WU-FTPD 2.6.1 release notes

HP-UX 11i v1, HP-UX 11i v2, HP-UX 11i v3

HP Part Number: 5900-1547
Published: January 2011

Legal Notices

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing here should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Contents

1 WU-FTPD 2.6.1 Release Notes

This document discusses the most recent product information pertaining to WU-FTPD 2.6.1. It also discusses how to install WU-FTPD 2.6.1 on the HP-UX 11i v1, HP-UX 11i v2, and HP-UX 11i v3 operating systems.

This document addresses the following topics:

Announcement

The File Transfer Protocol (FTP) enables you to transfer files between a client host system and a remote server host system. On the client system, a file transfer program provides a user interface to FTP; on the server, the requests are handled by the FTP daemon, `ftpd`. WU-FTPD 2.6.1 is an HP implementation of the FTP daemon based on the replacement FTP daemon, developed at Washington University.

WU-FTPD 2.6.1 is the latest version of WU-FTPD 2.6.1 available on the HP-UX 11i v1, and HP-UX 11i v3 operating systems. It is available for download at:

http://h20293.www2.hp.com/

For the HP-UX 11i v2 operating system, WU-FTPD 2.6.1 is available in the core HP-UX operating system.

Starting with March 2008, WU-FTPD 2.6.1 for HP-UX 11i v3 is also available in the DVD delivered as part of the HP-UX 11i v3 operating system.

For the HP-UX 11i v1 operating system, WU-FTPD 2.6.1 adds new functionality to WU-FTPD 2.4, which is delivered as part of the core HP-UX 11i v1 operating system.

IMPORTANT: If you do not want to use the new features of WU-FTPD 2.6.1, you need not modify your existing configuration settings apart from certain exceptions discussed in "Compatibility Information" (page 42).

The following RFCs are implemented in WU-FTPD 2.6.1:

- RFC 959 (*FILE TRANSFER PROTOCOL (FTP)*)
- RFC 1639 (*FTP Operation Over Big Address Records (FOOBAR)*)
- RFC 2428 (*FTP Extensions for IPv6 and NATs*)

What Is In This Version

The revision of WU-FTPD 2.6.1, `B.11.11.01.014` on the HP-UX 11i v1 operating system contains defect fixes.

For information on defect fixes, see "Defects Fixed in This Release" (page 45).

WU-FTPD 2.6.1 Features

Following are the WU-FTPD 2.6.1 features supported on the HP-UX 11i v1, HP-UX 11i v2 , and HP-UX 11i v3 operating systems:

NOTE: Except for the TLS/SSL feature, all the features discussed in this section are available in WU-FTPD 2.6.1 on the HP-UX 11i v1 operating system.

Support for TLS/SSL

The Transport Layer Security/Secure Socket Layer (TLS/SSL) feature enables the HP-UX FTP product to use the security features provided by OpenSSL. When this feature is enabled, HP-UX FTP provides a secured FTP session and a secure file transfer.

This section discusses the various components used by TLS/SSL to provide security services. It also discusses the prerequisites for configuring the TLS/SSL feature, the procedure to generate certificates and keys using OpenSSL, and to configure an FTP client and server in an TLS/SSL environment.

NOTE: The TLS/SSL feature is available on the HP-UX 11i v2 and HP-UX 11i v3 operating systems.

You can install the WU-FTPD 2.6.1 enhancement bundle, which you can download from http://www.software.hp.com, to obtain the TLS/SSL feature on the HP-UX 11i v2 operating system. The WU-FTPD 2.6.1 enhancement bundle contains the latest core patch required for the TLS/SSL feature on the HP-UX 11i v2 operating system.

The WU-FTPD 2.6.1 software bundle contains the FTP daemon with SSL support for the HP-UX 11i v3 operating system. You can download the WU-FTPD 2.6.1 software bundle from the software depot at http://www.software.hp.com

IMPORTANT: WU-FTP 2.6.1 includes the software developed by the OpenSSL project for use in the OpenSSL toolkit available at:

http://www.openssl.org/

This section addresses the following topics:

- "Cryptography Algorithm" (page 7)
- "Prerequisites for Configuring the TLS/SSL Feature" (page 8)
- "Generating Certificates and Keys Using OpenSSL 0.9.7m" (page 9)
- "Configuring a WU-FTPD TLS Server and an FTP Client" (page 14)
- "Basic Configuration for Secured File Transfer" (page 18)

Cryptography Algorithm

The TLS subsystem uses the following components to provide services, such as integrity checking, authentication, and confidentiality:

- **Private key algorithms**, or symmetrical cryptography. This component uses a shared secret and the key, for both encryption and decryption of a message. Input data is mathematically processed using the private key algorithm and the key, to produce the ciphertext output that must be decrypted by the recipient. Commonly used private key algorithms include `DES`, `Blowfish`, `AES`, and `IDEA`.
- **Public key algorithms**. These algorithms use two mathematically related keys to separate the process of encryption and decryption. By using functions that are easy to perform in one direction but difficult to perform in the opposite direction, the two keys provide a high level of security if large numbers are used. Commonly used public key algorithms include `RSA`, `El Gamal`, and `Diffie-Hellman`.

  While establishing a TLS session, you can use public key cryptography to exchange a session key that is used in a private key algorithm. You can also use these public keys to authenticate the server and, if required, the client, and to provide session-level encryption and confidentiality for the entire session.

- **Hash algorithms**. These algorithms are a set of one-way functions that accept a variable length input, and, after mathematical processing, produce a fixed length output. The transformations of the data produce a fingerprint of the input. The minor changes to the input appear as large changes in the output. Popular hash algorithms include `SHA-1`, `MD5`, and `RIPEMD`.

  Hash algorithms are used for integrity checking; that is, to ensure that data is not tampered during transmission.

