

DEFINE FTPD

The DEFINE FTPD command initiates an FTP Daemon (server).

Syntax: `DEFine FTPd ID=id [,PORT=21] [,MAXACTive=3] [,UNIX={Yes|No|Binary}]`
`[,TRANSlate=name16] [,TIMEOut=2m] [,BSize=64K]`
`[,WELCOMe=member] [,EXTtypes={Yes|No}]`
`[,EXTRADATA={FAIL|WARN|IGNORE|ACCEPT}]`
`[,DYNfiles={Yes|No}] [,ALLowabort={Yes|No}] [,HESitate=0]`
`[,IDLEtimeout=0] [,SITELAST={Yes|No}]`
`[,SSL={YES|NO|YESCLAuth}]`
`[,SSLKEY=member] [,SSLVERsion={SSLV3|TLSV1}]`
`[,SSLCIPHER={ALL|WEAK|STRONG|AES|DES|NULL|HARDware|MEDIUM}]`
`[,SSLDATAconn=={CLEAR|PRIVATE}] [,ZEROerr={Yes|No}]`
`[,IPaddr=ip4addr] [,JOURnal={Yes|No}]`
`[,UPPERcase={Yes|No}] [,SEnDFast={Yes|No}]`
`[,REXX={Yes|No}] [,SEnDWack={Yes|No}]`

- Arguments:
- | | |
|------------|--|
| ID= | - A unique name to identify this Daemon. |
| PORT= | - Specifies the TCP/IP port number to be monitored by this FTP Daemon. The default port is 21. You can specify any value between 0 and 65,535, but you should avoid values below 4096 to prevent collision with ports that have standard uses. You should also be aware that FTP uses the specified port to establish the control connection. When data is transferred another port is assigned for the data connection. |
| BSize= | - The size of the buffers that FTP will use. The default size is 65,536. You may override to any value between 4096 and 131,072. |
| EXTtypes= | - Most GUI FTP clients “lie” about file types. They always specify “binary” transfers. Since this will not work if the file contains text, the External Types Table maps files to specific types by examining the extension.
Yes - When YES (the default) is specified, the FTP Daemon will consult the EXTYPES.L member (as loaded by the RELOAD EXTYPES command).
No - The FTP Daemon will not consult the EXTYPES.L member and all processing options must be specified explicitly. |
| JOURnal= | - For HFS files only, this parameter controls journaling for recovery purposes.
Yes - Causes journaling when transferring HFS files.
No - Journaling will not be performed (default). |
| UPPERcase= | - The RFC’s defining FTP behavior state that the client can only rely on the numeric value of each reply. The text is for human use only. If translation issues require it, then this parameter can ensure that text be uppercase only.
Yes - Causes responses to the client to be shifted to uppercase.
No - Responses to the client are mixed case (default). |
| SEnDWack= | - Normally, the FTP Daemon transmits data as quickly as it can be written to the network. However, depending on file system overhead, this can monopolize the CPU and/or network. This parameter can be used to cause the Daemon to pace outbound transmission. |
-

DEFINE FTPD (continued)

- Yes - Causes the Daemon to wait for a response from the remote client after each transfer buffer is sent. This helps "pace" transmissions to prevent monopolizing the CPU and network.
- No - (Default) The Daemon does not automatically wait for each buffer to be acknowledged before filling another one.
-
- REXX= - It is possible for FTP users to execute REXX procedures in the TCP/IP partition. However, this exposes the stack to delays and instability. If care is taken, however, this capability may be enabled.
- Yes - Permits users to execute REXX execs by means of the SITE REXX command. Be aware that allowing this feature exposes the stack to potential security, performance, and integrity issues.
- No - (Default) The SITE REXX command is not permitted.
-
- SENDFast= - Normal operation has the FTP Daemon filling one buffer while the other is being transmitted. This is generally sufficient to keep the data flowing at its maximum speed. However, if system configuration is such that this provides inadequate buffering, an alternate method will allow additional buffering.
- Yes - Transfer buffers are filled and queued until the total bytes queued reaches a value equal to four transfer buffers.
- No - (Default) Two transfer buffers are filled and queued for outbound transmittal. A buffer is not refilled until the remote stack acknowledges receipt of the data.
-
- HESitate= - This value is used to "pace" data transmission when there is concern that the data transfer rate may exceed what the network and system can support. When in effect, a pause of the specified length is effected following each SEND and RECEIVE. Allowable values range from 0 through 99,999 300th seconds.
- Specifying a value of "0" (the default) prevents pacing.
-
- IDLEtimeout= - If non-zero, this specifies the maximum time (in 300th seconds) that an FTP session can remain idle before being terminated. Idle time does not accrue while a data connection is open. Allowable values range from 0 (default) through 500m (minutes).
-
- SSL= - When defining an FTP Daemon, you must choose whether connections will be "normal" (unencrypted, clear text) or "secure" (encrypted and authenticated at some level). Use of SSL requires that the client and remote host also support encryption. When SSL is employed, it is a normal practice to use a port other than the standard value of 21.
- NO - The default. Encrypted connections are not supported.
- YES - Connections will be encrypted.
- YESAUth - Connections will be encrypted and the client must provide an authorization certificate.
-
- SSLKEY= - The *library.sublibrary.member* from which SSL processing will obtain members of types ".prvk", ".cert", and ".root".
-
- SSLVERsion= - This specifies the minimum version of the SSL or TLS protocol that clients must use when connecting.
- SSLV3 - SSL version 3.00 (Default)
- TLSV1 - SSL version 3.01. The TLS protocol contains significant security corrections and enhancements, but not all clients support it.
-

