVICES

Once the IT module has been rebooted, open a DOS Command window. The following commands were executed in order to put a file from the PC (C:\test.txt) into the IT's user directory.

```
C:\>ftp 161.218.120.55
Connected to 161.218.120.55.
220 CP 443-1 IT FTP-Server V1.03 ready for new user
User (161.218.120.55:(none)): steve
331 User name okay, need password.
Password:
230 User logged in, proceed.
ftp> cd user
250 Requested file action okay, completed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
total 3
drw-rw-rw-     1  root root 0    JAN   01 00:04   .
drwxrwxrwx     1  root root 0    JAN   01 00:00   ..
-rw-rw-rw-     1  root root 5922   JAN   01 00:05   indexe.htm
226 Transfer ok. Closing data connection.
ftp: 181 bytes received in 0.10Seconds 1.81Kbytes/sec.
ftp> put test.txt
200 Command okay.
150 File status okay; about to open data connection.
226 Transfer ok. Closing data connection.
ftp: 17 bytes sent in 0.00Seconds 17000.00Kbytes/sec.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
total 4
drw-rw-rw-     1  root root 0    JAN   01 00:04   .
drwxrwxrwx     1  root root 0    JAN   01 00:00   ..
-rw-rw-rw-     1  root root 5922   JAN   01 00:05   indexe.htm
-rw-rw-rw-     1  root root 16   JAN   01 00:30   test.txt
226 Transfer ok. Closing data connection.
ftp: 242 bytes received in 0.10Seconds 2.42Kbytes/sec.
ftp> quit
221 Closing control connection; Thank you for using our FTP server.

C:\>
```

Each command will be discussed in further detail. Prompts will be displayed in bold type and data typed in by the user will be in italics.

```
C:\>ftp 161.218.120.55
Connected to 161.218.120.55.
220 CP 443-1 IT FTP-Server V1.03 ready for new user
User (161.218.120.55:(none)): steve
331 User name okay, need password.
Password:
230 User logged in, proceed.
ftp>
```

From the root directory of the PC, which is where the file 'test.txt' is located, the user executes the FTP command with the IP address of the IT module identifying the FTP server. The PC and the

IT module **MUST** be in the same IP-Subnet segment as defined by their respective net masks. In this example, the net masks on the PC and the IT module were 255.255.0.0 and both IP addresses were within the same class B segment, 161.218.x.y.

Once the connection is established, the IT module requires a user login. The account for ‘steve’ was set up above on the IT module, so the login name ‘steve’ is entered and, when prompted, the assigned password is entered. Once account verification is completed, the IT module provides an FTP prompt.

```
ftp> cd user
250 Requested file action okay, completed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
total 3
drw-rw-rw-      1  root root 0   JAN   01 00:04   .
drwxrwxrwx      1  root root 0   JAN   01 00:00   ..
-rw-rw-rw-      1  root root 5922   JAN   01 00:05   indexe.htm
226 Transfer ok. Closing data connection.
ftp: 181 bytes received in 0.10Seconds 1.81Kbytes/sec.
ftp>
```

In the above listing, the current directory (which is the root directory at login) is changed to the ‘user’ directory and the file listing is displayed. Notice that the file ‘test.txt’ does not appear in this directory listing.

```
ftp> put test.txt
200 Command okay.
150 File status okay; about to open data connection.
226 Transfer ok. Closing data connection.
ftp: 17 bytes sent in 0.00Seconds 17000.00Kbytes/sec.
ftp>
```

As shown above, the ‘put’ command is executed which will search in the PC’s current directory (root) for the file ‘test.txt’ and copy it to the IT’s current directory (/user).

```
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
total 4
drw-rw-rw-      1  root root 0   JAN   01 00:04   .
drwxrwxrwx      1  root root 0   JAN   01 00:00   ..
-rw-rw-rw-      1  root root 5922   JAN   01 00:05   indexe.htm
-rw-rw-rw-      1  root root 16   JAN   01 00:30   test.txt
226 Transfer ok. Closing data connection.
ftp: 242 bytes received in 0.10Seconds 2.42Kbytes/sec.
ftp>
```

In the above listing, the directory is displayed and the file ‘test.txt’ is shown to be in the /user directory.

```
ftp> quit
221 Closing control connection; Thank you for using our FTP server.

C:\>
```

As shown above, the command ‘quit’ is used to terminate the FTP session. The DOS command prompt is then restored.

FTP Server with the SIMATIC NET CP 443-1 IT

The CP 443-1 IT module is capable of providing FTP client and server functions. For this example, a PC with Windows 2000 (Service Pack 2) is the FTP client and the IT module is the FTP server. A file will be generated by the S7-400 CPU and retrieved on the PC. The CP version information is:

Part number	**Hardware**	**Software**
6GK7 443-1GX11-0XE0	1	V 1.1.0

OVERVIEW

In order to generate a file from the S7-400 CPU and retrieve it via FTP on a PC, the following procedures will be detailed:

Configuration and parameterization of the S7-400 station
Modification of the file_db.txt configuration file on the CP IT
Creation of a user DB on the S7-400 CPU
Sending data from the CPU to the PC

CONFIGURATION

Configure the IT module in the S7-400 station and assign an IP address and subnet mask that allows the CP to communicate with the PC. Configure a user on the IT module with administrative rights (full access). Once these parameters have been assigned, download the hardware configuration to the S7-400 station.

FILE_DB.TXT

On the CP IT, in the /config folder, there is a file named ‘file_db.txt’. An example of this file is listed below. For this example, the CPU in the S7-400 station is identified as ‘cpu1’ and is located in slot 3 of the central rack (rack 0). Data block DB 200 will be used as the communication DB to bring data out of the CPU into the CP IT, and then use FTP to retrieve this file into the PC.

```
# CONFIGURATION FILE for file transfer between an FTP client of a remote system
# and an S7-CPU using the FTP server of the IT-CP

# This is an ASCII file and may be edited.
# This file must be located in the directory "/config" of the file system
# of the IT-CP. Its file name must be "file_db.txt" (all lowercase).

# All lines that do not begin with "cpu" (lowercase AND no leading blanks)
# are interpreted as comment.
# Maximum length per line is 256 characters.
# Delimiters are (one or more) blanks or tabs.

# The following table defines the rack and slot of the CPU(s).
# Definitions of "cpu1", "cpu2", "cpu3" and "cpu4" are allowed.

# CPU      Rack      Slot
# ----------------------------
cpu1        0         3

# The following table defines pairs of file names and file DBs in the CPU.
# The maximum number of pairs is 100.
# The file name must begin with "cpuX" (where X = 1, 2, 3 or 4).
# Note that "cpuX" must be defined in the table above!
# The file name must consist of the characters "a-z", "A-Z", "0-9", "_" or "."
# It must not include a path. The maximum length of a file name is 64 characters.

# File Name            File DB Number
# --------------------------------------
cpu1db200              200
```

USER DATA BLOCK

A user DB must be configured with an FTP file system header preceding the data. In this example, DB 200 is configured with the FTP header and 200 bytes of data. The initial values for the header are also included in a separate listing.

- STEP1 : Copy the FTP header UDT into your project
  The FTP header UDT is located in the ‘SIMATIC_NET_CP’ library under the ‘S7-400’ folder. It is UDT 1 “FILE_DB_HEADER”.