Prerequisites for Configuring the TLS/SSL Feature

Following are the prerequisites for configuring the TLS/SSL feature:

- The OpenSSL software

  OpenSSL is an open source product that offers a general purpose cryptography library and implementation of the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols. OpenSSL is tested and supported on different HP-UX operating systems. OpenSSL A.00.09.07m is the latest version of OpenSSL available on the HP-UX 11i v2 operating system. It is available to download at:

  http://www.software.hp.com

  The release notes for OpenSSL A.00.09.07m is available at:

  www.hp.com/go/hpux-security-docs. On this page, select **HP-UX OpenSSL Software**.

- The WU-FTPD 2.6.1 TLS enhancement bundle

  The FTP-TLS enhancement bundle, `ftp-ssl-ncf`, contains TLS enhancement libraries for the FTP client and server. The *ftp*(1) client and the *ftpd*(1M) server use these enhancement libraries with OpenSSL to perform security operations.

Certificates and Authorities

A certificate is a collection of information that uniquely identifies a client or a server. It includes descriptive fields, such as the name of an organization and its location, and cryptographic information, such as keys and signatures.

The private key of an asymmetrical key pair can be used to sign the content that, when decrypted using the public key, establishes the signature. This signature can be used to offer proof of identity. The public key infrastructure (PKI) uses a hierarchy of trustworthiness for the validation of identities, in addition to signing certificates and keys. This is in contrast to the web of trust used in pretty good protection (PGP), which has no central authority.

The central authority in a PKI issues a Certificate Authority (CA), a definitive certificate that contains the information and the public key of the server. This CA can be used to sign other certificates, by signing the public key of a requesting body, such as your server, with the private key. The trust in identity is transitive, because the CA is recognized by all the involved parties as authoritative: *"I trust the CA, and the CA says that it is you, so it must be true."*

Certificates can be revoked because of expiration or compromise in security. To do this, the issuing body provides a certificate revocation list (CRL) that identifies the certificates to be invalidated. This is also trusted because strong proof is provided through the trust mechanisms.

Certificates are available in different formats, though Privacy Enhanced Mail (PEM) is the most widely used format. The PEM encoding is an ASCII text representation of the binary data in the `ASN.1` format. The `X.509` standard defines the distinguished name (DN) format used in these certificates.

A certificate contains the following information that accompanies the cryptographic keys:

- Common name (CN) being certified
- Organization (O) associated
- Organizational unit (OU), such as a department within an organization
- City or location (L) where an organization is located
- State or province (SP) where the city is located
- Country (C) in the International Organization for Standardization (ISO) format (such as U.S.)

The DN is a combination of the different certificate information. The PEM-encoded certificate contains this information along with the DN of the issuer, the validity period of the certificate, various administration information, such as a serial number of the certificate, and any other required information, such as Netscape-specific tags. These certificates are used to establish the identity and trustworthiness of the presenter, such as a server or a client. These certificates are also used to authenticate the connecting party and to take appropriate action, such as allowing a connection to proceed, and mail relaying, or entry into a network. You can either use the commercial TLS/SSL certificates (certs) to verify the identity of the WU-FTPD 2.6.1 server, or create your own certificates for the WU-FTPD 2.6.1 servers.

Generating Certificates and Keys Using OpenSSL 0.9.7m

The FTP client in an HP-UX operating system (HP-UX FTP) is compatible only with standard X.509 certificates in PEM format. HP-UX FTP supports certificates of the following encryption types:

- Rivest Shamir Adleman (RSA) encryption
- Digital Signature Algorithm (DSA) encryption

You can use any encryption to generate certificates to use with HP-UX FTP to secure the file transfer. For information on creating RSA and DSA certificates, see "Creating DSA Certificates and Keys" (page 13).

The OpenSSL script, `/opt/openssl/misc/CA.pl`, can be used to generate certificates and keys. By default, the certificate files are created in an encrypted format using the Data Encryption Standard (DES) encryption. You must log in as a superuser and modify the `CA.pl` script to prevent the created certificate files from being DES encrypted.

NOTE: Third party CAs, certificates, and keys in the PEM format can also be used in the FTP client and server.

For example, if you already have the third party X.509 CA certificate in PEM format and you want to use this certificate for the FTP server, specify the path of the certificate in the FTP server configuration file, that is, `CAfile=/etc/opt/certs/CA.pem`. Similarly, you can also use third party certificates and key by specifying their appropriate locations in the configuration file or on the command line.

Creating RSA Certificates and Keys

Follow this procedure to generate certificates and keys:

1. Change the directory to `/opt/openssl/misc`:

   ```
   cd /opt/openssl/misc
   ```

2. Copy the `CA.pl` script to the `CA.pl.ORIGINAL` script:

   ```
   cp CA.pl CA.pl.ORIGINAL
   ```

3. Replace the entries marked with numbers in the following `CA.pl` script:

   ```
   exit 0;
   } elsif (/^-newcert$/) {
     # create a certificate

   system ("$REQ -new -x509 -keyout newkey.pem -out newcert.pem $DAYS"
   );❶

   $RET=$?;
    print "Certificate is in newcert.pem, private key is in newkey.pem\n"
    } elsif (/^-newreq$/) {
   system ("$REQ -new -keyout newkey.pem -out newreq.pem $DAYS");
   ❷

   $RET=$?;
   print "Request is in newreq.pem, private key is in newkey.pem\n";
    } elsif (/^-newreq-nodes$/)
   ```

   ❶ Replace this line with the following:

   ```
   system ("$REQ -new -nodes -x509 -keyout newkey.pem -out newcert.pem $DAYS");
   ```

   ❷ Replace this line with the following:

```
system ("$REQ -new -nodes -keyout newkey.pem -out newreq.pem
$DAYS");
```

The only change is the addition of the `-nodes` option while generating certificates. If you do not include this option, you must use the configuration or command-line option `password` in the FTP server and `ssl_password` in the FTP client, respectively.