DEFINE FTPD *(continued)*

SSLCIPHER= - For SSL sessions, this parameter determines which ciphers will be available.

ALL - Default. All ciphers are permitted.

WEAK - The following weak ciphers will be permitted:

- 09 RSA_DESCBC_SHA
- 08 RSA_DES40CBC_SHA1
- 62 RSAEXPORT_DESCBC_SHA

MEDIUM - The following medium strength ciphers will be permitted:

- 2F RSA_AES128CBC_SHA
- 0A RSA_3DESCBC_SHA
- 09 RSA_DESCBC_SHA

STRONG - The following strong ciphers will be permitted:

- 35 RSA_AES256CBC_SHA
- 2F RSA_AES128CBC_SHA
- 0A RSA_3DESCBC_SHA

AES - The following ciphers will be permitted:

- 35 RSA_AES256CBC_SHA
- 2F RSA_AES128CBC_SHA

DES - The following ciphers will be permitted:

- 0A RSA_3DESCBC_SHA
- 09 RSA_DESCBC_SHA
- 08 RSA_DES40CBC_SHA1
- 62 RSAEXPORT_DESCBC_SHA

NULL - The following null ciphers will be permitted:

- 02 RSA_NULL_SHA1
- 01 RSA_NULL_MD5

HARDware - When HARWARE is specified, the Crypto Assist hardware will be queried for all available CPACF assists. Caution must be exercised when using this option to ensure that your CPU supports the KMC query instruction.

SSLDATAconn= - This option controls encryption on the data connection.

CLEAR - Causes data to be transmitted in the clear, without encryption (default).

PRIVATE - Causes data to be transmitted in an encrypted state.

SITELAST= - This parameter determines a user's ability to override SITE values that were specified on individual DEFINE FILE commands.

DEFINE FTPD (continued)

- Yes - When YES is specified, SITE commands will override values explicitly coded on DEFINE FILE commands. This does not include values obtained from the EXTTYPES table.
- No - Specifying NO (the default) prevents users from overriding values specified by DEFINE FILE commands.
-
- DYNfiles= - In some cases it may be desirable to allow users to specify files by DLBL, rather than restrict them to the TCP/IP file system.
- Yes - When YES (the default) is specified, files can be accessed outside of the TCP/IP file system by specifying them by DLBL.
- No - Specifying NO will prevent access to files by DLBL, restricting access only to those files defined via the DEFINE FILE command. This is the recommended setting to prevent unintended access to files.
-
- ALLowabort= - Once data transmission begins, the control connection remains idle until it completes. An exception is the "abort" command. If enabled, the Daemon monitors the control connection during data transfer and will allow the client to abort the transfer.
- Yes - When YES (the default) is specified, the client can send an abort command (ABRT) to cause the current data transmission to stop.
- No - The Daemon ignores any input on the control connection until the data transfer completes.
-
- EXTRADATA= - This option controls what is done when an incoming text file has "extra" data at the end. "Extra" data (which can only occur in a text file) is defined as a text string that is not correctly ended with a CR/LF or other valid delimiter.
- FAIL - (Default) The transfer fails and the file is not stored. An FTP343E message is generated and a 5xx failure code is sent to the client.
- WARN - An FTP343W message is generated, but the file is stored. A normal code is sent to the client. The extra data is discarded.
- IGNORE - No messages are generated. The file is stored and the extra data is discarded.
- ACCEPT - No messages are generated. The file is stored with the extra data, as if it were correctly-delimited complete record.
-
- MAXACTIVE= - Each FTP Daemon supports multiple concurrent sessions. This parameter allows you to set the maximum number of user sessions that can be active at any given time. Values range from 1 to 32,767. The default is 3.
-
- UNIX= - In general, GUI FTP clients interpret and display information related to the FTP session. In order for this to happen, the information provided by the Daemon must be in a form that is expected by the client. The expected format is what is produced by a Unix Daemon.
- Binary - When the FTP client specifies "BINARY" (TYPE I), a true binary transfer will take place. The DIR command will return a directory list in the standard Unix format that is understood by GUI clients.
- Yes - The FTP Daemon will mimic Unix operation. This is important if you will be using "graphical" (GUI) FTP clients, since many of these cannot recognize a VSE-style directory list. If continued Unix mode operation is not desired, it can be turned-off by using an FTP "CD \" or "SITE UNIX OFF" command.
-