- STEP2 : Create user DB
  Data block DB 200 is created for this example as a shared data block and the following declarations are entered:

Name	**Type**	**Comment**
HDR	UDT 1	FTP file system header
DATA	ARRAY [1..200] BYTE	User data (200 bytes)

- STEP 3: Initialize FTP header
  Change view of user-DB from declaration view to data view and make the following assignments to the referenced parameters:

NAME	**VALUE**
EXIST	TRUE
WRITEACCESS	TRUE
ACT_LENGTH	L#200
MAX_LENGTH	L#200

Once the user-DB has been created and initialized, download it to the CPU.

SENDING DATA

Now that the configuration file '/config/file_db.txt' has been modified for the CP IT and a user-DB has been created and downloaded for the S7-400 CPU, data exchange is now possible. A variable table (VAT) is created in STEP7 with the first 20 bytes of the data array in DB 200 as shown below.

Address	**Symbol**	**Display Format**	**Status value**	**Modify value**
DB200.DBB 20	FTP DATA.DATA[1]	CHARACTER	'T'	'T'
DB200.DBB 21	FTP DATA.DATA[2]	CHARACTER	'h'	'h'
DB200.DBB 22	FTP DATA.DATA[3]	CHARACTER	'i'	'i'
DB200.DBB 23	FTP DATA.DATA[4]	CHARACTER	's'	's'
DB200.DBB 24	FTP DATA.DATA[5]	CHARACTER	' '	' '
DB200.DBB 25	FTP DATA.DATA[6]	CHARACTER	'i'	'i'
DB200.DBB 26	FTP DATA.DATA[7]	CHARACTER	's'	's'
DB200.DBB 27	FTP DATA.DATA[8]	CHARACTER	' '	' '
DB200.DBB 28	FTP DATA.DATA[9]	CHARACTER	'a'	'a'
DB200.DBB 29	FTP DATA.DATA[10]	CHARACTER	' '	' '
DB200.DBB 30	FTP DATA.DATA[11]	CHARACTER	't'	't'
DB200.DBB 31	FTP DATA.DATA[12]	CHARACTER	'e'	'e'
DB200.DBB 32	FTP DATA.DATA[13]	CHARACTER	's'	's'
DB200.DBB 33	FTP DATA.DATA[14]	CHARACTER	't'	't'
DB200.DBB 34	FTP DATA.DATA[15]	CHARACTER	'.'	'.'
DB200.DBB 35	FTP DATA.DATA[16]	CHARACTER	B#16#00	
DB200.DBB 36	FTP DATA.DATA[17]	CHARACTER	B#16#00	
DB200.DBB 37	FTP DATA.DATA[18]	CHARACTER	B#16#00	
DB200.DBB 38	FTP DATA.DATA[19]	CHARACTER	B#16#00	
DB200.DBB 39	FTP DATA.DATA[20]	CHARACTER	B#16#00	

The ASCII values in the **Modify value** column were transmitted to the data block. An FTP connection was opened on the PC to connect to the CP IT as shown below and the file was retrieved to the root folder on the local drive of the PC (C:\). Once this file was retrieved, the FTP session was closed and the file was displayed using the DOS 'type' command.

```
C:\>ftp 161.218.120.55
Connected to 161.218.120.55.
220 CP 443-1 IT FTP-Server V1.03 ready for new user
User (161.218.120.55:(none)): example
331 User name okay, need password.
Password:
230 User logged in, proceed.
ftp> cd /cpu1
250 Requested file action okay, completed.
ftp> dir
200 Command okay.
150 File status okay; about to open data connection.
-rw-rw-rw-      1  root root 200   JAN   01 00:00   cpu1db200  FileDB 200
226 Transfer ok. Closing data connection.
ftp: 75 bytes received in 0.12Seconds 0.62Kbytes/sec.
ftp> get cpu1db200
200 Command okay.
150 File status okay; about to open data connection.
226 Transfer ok. Closing data connection.
ftp: 200 bytes received in 0.03Seconds 6.67Kbytes/sec.
ftp> quit
221 Closing control connection; Thank you for using our FTP server.

C:\>type cpu1db200
This is a test.
```

FTP Client with Windows 2000

For this example, a PC with Windows 2000 service pack 2 was utilized to retrieve a file from the CP 443-1 IT.

OVERVIEW

In this section, a batch file will be presented which can be run with the Windows 2000 Task Scheduler to automatically retrieve, via ftp, a file from the CP 443-1 IT module. The batch file creates a text file of ftp commands and a log of the file transfer. Abbreviated information is presented with regards to ftp and the Windows “start” command. This section is concluded with step-by-step instructions on how to schedule this batch file for automatic execution on a Windows 2000 platform.

BATCH PROGRAMMING

The listing below is a batch file that was executed on the PC to retrieve a CPU file on the IT module. It is named ‘auto_ftp.bat’ and is stored in the folder ‘c:\files\SRPT’. The batch file is divided into four steps, each of which is explained in further detail.

```
::============================================================
::  FTP File Transfer from IT Server to local PC
::============================================================
::  This batch file obtains a file from a Siemens CP 443-1 IT
::  module using FTP.  The PC from which this is launched is
::  the client and the IT module is the server.  The IT mod-
::  ule is a 6GK7 443-1GX11-0XE0 (HW v1, FW v1.1.0) and the
::  PC is Windows2000 SP2.  For details on the "START" and
::  "FTP" commands, refer to the online help in Windows2000.
::============================================================

::------------------------------------------------------------
::  STEP 1 : Create a text file containing the FTP commands.
::------------------------------------------------------------

echo steve>  ftpscript.dat
echo password>> ftpscript.dat
echo cd /cpu1                           >> ftpscript.dat

:: Remove "::" on next next line if file is binary
:: echo binary                          >> ftpscript.dat

echo lcd c:\files\SRPT                  >> ftpscript.dat
echo get cpu1db200                      >> ftpscript.dat
echo quit                               >> ftpscript.dat

::------------------------------------------------------------
::  STEP 2 : Prepare transfer log file header.
::------------------------------------------------------------

echo %date% %time% Transferring file     >  transfer.log

::------------------------------------------------------------
::  STEP 3 : Create a command shell to execute the FTP file
::           retrieval.  Wait for FTP to finish before
::           closing shell.
::------------------------------------------------------------

START /WAIT FTP -v -s:ftpscript.dat 161.218.120.55

::------------------------------------------------------------
::  STEP 4 : Close transfer log file and delete text file
::           containing the FTP commands (optional).
::------------------------------------------------------------

echo %date% %time% Transfer complete     >> transfer.log

:: DEL ftpscript.dat

:END
```

```
::------------------------------------------------------------
::  STEP 1 : Create a text file containing the FTP commands.
::------------------------------------------------------------

echo steve>  ftpscript.dat
echo password>> ftpscript.dat
echo cd /cpu1                            >> ftpscript.dat

:: Remove "::" on next next line if file is binary
:: echo binary                           >> ftpscript.dat

echo lcd c:\files\SRPT                   >> ftpscript.dat
echo get cpu1db200                       >> ftpscript.dat
echo quit                                >> ftpscript.dat
```

- STEP 1: Create a text file containing the FTP commands.