NOTE: You must modify the first line in the `CA.pl` script to the location of the perl interpreter on your system. Otherwise, the following error message is logged in the `/var/adm/syslog/syslog.log` file:

```
interpreter “/opt/perl/bin/perl” not found
```

4. Follow this procedure to create your own CA, and to create certificates and keys for your FTP server:

   a. Create a CA:

   ```
   $ ./CA.pl –newca
   ```

   The following message is displayed:

   ```
   CA certificate filename (or enter to create)
   ```

   Enter the file name or press **Enter**.

   The following message is displayed:

   ```
   Making CA certificate...
   Generating a 1024 bit RSA private key
   .................++++++....++++++
   writing new private key to
   './demoCA/private/cakey.pem'
   Enter PEM pass phrase:
   ```

   Enter the passphrase.

   NOTE: To secure your CA, select a unique passphrase and sign a certificate.

   The following message is displayed:

   ```
   Verifying - Enter PEM pass phrase:
   ```

   Enter the passphrase again.

   The following message is displayed:

   ```
   You are about to be asked to enter information
   that will be incorporated into your certificate request.
   ```

   Enter the organization name, location, and your name.

   After you answer the questions prompted by the `./CA.pl –newca` command, the following files are created:

- The `./demoCA/cacert.pem` file. This is the CA certificate file you can exchange with communication partners for TLS authentication or verification.
- The `./demoCA/private/cakey.pem` file. This is the private key file of the CA and is passphrase-protected. You can use this private key to sign or revoke certificates.

NOTE: Do not exchange the private key file with communication partners.

b. Generate the certificate and the key pair for the FTP server:

```
$ ./CA.pl -newreq
```

The following output is displayed:

```
Generating a 1024 bit RSA private key...
+++++............................++++++
writing new private key to 'newkey.pem'
-----
You are about to be asked to enter information that will
be incorporatedinto your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave
some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

Enter your organization name, location, and name.

The `./CA.pl -newreq` command creates the following files:

- The private key of the FTP server (`./newkey.pem`)
- The original (unsigned) certificate request (`./newreq.pem`)

c. Certificate using the CA created in Step a:

```
$ CA.pl -sign
```

A signed public certificate, `./newcert.pem` (with its public key), is created for the FTP server.

5. Create a subdirectory `security` under the `/etc/ftpd` directory:

```
mkdir -p /etc/ftpd/security
```

6. Change the directory location to `security`:

```
cd /etc/ftpd/security
```

7. Copy the previously created CA certificate, the FTP server certificate, and the key from the `/opt/openssl/misc/` directory to the `/etc/ftpd/security` directory:

```
cp /opt/openssl/misc/demoCA/cacert.pem
/etc/ftpd/security/ftpd-rsa-ca.pem
```