DEFINE FTPD (continued)

	No	- The FTP Daemon will operate in "VSE mode". This is the default. Directory lists will be returned in "VSE" format, with information pertinent to mainframe-based files. To entry Unix mode, the user may use a "CD /" or "SITE UNIX ON" command.
WELCOME=		- This specifies the name of the VSE member that contains the text that will be sent to the user at the start of each new session. If the name, for example, is "WELCOME", then "WELCOME.L" must be catalogued in the LIBDEF chain as part of the TCP/IP initialization. No text is used in columns 73-80.
TRANslate=		- Specifies the name of an optional translate table for ASCII/EBCDIC translation. This specification overrides the SITE TRANSLATE command. It does not override the TRANSLATE= parameter of the DEFINE FILE command associated with the transfer. If you want FTP users to be able to specify a translate table, you should omit this parameter on the DEFINE FTPD command.
TIMEOut=		- Specifies a time-out interval for the data connection. Although the FTP protocols do not recognize the concept of time-out, it is essential from a practical standpoint. When the time interval is exceeded without a response of any kind from the remote host, the FTP session is terminated. Note that this occurs only during actual data transmission; the Daemon will wait for the IDLETIMEOUT value on a new command. The default value is 2 minutes. The maximum value is 500 minutes.
ZEROerr=		- This parameter determines how the Daemon will react when asked to process an empty (null) file.
	Yes	- A 500-level error message (fatal error) is generated when transferring an empty file (one with zero data bytes) is attempted. This is the default.
	No	- The transfer of an empty file is considered acceptable.
IPaddr=		- If specified, an inbound connection request will have the remote IP address compared against this value. Depending upon the value established by the CONNECT_SEQUENCE command, selection will be either first-fit or best-fit.
		Matching is not performed when the client and Daemon are executing under the same stack.

Example:

```
IPN237I define ftpd,id=ftp01,port=21,unix=yes,welcome=welcome,maxact=5
FTP900I FTP Daemon: FTP01 listening on 192.168.1.161,21
```

- Notes:
- There is no MODIFY FTPD command. To change specifications, you must delete and redefine the FTP Daemon.
 - File transfers can easily monopolize your network interface and even the network itself. Running multiple concurrent file transfers can result in reduced transfer speed and can degrade network response time. Consider this carefully when determining the number of FTP Daemons that you will define. See the *TCP/IP for VSE Installation Guide* for more information about defining FTP Daemons.
 - The FTPBATCH utility program can be configured to run as an FTP server. This allows you to adjust its processing priority to a lower value without affecting the time-dependent portions of the stack.
 - See the *TCP/IP for VSE User's Guide* for information about running in UNIX compatibility mode.
-

DEFINE FTPD *(continued)*

Related Commands:	CONNECT_SEQUENCE	- Control how connection requests are allocated by IP address pattern-checking.
	DEFINE FILE	- Define a file in the TCP/IP file systems and associate it with a file I/O driver.
	DEFINE TRANSLATION	- Load and control ASCII/EBCDIC translation tables.
	DELETE FTPD	- Terminate a File Transfer Protocol Daemon.
	PORTRANGE	- Control range for dynamic port assignment.
	QUERY ACTIVE	- Displays the status of active Daemons.
	QUERY FTPDS	- Displays the status of the File Transfer Protocol Daemons.
	RELOAD	- Reload a control table.