  The ‘echo’ command is used to generate an FTP script file (ftpscript.dat) as the text is redirected into this file.

  ```
  echo steve> ftpscript.dat
  ```

  With this command, the data file ‘ftpscript.dat’ is created with the single redirect ‘>’ and the ASCII characters ‘steve’ are added. This is the login account which exists on the IT module for this example.

  ```
  echo password>> ftpscript.dat
  ```

  With this command, the data file ‘ftpscript.dat’ is appended with the double redirect ‘>>’ and the ASCII characters ‘password’ are added on a new line below the existing ‘steve’ characters. This is the password for the account ‘steve’.

  Subsequent echo commands in this step continue to append to the script file. The completed file is listed subsequently and the commands are explained further.

```
::------------------------------------------------------------
::  STEP 2 : Prepare transfer log file header.
::------------------------------------------------------------

echo %date% %time% Transferring file     >  transfer.log
```

- STEP 2: Prepare transfer log file header.

  The ‘echo’ command is used to create a transfer log file and stamp it with the date and time before the ftp file transfer occurs.

```
::------------------------------------------------------------
::  STEP 3 : Create a command shell to execute the FTP file
::           retrieval.  Wait for FTP to finish before
::           closing shell.
::------------------------------------------------------------

START /WAIT FTP -v -s:ftpscript.dat 161.218.120.55
```

- STEP 3: Create a command shell and execute the FTP file transfer.

  The Windows ‘start’ command is executed to begin a new instance of the command prompt window. It is conditionally spawned, having to wait (‘/wait’) for the subsequent ‘ftp’ command to execute before closing the spawned command prompt window. The ftp portion of this command line is:

  ```
  FTP –v –s:ftpscript.dat 161.218.120.55
  ```

  The conditions on the ftp command include disabling verbose server responses (-v) and utilization of a text file containing ftp commands (-s:ftpscript.dat). The ftp server is specified by an IP address (161.218.120.55) and must be on the same subnet as the PC.

```
::------------------------------------------------------------
::  STEP 4 : Close transfer log file and delete text file
::           containing the FTP commands (optional).
::------------------------------------------------------------

echo %date% %time% Transfer complete     >> transfer.log

:: DEL ftpscript.dat

:END
```

- STEP 4 : Close the log file and delete text file of ftp commands.

  The log file is appended with a date and time stamp as well as notification of completion. The user has the choice of deleting the text file of ftp commands (ftpscript.dat). The last line in the file is an end label (`:END`).

FTP COMMAND FILE

The listing below is of the ftp text command file generated by step 1 above.

```
steve
password
cd /cpu1
lcd c:\files\SRPT
get cpu1db200
quit
```

Each line of this file is explained below.

`steve`	login for the user account configured on the IT module
`password`	password for the user account ‘steve’
`cd /cpu1`	change the current directory on the IT module to /cpu1
`lcd c:\files\SRPT`	change the current directory on the PC to c:\files\srpt
`get cpu1db200`	get the file cpu1db200 from the IT module and store it on the PC
`quit`	terminate the ftp session and exit

TRANSFER LOG

Below is a listing of a transfer log file (transfer.log) generated by steps 2 and 4 of the ftp batch file (auto_ftp.bat). It stamps before and after the file transfer occurs as a record of the transaction.

```
Thu 03/21/2002 15:35:01.92 Transferring file
Thu 03/21/2002 15:35:03.36 Transfer complete
```

WINDOWS 2000 ‘START’ COMMAND

START Starts a separate window to run a specified program or command.

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

    "title"     Title to display in  window title bar.
    path        Starting directory
    B           Start application without creating a new window. The
                application has ^C handling ignored. Unless the application
                enables ^C processing, ^Break is the only way to interrupt
                the application
    I           The new environment will be the original environment passed
                to the cmd.exe and not the current environment.
    MIN         Start window minimized
    MAX         Start window maximized
    SEPARATE    Start 16-bit Windows program in separate memory space
    SHARED      Start 16-bit Windows program in shared memory space
    LOW         Start application in the IDLE priority class
    NORMAL      Start application in the NORMAL priority class
    HIGH        Start application in the HIGH priority class
    REALTIME    Start application in the REALTIME priority class
    ABOVENORMAL Start application in the ABOVENORMAL priority class
    BELOWNORMAL Start application in the BELOWNORMAL priority class
    WAIT        Start application and wait for it to terminate
    command/program
                If it is an internal cmd command or a batch file then
                the command processor is run with the /K switch to cmd.exe.
                This means that the window will remain after the command
                has been run.

                If it is not an internal cmd command or batch file then
                it is a program and will run as either a windowed application
                or a console application.

    parameters  These are the parameters passed to the command/program
```

If Command Extensions are enabled, external command invocation through the command line or the START command changes as follows:

non-executable files may be invoked through their file association just by typing the name of the file as a command. (e.g. WORD.DOC would launch the application associated with the .DOC file extension). See the ASSOC and FTYPE commands for how to create these associations from within a command script.

When executing an application that is a 32-bit GUI application, CMD.EXE does not wait for the application to terminate before returning to the command prompt. This new behavior does NOT occur if executing within a command script.

When executing a command line whose first token is the string "CMD” without an extension or path qualifier, then "CMD" is replaced with the value of the COMSPEC variable. This prevents picking up CMD.EXE from the current directory.

When executing a command line whose first token does NOT contain an extension, then CMD.EXE uses the value of the PATHEXT environment variable to determine which extensions to look for and in what order. The default value for the PATHEXT variable is:

```
        .COM;.EXE;.BAT;.CMD
```

Notice the syntax is the same as the PATH variable, with semicolons separating the different elements.

When searching for an executable, if there is no match on any extension, then looks to see if the name matches a directory name. If it does, the START command launches the Explorer on that path. If done from the command line, it is the equivalent to doing a CD /D to that path.

WINDOWS 2000 'FTP' COMMAND

The 'FTP' command transfers files to and from a computer running an FTP server service (sometimes called a daemon). Ftp can be used interactively. Click ftp commands in the Related Topics list for a description of available ftp subcommands. This command is available only if the TCP/IP protocol has been installed. Ftp is a service, that, once started, creates a sub-environment in which you can use ftp commands, and from which you can return to the Windows 2000 command prompt by typing the quit subcommand. When the ftp sub-environment is running, it is indicated by the ftp command prompt.