```
cp /opt/openssl/misc/newkey.pem
/etc/ftpd/security/ftpd-rsa-key.pem
cp /opt/openssl/misc/newcert.pem
/etc/ftpd/security/ftpd-rsa-cert.pem
```

The FTP server is now ready with the signed public certificate and the private key.

Creating DSA Certificates and Keys

This section provides an example to create DSA certificates and keys.

NOTE: You must create DSA certificates only if you want to use DSA certificates instead of RSA certificates.

Following is an example to create DSA certificates and keys:

1. Create DSA parameters:

   ```
   openssl dsaparam -out dsap.pem 1024
   ```

2. Create a DSA CA certificate and private key:

   ```
   openssl req -x509 -newkey dsa:dsap.pem -keyout cacert.pem
   -out cacert.pem
   ```

3. Create the CA directories and files:

   ```
   /opt/openssl/misc/CA.pl -newca
   ```

   Enter `cacert.pem` when prompted for the CA file name.

4. Create a DSA certificate request and private key (a different set of parameters can optionally be created first):

   ```
   openssl req -out newreq.pem -newkey dsa:dsap.pem
   ```

5. Sign the request:

   ```
   CA.pl -signreq
   ```

   The `newcert.pem` and `privkey.pem` files are created. `newcert.pem` is the certificate that must be loaded by `ftpd` and `cacert.pem` is the CA certificate that must be loaded by FTP to verify the server certificate.

NOTE: By default, the `CA.pl` script requests for a password to protect the private keys. If you are protecting the password with a PEM passphrase, enable the `ftpd -z password=value` option and set the appropriate password.

Configuring a WU-FTPD TLS Server and an FTP Client

This section addresses the following topics:

- "Configuring an FTP Server in a TLS/SSL Environment" (page 14)
- "Configuring an FTP Client in a TLS/SSL Environment" (page 15)

Consider the following points before configuring an FTP TLS server and an FTP client:

- You cannot use TLS security mechanism to secure third party file transfers (PROXY transfer).
- TLS security mechanism does not use the TCP `sendfile()` API to transfer data contents. Therefore, even if the `sendfile()` API is configured, the TLS security mechanism overrides the configuration.
- The `usetls`, `rsacert`, `rsakey`, and `CAfile` are the minimum set of configuration flags or options that must be enabled for securing FTP control connection using TLS. This is also the minimum configuration that is sufficient for a user to login from an FTP client provided the certificate sent by the FTP client is successfully verified by the CA certificate loaded by the FTP server.
- If both the TLS/SSL and Kerberos security features are enabled in FTP, the TLS/SSL feature obtain precedence over the Kerberos feature during logon. Therefore, the user is prompted for the username and password even though Kerberos is enabled in the system.

Configuring an FTP Server in a TLS/SSL Environment

To configure an FTP server in a TLS/SSL environment, complete the following steps:

1. Ensure that the OpenSSL software is installed in the system.
2. For the HP-UX 11i v2 operating system, the WU-FTPD 2.6.1 software bundle provides the FTP product bundle and the SSL libraries as two independent products. So, ensure that the `ftp-ssl-ncf` FTP TLS enhancement software is installed in the system. Run the following command to ensure that the software is installed:

   ```
   # swlist -l product | grep ftp-ssl-ncf
   ```

   The following output is displayed if the software is installed in the system:

   ```
   ftp-ssl-ncf   B.11.23.01.001     ftp-ssl-ncf web release
   ```

For the HP-UX 11i v3 operating system, the WU-FTPD 2.6.1 software bundle provides the FTP server and the SSL libraries as a single product. So no additional software is required to be installed on the system.

3. Configure OpenSSL and generate X.509 certificates and keys before starting the FTP server.
4. Enable TLS configuration for the FTP server using either of the following methods:

   • Using `-z` command-line option in *ftpd*(1M).

   • Using the TLS configuration file. To use the configuration file, specify the following option as part of the command-line argument for *ftpd*(1M):

   ```
   ftpd -z config=/etc/ftpd/security/tls.conf
   ```

5. Use one or more of the following options to configure TLS:

   ```
       allow_auth_ssl
       bad_auth_ssl_reply
       certsok
       clientcert
       logalldata
       protect_user/tlsonly
       tlsdata
       usetls
       CAfile=file
       CApath=dir
       authmode=OPTION
       cert/rsacert=file
       certpass=OPTION
       cipher=OPTION
       config=file
       crlfile=file
       crldir=dir
       debug=level
       dhparam=file
       dsacert=file
       dsakey=file
       key/rsakey=file
       password=value
       randfile=file
       systemcertdir=dir
   ```

   For information on the configuration options, see *ftpd*(1M).

NOTE: The TLS configuration flag `usetls` enables TLS security mechanism in *ftpd*(1M). Therefore, if you do not enable this flag, you cannot configure *ftpd*(1M) with TLS even if you enable all other TLS configuration flags and options.

Configuring an FTP Client in a TLS/SSL Environment

To configure an FTP client in a TLS/SSL environment, complete the following steps:

1. Ensure that the OpenSSL software in installed in the system.
2. For the HP-UX 11i v2 operating system, the WU-FTPD 2.6.1 software bundle provides the FTP product bundle and the SSL libraries as two independent products. So, ensure that the `ftp-ssl-ncf` FTP TLS enhancement software is installed in the system. Run the following command to ensure that the software is installed:

   ```
   # swlist -l product | grep ftp-ssl-ncf
   ```

   The following output is displayed if the software is installed in the system:

   ```
   ftp-ssl-ncf   B.11.23.01.001     ftp-ssl-ncf web release
   ```

   For the HP-UX 11i v3 operating system, the WU-FTPD 2.6.1 software bundle provides the FTP server and the SSL libraries as a single product. So no additional software is required to be installed on the system.
3. Configure the OpenSSL certificates and keys before you start the FTP client.

   NOTE: This step is optional and required only if you use Client Certificates for authentication.
4. Load the SSL-related variables using either of the following methods:
   - **Using the Environment Variables**

     To load the FTP SSL-related environment variables, use the following command:

     ```
     #export environment_variable_name=value
     ```

     To unset the variable, run the following command:

     ```
     #unset environment_variable_name
     ```

     Following are the environment variables in FTP:

     `FTP_USESSL` Specifies if the client enables TLS security mechanism.

     `FTP_SSL_MODE` Specifies the secure mode. If mode is set to `tls`, the client sends the `AUTH_TLS` command to the server. If mode is set to `ssl`, the client sends the `AUTH_SSL` command to the sever. If mode is set to secure, the client sends the `AUTH_TLS` command. If this fails, the client sends the `AUTH_SSL` command to the server.

     `FTP_SSL_VERIFY_MODE` Specifies if the peer needs to be verified. If this flag is set to `1`, peer is verified. By default, the verify flag is disabled.

     `FTP_SSL_DEBUG_MODE` Specifies if the SSL features must be invoked in debug mode. If the debug mode is set to `2`, extended logging is performed.

`FTP_SSL_NOFALLBACK`	Specifies if SSL fallback needs to be enabled. By default, fallback is enabled.
`FTP_SSL_PROT`	Specifies whether the data channel encryption is enabled. By default, it is enabled.
`FTP_SSL_RANDFILE`	Specifies the file used for seeding random number generator.
`FTP_SSL_LOGFILE`	Specifies the logfile for the debug mode.
`FTP_SSL_CONFIG_FILE`	Specifies the file that contains the configuration options related to TLS.
`FTP_SSL_CERT_FILE`	Specifies the location of the client's certificate file.
`FTP_SSL_DSACERT_FILE`	Specifies the location of the client's DSA certificate file.
`FTP_SSL_KEYT_FILE`	Specifies the client's key file.
`FTP_SSL_DSAKEY_FILE`	Specifies the location of the client's DSA Key file.
`FTP_SSL_CIPHER`	Specifies the cipher list.
`FTP_SSL_CA_FILE`	Specifies the CA certificate.
`FTP_SSL_CA_PATH`	Specifies the pathname for CA certificate.
`FTP_SSL_CRL_FILE`	Specifies the CRL file location for the FTP client.
`FTP_SSL_CRL_PATH`	Specifies the CRL file pathname.
`FTP_TLS_PASSWD`	Specifies the password to decrypt the PEM key file(s).

NOTE: For information on the default values, see the *ftp*(1) manpage.

Alternatively, you can also create a script that contains all the `export` commands and another script that contains all the `unset` commands:

Following are sample entries in a script that contains the `export` commands:

```
FTP_USESSL=1
FTP_SSL_MODE=secure
FTP_SSL_VERIFY_MODE=1
FTP_SSL_DEBUG_MODE=2
FTP_SSL_NOFALLBACK=1
FTP_SSL_PROT=1
FTP_SSL_FILE_MODE=pem
FTP_SSL_RANDFILE=/dev/urandom
FTP_SSL_LOGFILE=/tmp/ssl.log
FTP_SSL_CONFIG_FILE=flist.txt
FTP_SSL_CERT_FILE=/home/SSL/CERTS/client-cert.pem
FTP_SSL_DSACERT_FILE=/home/SSL/CERTS/dsaclient-cert.pem
```