```
ftp [-v] [-n] [-i] [-d] [-g] [-s:filename] [-a] [-w:windowsize] [computer]
```

Parameters

-v Suppresses display of remote server responses.

-n Suppresses autologin upon initial connection.

-I Turns off interactive prompting during multiple file transfers.

-d Enables debugging, displaying all ftp commands passed between the client and server.

-g Disables file name globbing, which permits the use of wildcard characters (* and ?) in local file and path names. (See the glob command in the online Command Reference.)

-s:*filename* Specifies a text file containing ftp commands; the commands automatically run after ftp starts. No spaces are allowed in this parameter. Use this switch instead of redirection (>).

-a Use any local interface when binding data connection.

-w:*windowsize* Overrides the default transfer buffer size of 4096.

computer Specifies the computer name or IP address of the remote computer to connect to. The computer, if specified, must be the last parameter on the line.

AVAILABLE FTP COMMANDS ON THE FTP-SHELL

```
!             delete        literal       prompt        send
?             debug         ls            put           status
append        dir           mdelete       pwd           trace
ascii         disconnect    mdir          quit          type
bell          get           mget          quote         user
binary        glob          mkdir         recv          verbose
bye           hash          mls           remotehelp
cd            help          mput          rename
close         lcd           open          rmdir
```

!	Escape to the shell
?	Print local help information
append	Append to a file
ascii	Set ascii transfer type
bell	Beep when command completed
binary	Set binary transfer type
bye	Terminate ftp session and exit
cd	Change remote working directory
close	Terminate ftp session
delete	Delete remote file
debug	Toggle debugging mode
dir	List contents of remote directory
disconnect	Terminate ftp session
get	Receive file
glob	Toggle metacharacter expansion of local file names
hash	Toggle printing `#' for each buffer transferred
help	Print local help information
lcd	Change local working directory
literal	Send arbitrary ftp command
ls	List contents of remote directory
mdelete	Delete multiple files
mdir	List contents of multiple remote directories
mget	Get multiple files
mkdir	Make directory on the remote machine
mls	List contents of multiple remote directories
mput	Send multiple files
open	Connect to remote ftp
prompt	Force interactive prompting on multiple commands
put	Send one file
pwd	Print working directory on remote machine
quit	Terminate ftp session and exit
quote	Send arbitrary ftp command
recv	Receive file
remotehelp	Get help from remote server
rename	Rename file
rmdir	Remove directory on the remote machine
send	Send one file
status	Show current status
trace	Toggle packet tracing
type	Set file transfer type
user	Send new user information
verbose	Toggle verbose mode

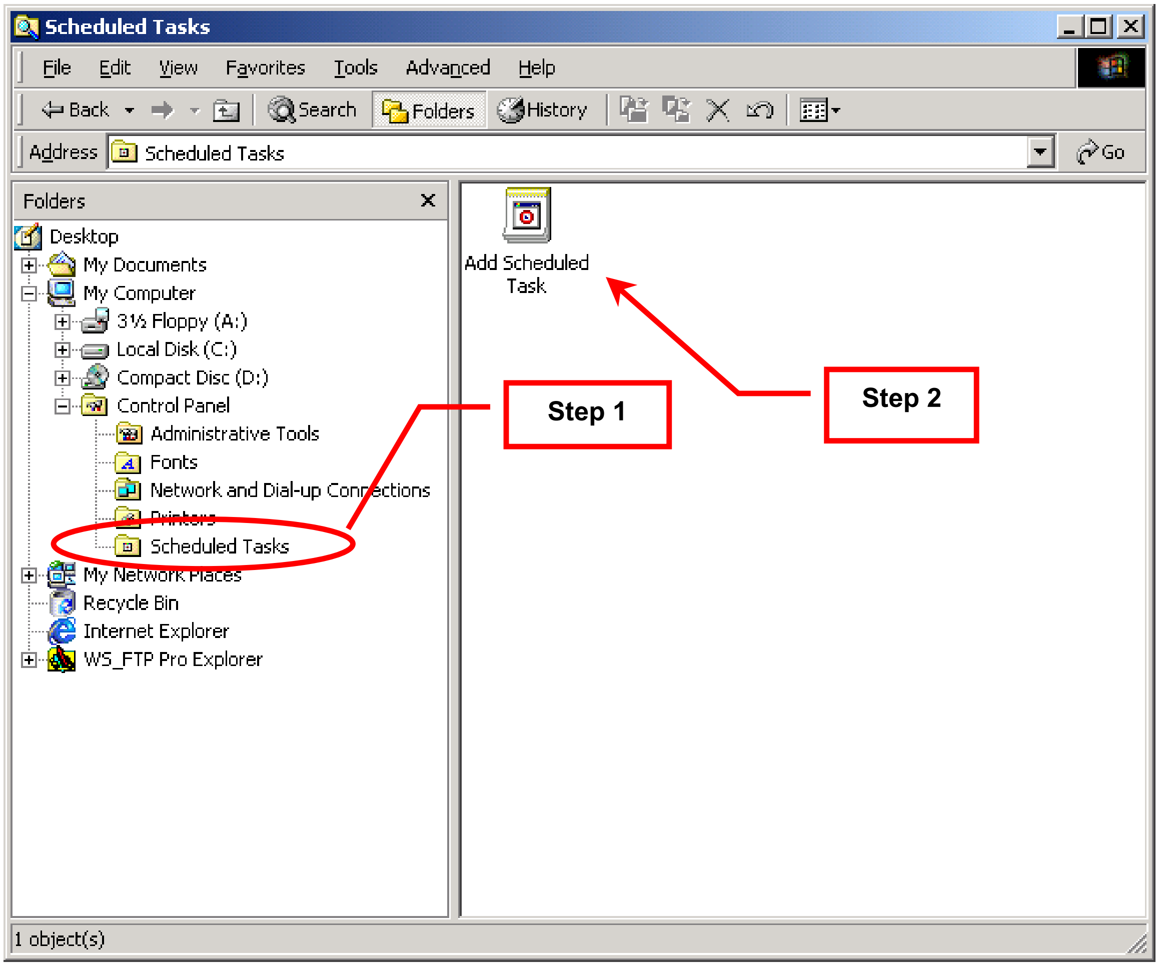
WINDOWS 2000 TASK SCHEDULER

In order to automate the process of retrieving the file ‘cpu1db200’ by the batch file ‘auto_ftp.bat’, the following steps should be implemented on Windows 2000.

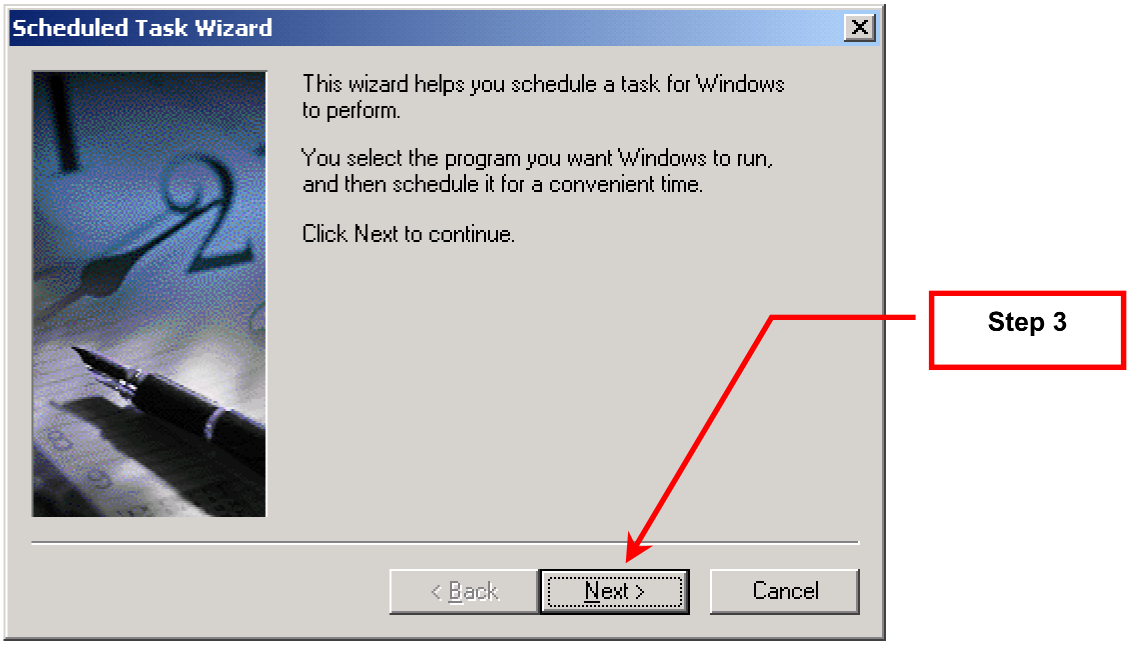
1. Open the Windows Explorer and browse to the folder:

```
\My Computer\Control Panel\Scheduled Tasks
```

2. Open the icon “Add Scheduled Task”.



3. When the Scheduled Task Wizard appears, click the “Next” button to continue.

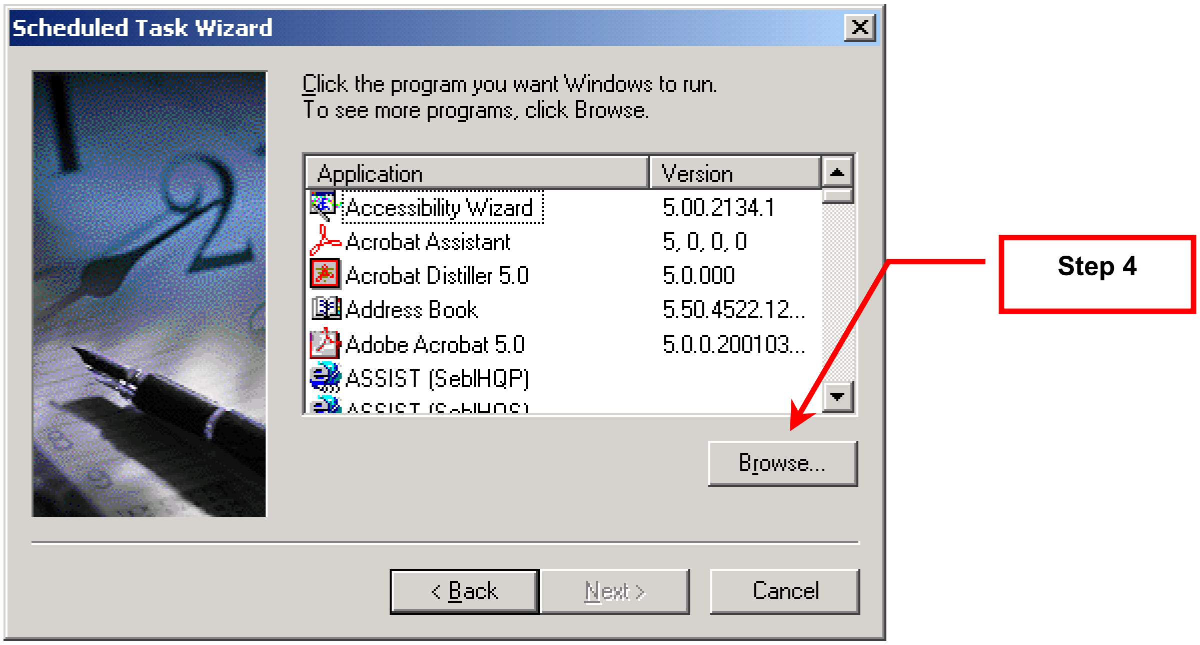


This wizard helps you schedule a task for Windows to perform.

You select the program you want Windows to run, and then schedule it for a convenient time.

Click Next to continue.

4. The program selection window will appear; select the “Browse” button.



Click the program you want Windows to run.
To see more programs, click Browse.

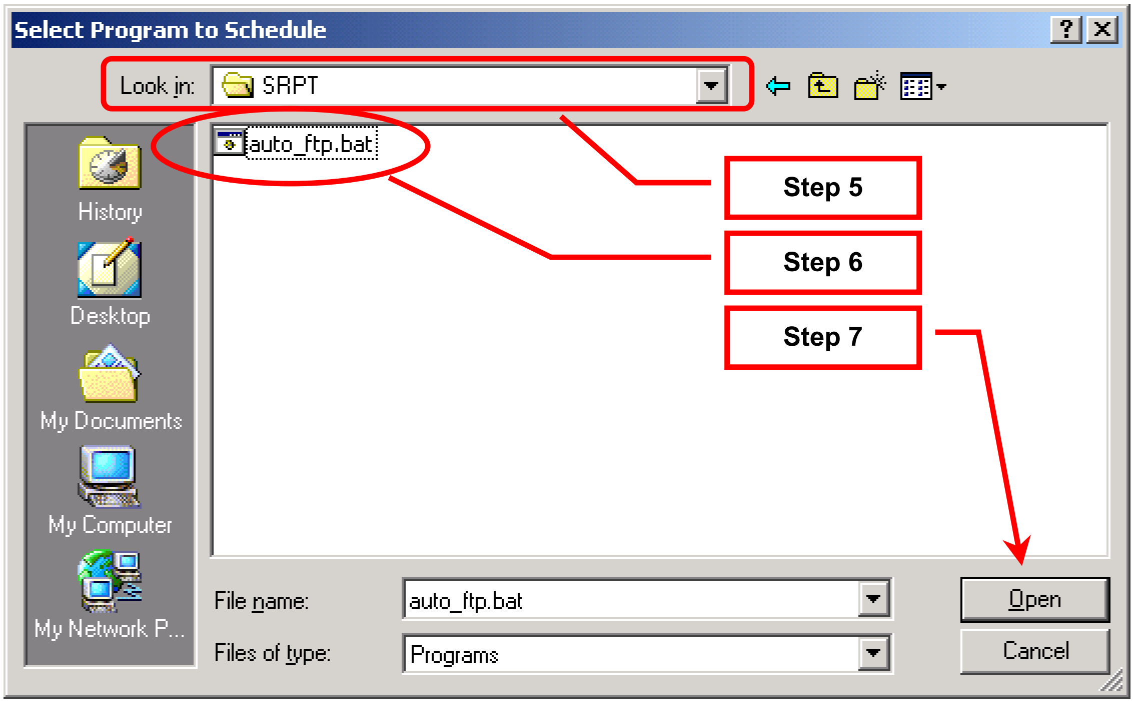
Application	Version
Accessibility Wizard	5.00.2134.1
Acrobat Assistant	5, 0, 0, 0
Acrobat Distiller 5.0	5.0.000
Address Book	5.50.4522.12...
Adobe Acrobat 5.0	5.0.0.200103...
ASSIST (SeblHQP)	
ASSIST (SeblHQS)	

5. Browse to the folder with the ftp batch file in the “Look in:” window. For this example, it is:

```
        c:\files\SRPT
```

6. Highlight the ftp batch file.

7. Press the “Open” button.



Select Program to Schedule

Look in: SRPT

auto_ftp.bat

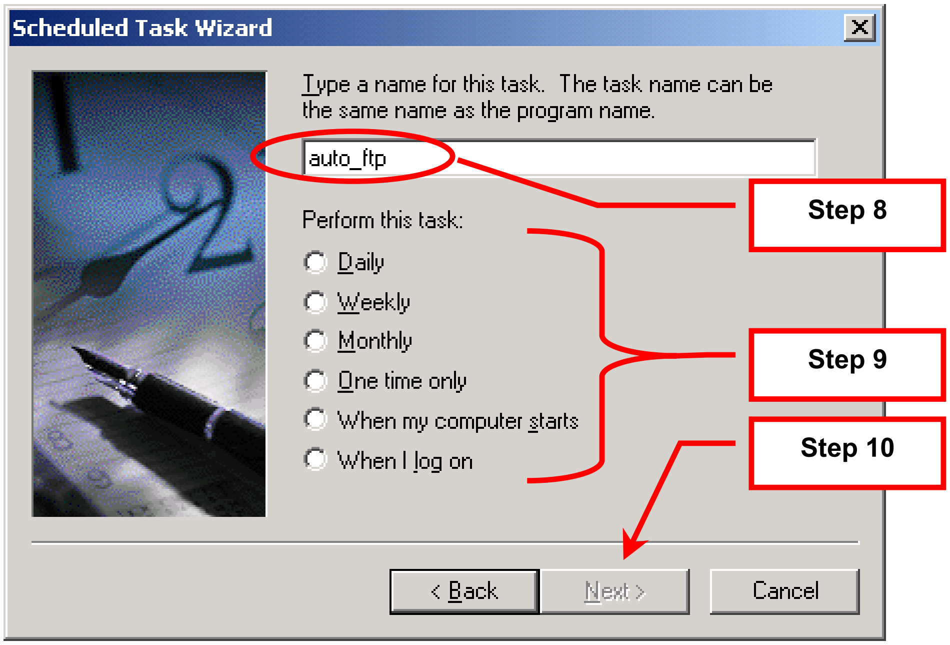
File name: auto_ftp.bat

Files of type: Programs

8. Enter a name for the scheduled task.

9. Select the frequency of running ‘auto_ftp.bat’.

10. Press the “Next” button after this information is entered.



Scheduled Task Wizard

Type a name for this task. The task name can be the same name as the program name.

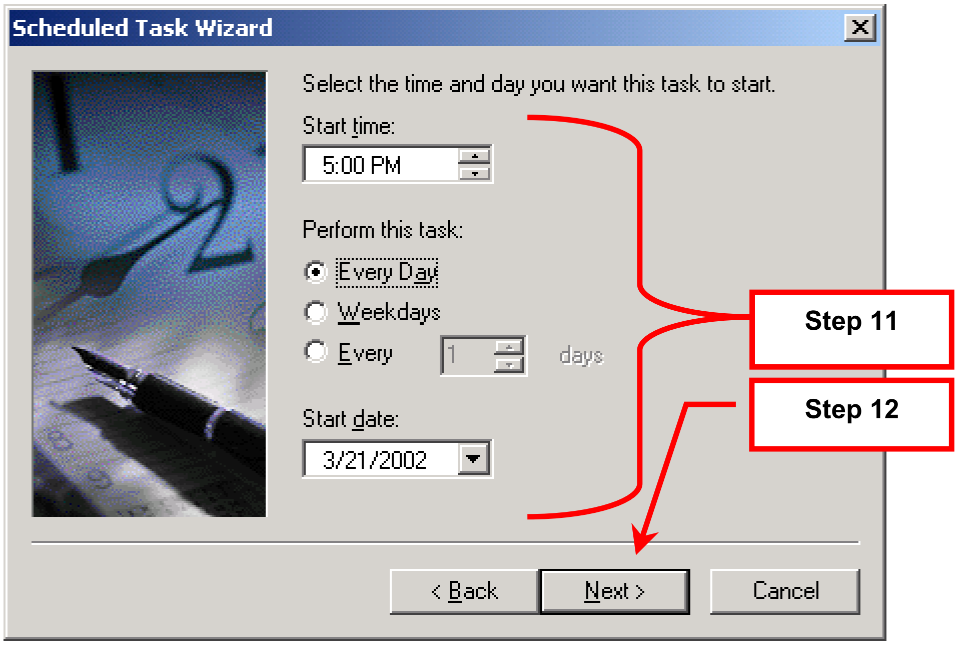
auto_ftp

Perform this task:

○ Daily
○ Weekly
○ Monthly
○ One time only
○ When my computer starts
○ When I log on

11. Select the starting date and time.

12. Press “Next” to continue.



Scheduled Task Wizard

Select the time and day you want this task to start.

Start time: 5:00 PM

Perform this task:

● Every Day

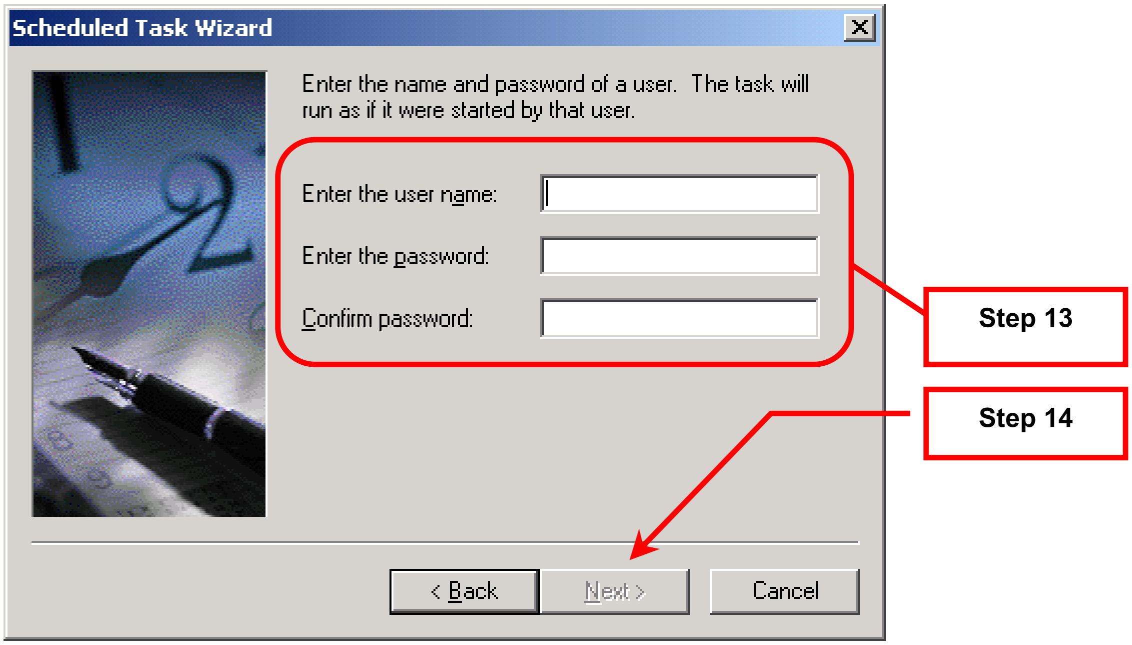
○ Weekdays

○ Every 1 days

Start date: 3/21/2002

13. Enter the Windows 2000 account name and password for the scheduled task to run under.

14. Press “Next” to continue.



Scheduled Task Wizard

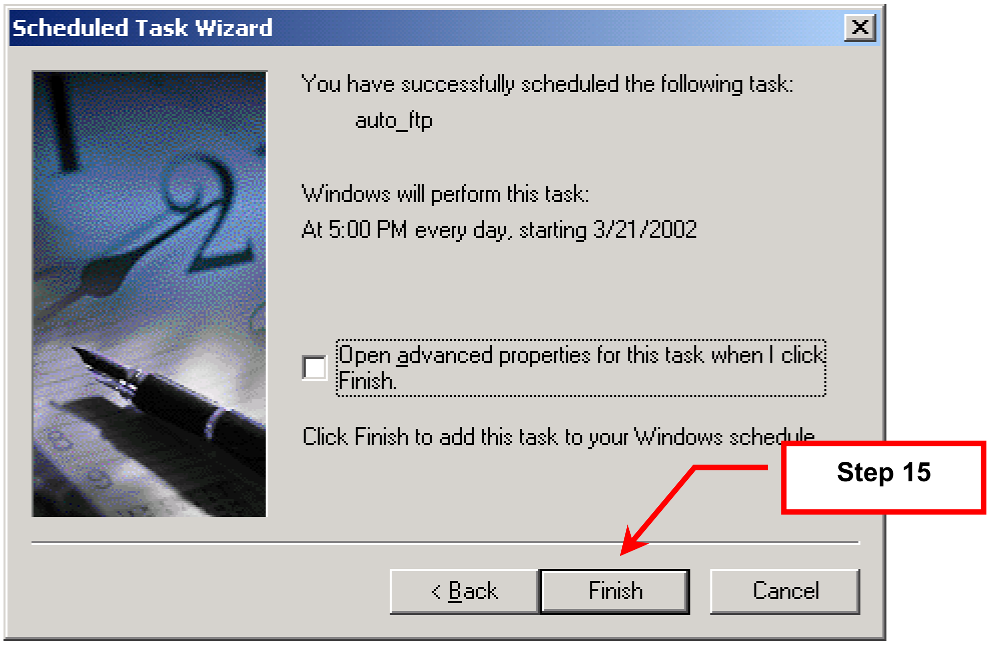
Enter the name and password of a user. The task will run as if it were started by that user.

Enter the user name:

Enter the password:

Confirm password:

15. Click “Finish” to complete the configuration.



Scheduled Task Wizard

You have successfully scheduled the following task:

auto_ftp

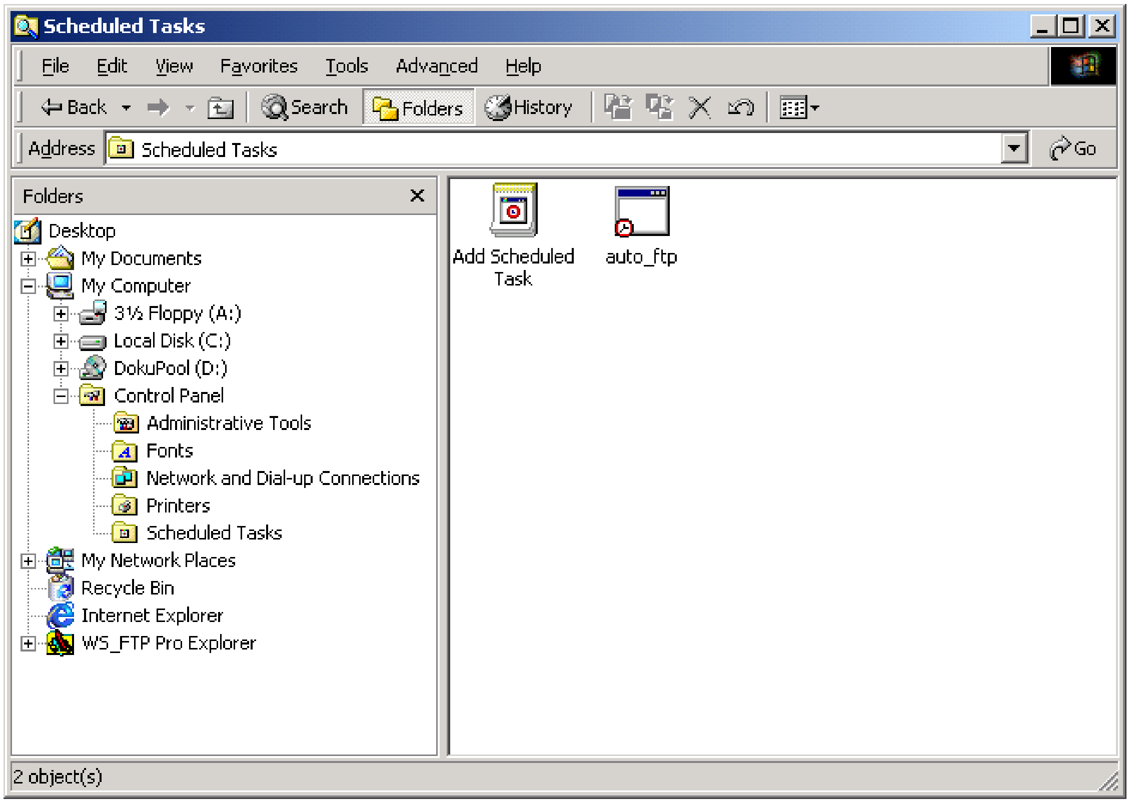