```
FTP_SSL_KEYT_FILE=/home/SSL/CERTS/server-key.pem
FTP_SSL_DSAKEY_FILE=/home/SSL/CERTS/dsaclient-key
FTP_SSL_CA_FILE=/home/SSL/CERTS/ca-cert.pem
```

- **Using the Configuration File**

  You can include all the environment variables in a configuration file and invoke FTP as follows:

  ```
  # ftp -z config=config_filename server_name
  ```

- **Using the Command Line**

  ```
  ftp -z debug=2 -z secure -z logfile=/tmp/ssl.log -z\
  CAfile=/var/opt/ftp/CA-Certs/ca-cert.pem -z \
  CApath=/var/opt/ftp/CA-Certs/ -z\
  rsacert=/var/opt/ftp/CA-Certs/client-cert.pem -z\
  rsakey=/var/opt/ftp/CA-Certs/client-key.pem server_name
  ```

Basic Configuration for Secured File Transfer

This section discusses the basic configuration required for secured file transfer in an FTP server and client.

To configure secured file transfer in an FTP server, complete the following steps:

1. Generate the following certificates and key using HP-UX OpenSSL with the procedure discussed in “Generating Certificates and Keys Using OpenSSL 0.9.7m” (page 9):
   a. X.509 RSA Certificate Authority (CA).
   b. X.509 RSA server certificate signed by the CA certificate (certificate file).
   c. X.509 RSA private key associated with the RSA server certificate (key file).
2. Copy the CA file, certificate file, and key file to the `/etc/ftpd/security` directory in the server, for example, `/etc/ftpd/security/ca.pem`, `/etc/ftpd/security/ftpd-rsa-cert.pem`, and `/etc/ftpd/security/ftpd-rsa-key.pem`, respectively.
3. Configure the FTP server using either of the following methods:

   - Using Command-Line Options

     Include the command-line options in the FTP service entry in the `/etc/inetd.conf` file as follows:

     ```
     ftp stream tcp6 nowait root /usr/lbin/ftpd ftpd -l -L -a
     -z usetls -z tlsdata -z
     cert=/etc/ftpd/security/ftpd-rsa-cert.pem -z
     ```

```
key=/etc/ftpd/security/ftpd-rsa-key.pem -z
CAfile=/etc/ftpd/security/ftpd-rsa-ca.pem
```

- Using the Configuration File

  Specify the TLS configuration file in the FTP service entry in the `/etc/inetd.conf` file.

  Following is the FTP service entry in the `/etc/inetd.conf` file:

  ```
  ftp stream tcp6 nowait root /usr/lbin/ftpd ftpd -l -L -a
  -z usetls -z config=/etc/ftpd/security/tls.conf
  ```

  Following are the contents of the `/etc/ftpd/security/tls.conf` TLS configuration file:

  ```
  usetls
  tlsdata
  cert=/etc/ftpd/security/ftpd-rsa-cert.pem
  key=/etc/ftpd/security/ftpd-rsa-key.pem
  CAfile=/etc/ftpd/security/ftpd-rsa-ca.pem
  ```

To configure secured file transfer in an FTP client system, complete the following steps:

1. Generate the following certificates and key using HP-UX OpenSSL with the procedure discussed in "Generating Certificates and Keys Using OpenSSL 0.9.7m" (page 9):
   a. X509 RSA Certificate Authority (CA).
   b. X509 RSA server certificate signed by the CA certificate (certificate file).
   c. X509 RSA private key associated with the RSA server certificate (key file).
2. Copy the certificate file and key file to the home directory of the user in the client system, for example, `/home/user1/certificate.pem`, and `/home/user1/private-key.pem`, respectively.
3. Copy the CA file to a global location in the client system.

   NOTE: This step is optional and required only if you are using client certificates for authentication.

4. Start the FTP client using one of the following methods:
   - **Using Environment Variables**

     To start the FTP client using environment variables, export the following environment variables using the following commands:

     ```
     export FTP_USESSL=1
     export FTP_SSL_CA_FILE=/etc/ftpd/security/ca.pem
     export FTP_SSL_CERT_FILE=/home/user1/certificate.pem
     export FTP_SSL_KEYT_FILE=/home/user1/private-key.pem
     ```

   - **Using Command-Line Options**

     To start the FTP client using command-line options, run the following command:

```
ftp -z CAfile=/etc/ftpd/security/ca.pem -z
cert=/home/user1/certificate.pem -z
key=/home/user1/private-key.pem <server-name>
```

- **Using the Configuration File**

  To start the FTP client using a configuration file, run the following command:

  ```
  ftp -z config=<config-file> <server-name>
  ```

  where:

  config-file Specifies the name of the configuration file.

  server-name Specifies the name of the server to which date must be transferred.

5. Start the FTP client to initiate a secured file transfer to the FTP server system.

Following is a sample output when the FTP client connects to the FTP server:

```
client:/tmp>ftp server-machine
 Connected to server-machine.
 220 server-machine FTP server (Revision 1.1 Version wuftpd-2.6.1
   (PHNE_36065) Fri May 30
 15:30:32 GMT 2008) ready.
 234 AUTH TLS OK.
 [TLSv1/SSLv3, cipher DHE-RSA-AES256-SHA, 256 bits]
 Name (server-machine:root): abc
 232 User abc auto-logged in.
 Remote system type is UNIX.
 Using binary mode to transfer files.
 ftp> prot on
 200 PROT P ok.
 TLS/SSL protection of data connections on.
 ftp> ls
 200 PORT command successful.
 150 Opening ASCII mode private data connection for /usr/bin/ls.
 total 96
 -rw-r--r-- 1 abc users 23 May 30 14:01 end.txt
 -rw-rw-rw- 1 abc users 13 May 25 14:57 start.txt
 226 Transfer complete.
 ftp> get end.txt
 200 PORT command successful.
 150 Opening BINARY mode private data connection for end.txt (23 bytes).
 226 Transfer complete.
 23 bytes received in 0.07 seconds (0.33 Kbytes/s)
 ftp> by
 221-You have transferred 23 bytes in 1 files.
 221-Total traffic for this session was 1108 bytes in 2 transfers.
 221-Thank you for using the FTP service on charokee.ind.hp.com.
 221 Goodbye.
 client-machine:/tmp>
```

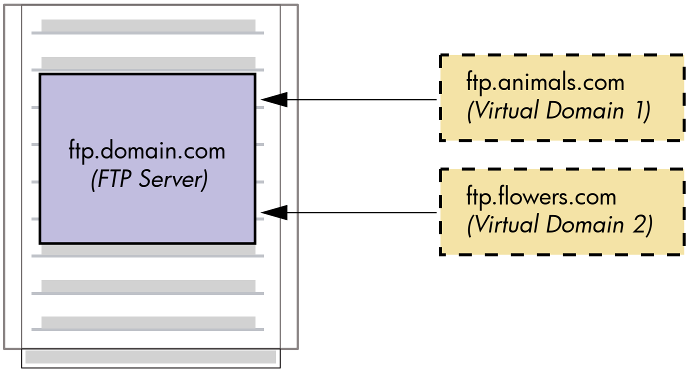
Virtual FTP Support

Virtual FTP support enables you to manage an FTP server for multiple domains on the same machine.

Virtual FTP allows an administrator to configure a system to display a different banner, log file, and directory to a user when the user is connected to different domains on the same system. The advantage of virtual FTP support is that the identity of the machine is hidden. Additionally, this feature enables a single machine to act as multiple FTP servers for multiple domains.

Figure 1 shows a graphical representation of an FTP server, `ftp.domain.com`, hosting two virtual domains, `ftp.animals.com` and `ftp.flowers.com`.

Figure 1 Structure of an FTP Server Hosting Two Virtual Domains



In Figure 1, a user connected to the FTP server `ftp.domain.com` through the domain `ftp.animals.com` receives a different banner and directory than a user who is connected to the same server through the domain `ftp.flowers.com`.

Setting up Virtual FTP Support

The configuration file `/etc/ftpd/ftpservers` contains a set of virtual domain names that the FTPD server can use for each virtual domain. Using the virtual domain name, you can define the FTP configuration files `ftpaccess`, `ftpusers`, `ftpgroups`, `ftphosts`, and `ftpconversion` files on a per-domain basis. If you want to place a copy of one or all the FTP configuration files in the virtual host directory, create a directory in the `/etc/ftpd` directory with a name similar to the virtual domain name, and copy the FTP configuration files to that directory. Otherwise, you can use the master copy.

NOTE: A sample configuration file exists in the `/usr/newconfig/etc/ftpd/examples` directory.

Example 1 The /etc/ftpd/ftpserver Configuration File Entry

The following example shows a possible entry in the `/etc/ftpd/ftpservers` configuration file:

```
123.123.123.123 /etc/ftpd/somedomain
```

In this example, when an FTP client connects to the server using the IP address 123.123.123.123, the FTPD server searches for the configuration files `ftpaccess`, `ftphosts`, `ftpusers`, `ftpgroups`, and `ftpconversions` under the directory `/etc/ftpd/somedomain`. If a match is not found or an invalid directory path is encountered, the default master configuration files in the `/etc/ftpd` directory are used instead.

Support for Virtual FTP

Virtual FTP is supported in the following ways:

- Without the *ftpservers*(4) file - By using the master `/etc/ftpd/ftpaccess` configuration file. For more information, see "“Without ftpservers (4) File” (page 22)."
- With the *ftpservers*(4) file - By using the virtual domain's `ftpaccess` configuration file. For more information, see “With *ftpservers*(4) File” (page 25).

Without ftpservers (4) File

WU-FTP 2.6.1 supports the following directives for Virtual FTP support in the master `/etc/ftpd/ftpaccess` configuration file:

- `virtual address allow username [ username ... ]`
- `virtual address deny username [ username ... ]`
- `virtual address private`
- `virtual address { root|banner|logfile } path`
- `virtual address { hostname|email } string`
- `virtual address incmail emailaddress`
- `virtual address mailfrom emailaddress`

Usage

This section describes the functionality of the various directives.

The `virtual address allow username`and `virtual address deny username` directives

These directives are used to allow or deny real and guest users. They can be used in the `/etc/ftpd/ftpaccess` file and the virtual domain specific `ftpaccess` file.

```
virtual address allow username [ username ... ]
virtual address deny username [ username ... ]
```

The `virtual address private` directive

This directive is used to deny anonymous FTP login. By default, anonymous users are allowed to log in a virtual FTP setup.

```
virtual address private
```

The `virtual address root path` and `virtual address banner path` directives

These directives are used to display the banner message and are used in the `/etc/ftpd/ftpaccess` file.