Windows will perform this task:

At 5:00 PM every day, starting 3/21/2002

□ Open advanced properties for this task when I click Finish.

Click Finish to add this task to your Windows schedule.

Once the configuration is complete, the ‘auto_ftp’ icon will appear in the Scheduled Tasks folder.



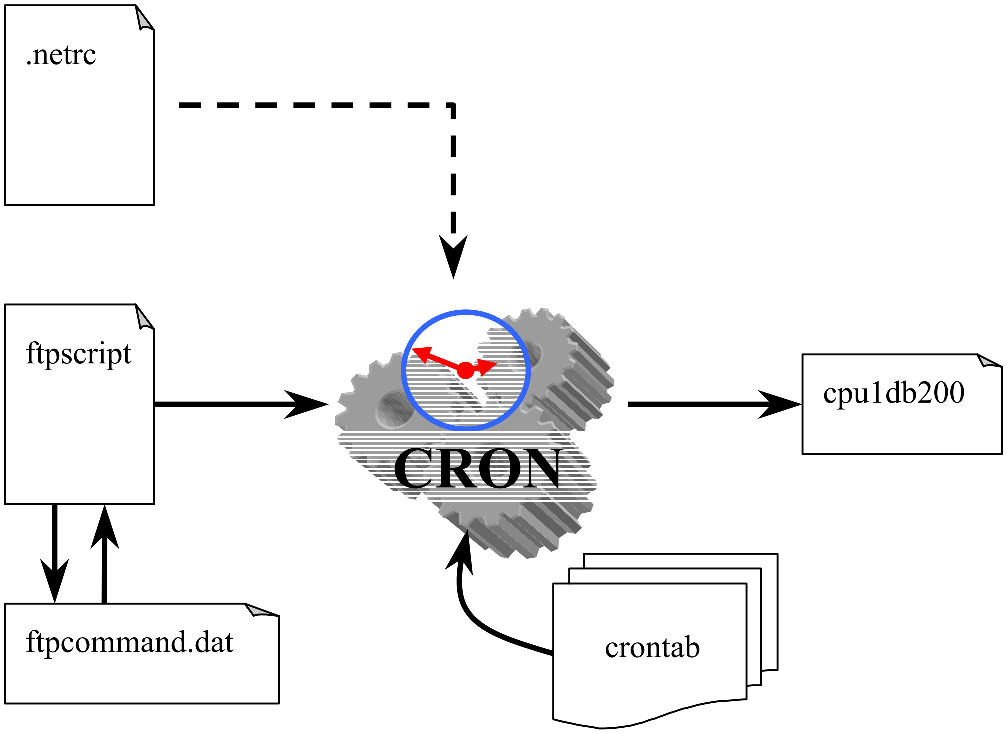
FTP Client with UNIX

For this example, a PC with Red Hat Linux 7 was utilized to retrieve a file from the CP 443-1 IT.

OVERVIEW

In this section, a shell file will be presented which can be run with the UNIX Cron daemon to automatically retrieve, via ftp, a file from the CP 443-1 IT module. The shell file redirects a text file of ftp commands as input to an ftp command line. This section is concluded with step-by-step instructions on how to schedule this shell file for automatic execution on a UNIX platform.

An overview of the utilized files and services appears below. Each item will be discussed in detail in the subsequent sections. As ftp has been presented in previous sections, details of this utility will not be presented as it is a reasonably uniform utility.



USER CONFIGURATION FOR FTP

The file .netrc contains configuration and autologin information for ftp. It resides in the users home directory. For detailed information about this file, consult the online documentation (man netrc).

The .netrc file as used in this example is displayed below. Each portion of the information is explained.

```
default login steve password example
```

default This is a wildcard machine name that allows any machine to connect via ftp.
login This key word is followed by the user name for the account on the IT module
password This key word is followed by the password for the login account before it

The permissions for this file must be set to hexadecimal 600 as shown in the listing below (generated by the command 'ls –al').

```
-rw-------    1 steken   steken         39 Apr  1 10:42 .netrc
```

SHELL PROGRAMMING

There are many shells to choose from on the system used, such as sh, csh, ksh, and bash. For this example, the bash shell was selected (Bourne-Again Shell). The ftp script file is listed below and will be analyzed as well.

```
# ============================================================
#  FTP File Transfer from IT Server to local PC
# ============================================================
#  This shell script obtains a file from a Siemens CP 443-1 IT
#  module using FTP.  The PC from which it is launched is the
#  client and the IT module is the server.  The IT module is a
#  6GK7 443-1GX11-0XE0 (HW v1, FW v1.1.0) and the PC is Red
#  Hat Linux 7.  For details on the commands used in this
#  script, consult the online docomentation (man).
# ============================================================
#
#  STEP 1: Create an ftp command file
#
echo cd /cpu1          >  ftpcommand.dat
echo binary            >> ftpcommand.dat
echo get cpu1db200     >> ftpcommand.dat
echo bye               >> ftpcommand.dat
#
#  STEP 2: Retrieve the file
#
ftp 161.218.120.55 < ftpcommand.dat
```

The top of the script has a brief introduction to the systems accessed during this procedure. It is divided into two sections: creating an ftp command file and retrieving the data from the IT module.

```
#
#  STEP 1: Create an ftp command file
#
echo cd /cpu1          >  ftpcommand.dat
echo binary            >> ftpcommand.dat
echo get cpu1db200     >> ftpcommand.dat
echo bye               >> ftpcommand.dat
```

- STEP 1: Create an ftp command file

The echo command is used to create (>) and append (>>) a data file titled ‘ftpcommand.dat’. The following ftp commands are issued sequentially:

`cd /cpu1`	Change remote (IT) working directory to /cpu1
`binary`	Set ftp transfer mode to ‘binary’ (handles binary and ASCII data)
`get cpu1db200`	Get the file ‘cpu1db200’ from the IT module
`bye`	Terminate the ftp session

```
#
#  STEP 2: Retrieve the file
#
ftp 161.218.120.55 < ftpcommand.dat
```

STEP 2: Retrieve the file

The ftp command is executed to retrieve the file on the IT module. The IP address of the IT module is used to specify the remote host while the data file ‘ftpcommand.dat’ is directed as the input to the ftp command. The .netrc file is referenced transparently for login user names and passwords for remote host ftp connections.

The permissions of the ftp script file must be correctly set in order to execute it. Note the permissions of the files used in this process as shown below (using the command ‘ls –al’).

```
-rw-------    1 steken   steken         39 Apr  1 10:42 .netrc
-rw-r--r--    1 steken   steken         38 Apr  2 12:21 ftpcommand.dat
-rwxr-xr-x    1 steken   steken        873 Apr  2 11:15 ftpscript
-rw-r--r--    1 steken   steken        200 Apr  2 12:21 cpu1db200
```

The ftp script file has ‘read’ and ‘execute’ permissions set for user, group, and others to allow anyone to execute the script.

FTP COMMAND FILE

The file ‘ftpcommand.dat’, which is created by the script file ‘ftpscript’ is listed below.

```
cd /cpu1
binary
get cpu1db200
bye
```

Each of the ftp commands is detailed below.

`cd /cpu1`	Change remote (IT) working directory to /cpu1
`binary`	Set ftp transfer mode to ‘binary’ (handles binary and ASCII data)
`get cpu1db200`	Get the file ‘cpu1db200’ from the IT module
`bye`	Terminate the ftp session

For more information concerning ftp commands, refer to the online help in Linux (man ftp).

UNIX CRON DAEMON

The Unix CRON daemon executes scheduled commands. The commands are entered via the ‘crontab’ utility. For this example, the ‘file ftpscript’ was entered as a command to execute every minute of a weekday from 8 am to 5 pm. The crontab file is listed below.

```
# use /bin/bash to run commands
SHELL=/bin/bash
# run (every minute) (8 am to 5 pm) (every day of month) (every month) (M-F)
* 8-17 * * mon-fri ./ftpscript
```

The first command line specifies the shell environment. For this example, bash was selected (Bourne-Again Shell). The next command line specifies what task to execute and when to execute it:

```
* 8-17 * * mon-fri ./ftpscript
```

There are 6 columns in this line and are detailed below.

Col	**Specifier**	**Description**
1	*	Run the command every minute …
2	8-17	… from 8:00 to 17:00 (5 pm) …
3	*	… on every day of the month (1-31) ...
4	*	… on every month (1-12) …
5	mon-fri	… from Monday to Friday.
6	./ftpscript	Execute ftpscript from the user’s home directory

For further information on cron and crontab, refer to the following manuals online:

```
man cron
man crontab
```

References

SIMATIC NET Industrial Communication CP 343-1 IT and CP 443-1 IT Instructions manual, release 3

SIMATIC NET Industrial Communication NCM S7 for Industrial Ethernet manual, release 4

SIMATIC NET CP 443-1 IT / Help reference on ‘Ftp’ command

Windows 2000 Professional / Help references on ‘Start’ and ‘Ftp’ commands

Red Hat Linux 7 online documentation