```
virtual address root path virtual address banner path
```

NOTE: The `virtual address root path` directive must be mentioned for the `virtual address banner path` directive to work. This directive is additionally required to allow anonymous FTP access in a virtual FTP setup.

The `virtual address banner path` directive must be used only in the `/etc/ftpd/ftpaccess` file and not in the virtual domain's `ftpaccess` file. The above directive overrides the `banner path` directive. If the master `/etc/ftpd/ftpaccess` configuration file has `banner path` directive but not `virtual address banner path` directive, the `banner path` directive does not have any effect on the behavior of the *ftpd*(1M) daemon.

The `virtual address root path` directive can also be used in conjunction with the `virtual address email string` directive.

For more information, see “The `virtual address root path` and `virtual address email string` directives.”

The `virtual address logfile path` directive

This directive is used to change the file where all the logging information of *ftpd*(1M) must be written. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: The `virtual address logfile path` directive does not require the `virtual address root` directive. This directive overrides the `logfile path` directive. If the `/etc/ftpd/ftpaccess` file has the `logfile path` directive but does not have the `virtual address logfile path` directive, then the `logfile path` directive does not affect the behavior of the *ftpd*(1M) daemon.

The `virtual address hostname string` directive

This directive is used to change the default hostname of the FTP server. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: The `virtual address hostname string` directive does not require the `virtual address root` directive. This directive overrides the `hostname string` directive. If the `/etc/ftpd/ftpaccess` file has the `hostname string` directive but does not have the `virtual address hostname string` directive, then the `hostname string` directive does not affect the behavior of the *ftpd*(IM) daemon. If both the `virtual address hostname string` directive and the `hostname string` directive are present in the `/etc/ftpd/ftpaccess` file, the `virtual address hostname string` directive does not affect the behavior of the *ftpd*(1M) daemon.

The `virtual address root path` and `virtual address email string` directives

These directives are used to change the email address of the FTP archive maintainer. These directives are used in the `/etc/ftpd/ftpaccess` file.

NOTE: The `virtual address root path` directive must be mentioned for the `virtual address email string` directive to work. This directive overrides the `email string` directive. If the `/etc/ftpd/ftpaccess` file has the `email string` directive but does not have the `virtual address email string` directive, the `email string` directive does not affect the behavior of the *ftpd*(1M) daemon. If both `virtual address email string` and `email string` directives are present in the `/etc/ftpd/ftpaccess` file, the `virtual address email string` takes precedence over the `email string` directive.

The `virtual address incmail emailaddress` directive

This directive is used to change the email address for anonymous upload notifications. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: The `virtual address incmail emailaddress` directive does not require the `virtual address root path` directive. This directive overrides the `incmail emailaddress` directive. If the master `/etc/ftpd/ftpaccess` configuration file has the `incmail emailaddress` directive but does not have the `virtual address incmail emailaddress` directive, the `incmail emailaddress` directive does not affect the behavior of the *ftpd*(1M) daemon.

The `virtual address mailfrom emailaddress` directive

This directive is used to change the sender's email address for anonymous upload notifications. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: The `virtual address mailfrom emailaddress` directive does not require the `virtual address root path` directive. This directive overrides the `mailfrom emailaddress` directive. If the master `/etc/ftpd/ftpaccess` configuration file has the `mailfrom emailaddress` directive but does not have the `virtual address mailfrom emailaddress` directive, the `mailfrom emailaddress` directive does not affect the behavior of the *ftpd*(1M) daemon.

With *ftpservers*(4) File

Use the following directives to achieve the configurations described in "Without ftpservers (4) File" (page 22):

- `virtual address allow username [ username ... ]`
- `virtual address deny username [ username ... ]`
- `virtual address private`
- `root path`
- `banner path`
- `logfile path`
- `hostname string`
- `email string`
- `incmail emailaddress`
- `mailfrom emailaddress`

Usage

This section describes the functionality of the various directives.

The `virtual address allow username` and `virtual address deny username` directives

These directives are used to allow or deny real and guest users to log in a virtual FTP setup. These directives can also be used in the master `/etc/ftpd/ftpaccess` file.

The `virtual address private` directive

This directive is used to deny anonymous access to virtual FTP setup. This directive can also be used in the master `/etc/ftpd/ftpaccess` file.

The `root path` directive

This directive is used to allow anonymous FTP access in a virtual FTP setup.

The `banner path` directive

This directive is used to display the banner message. This directive is used in the `/etc/ftpd/ftpacess` file.

NOTE: Do not use the `virtual address banner path` directive in the `ftpaccess` file of the virtual domain because the directive does not have any effect.

The `logfile path` directive

This directive is used to change the path of the *xferlog*(4) file. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: Do not use the `virtual address logfile path` directive in the `ftpaccess` file of the virtual domain because the directive does not have any effect.

The `hostname some.host.name` directive

This directive is used to change the hostname string. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: Do not use the `virtual address hostname some.host.name` directive in the virtual domain's `ftpaccess` file because it does not have any effect.

The `email emailaddress` directive

This directive is used to change the email address of the FTP archive maintainer. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: Do not use the `virtual address email emailaddress` directive in the virtual domain's `ftpaccess` file because it does not have any effect.

The `incmail emailaddress` directive

This directive is used to change the email address for anonymous upload notifications. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: Do not use the `virtual address incmail emailaddress` directive in the virtual domain's `ftpaccess` file because it does not have any effect.

The `mailfrom emailaddress` directive

This directive is used to change the sender's email address for anonymous upload notifications. This directive is used in the `/etc/ftpd/ftpaccess` file.

NOTE: Do not use the `virtual address mailfrom emailaddress` directive in the virtual domain's `ftpaccess` file because it does not have any effect.

Setting up a Virtual FTP Server

The procedure to set up a virtual FTP server is as follows:

1. Set up an IP alias for the FTP server machine using the `ifconfig` command. For example:

   ```
   ifconfig lan0:1 15.70.178.100 netmask 0xffffff00 up
   ```

   The IP address `15.70.178.100` is set as an alias to the interface `lan0`. Now you can access the FTP server machine with `lan0` as the interface, with the IP address `15.70.178.100`.

2. Declare the following directives in the `/etc/ftpd/ftpaccess` file:

   ```
   virtual 15.70.178.100 root/virtual
   virtual 15.70.178.100 banner / virtual/banner.msg
   virtual 15.70.178.100 logfile / virtual/xferlog
   ```

3. Create the `directory /virtual` and the files `banner.msg` and `xferlog` under the `/virtual` directory.
4. Log in as an anonymous user on the virtual FTP server (that is, `ftp 15.70.178.100`). The `banner.msg` file is displayed from the `/virtual` directory.

The root directory of the anonymous user is changed to the directory as specified in the virtual IP address root entry in the `/etc/ftpd/ftpaccess` file. For example:

```
virtual 15.70.178.100 root/virtual
```

In this example, the root directory of the anonymous user is changed to the `/virtual` directory.

You must ensure that the files referenced after changing the root directory exist in the virtual server (similar to the scenario for setting up an anonymous account).

The `privatepw` Utility

The administrative utility, `/usr/bin/privatepw`, is used to update the group access file information in the `/etc/ftpd/ftpgroups` file. The administrator can add, delete, and list enhanced access group information required for the commands `SITE GROUP` and `SITE GPASS`. The `/usr/bin/privatepw` command requires read and write permission for the appropriate ftpgroups file to modify the access group information. For more information, type man 4 ftpgroups at the HP-UX prompt.

New Clauses in the /etc/ftpd/ftpaccess File

The following new clauses are added in the /etc/ftpd/ftpaccess file:

- The email-on load Clause

  You can use this clause to specify email addresses for anonymous upload notifications and also to specify the email address of the sender. By default, the address of the sender is specified as wu-ftpd. You can specify this for virtual hosts also. If the recipient attempts to reply to a notification or if downstream mail problems generate bounces, ensure that the mailfrom address is a valid address, to avoid delivery problems.

  The syntax for the email-on load feature is as follows:

  -
    ```
    mailserver <hostname>
    ```

  -
    ```
    incmail <emailaddress>
    virtual <address> incmail <emailaddress>
    defaultserver incmail <email address>
    ```

  -
    ```
    mailfrom <emailaddress>
    virtual <address> mailfrom <emailaddress>
    defaultserver incmail <emailaddress>
    ```

  -
    ```
    deny-email <case-insensitive-email-address>
    ```

  If you specify virtual host addresses, the addresses only on a particular host receive notification messages of anonymous uploads. Otherwise, notifications are sent to the global addresses.

  The defaultserver addresses apply only to real hosts and not to virtual hosts. Hence, the real host receives notifications of uploads on its default anonymous area. However, with this option set, the virtual hosts are not notified. For more information on the email-on-load feature, see *ftpaccess*(4).

Following are examples of the `email-on-load` feature:

- `mailserver abc.com`

  This specifies the name of a mail server that accepts upload notifications for the FTP daemon. You can use this option to notify any user of anonymous uploads.

- `incmail def@abc.com`

  This specifies the email addresses to be notified of anonymous uploads.

- `mailfrom ghi@abc.com`

  This specifies the sender’s email address for anonymous upload notifications.

- Timeout Values

  You can configure timeout values used within the FTP daemon by using the `timeout` options. Table 1 describes the FTP daemon `timeout` values.

Table 1 FTP Daemon timeout Options

Option	Description
`accept`	The time period for which the daemon waits for an incoming (`PASV`-passive) data connection. The default value is 120 seconds.
`connect`	The time period the daemon waits before attempting to establish an outgoing (`PORT`-port) data connection. The default value is 120 seconds. The `connect` option affects the actual connection attempt. The daemon makes several attempts at regular intervals, sleeping between each attempt, before disconnecting. During the 120-minute timeframe, the daemon continues its attempt to establish a connection. If the daemon fails to establish a connection during this time period, it disconnects.
`data`	The time period the daemon waits for some activity on the data connection. The default value is 1200 seconds.
`idle`	The time period the daemon waits for the next command. The default value is 900 seconds.
`RFC931`	The maximum time period the daemon allows for entire RFC 931 (*Authentication Server*) conversation. The default value is 10 seconds.
`maxidle`	The `SITE IDLE` command allows the remote client to establish a higher value for the idle timeout. With the `maxidle` option set in the `/etc/ftpd/ftpaccess` file, you can override the value set with the `SITE IDLE` command. The default value is 1200 seconds.

The syntax for the `timeout` clauses are as follows:

```
timeout accept <seconds>
timeout connect <seconds>
timeout data <seconds>
timeout idle <seconds>
timeout maxidle <seconds>
timeout RFC931 <seconds>
```

Following are some examples for the `timeout` clause:

- `timeout idle 200`

  This displays the message `Current IDLE time limit is 200 seconds; max 7200`

- `timeout maxidle 6200`

  This displays the message `Current IDLE time limit is 200 seconds; max 6200`

- `timeout RFC931 0`

  This disables RFC 931-based authentication because 0 is specified.

- Enhanced DNS Extensions

  You can use this feature to refuse (or override) an FTP session when a reverse DNS lookup fails.

  The syntax for the enhanced DNS extension feature is as follows:

  ```
  dns refuse_mismatch <filename> [ override ]
  dns refuse_no_reverse <filename> [ override ]
  dns resolveroptions <options>
  ```

- Reported Address Control

  This feature enables you to impose control on the address reported in response to a `PASV`command and on the TCP port numbers that can be used for a passive data connection. When a control connection matching the classless inter-domain routing (`cidr`) requests a passive data connection (`PASV`), the `externalip` address is reported.

  The syntax for controlling the reported address is as follows:

  ```
  passive address <externalip> <cidr>
  ```

  ```
  passive ports <cidr> <min> <max>
  ```

Example 2 The `passive` Clause

The following is an example of a `passive` clause:

```
passive address 10.0.1.15 10.0.0.0/8
```

In this example, clients connecting from the class A network - 10 are informed that the passive connection is listening on the IP address `10.0.1.15`.

```
passive ports 10.0.0.0/8 90 100
```

In this example, if a control connection from the class A network - 10 exits, the port range within 90 and 100 is randomly selected for the daemon to listen.

NOTE: You cannot control the reported address in an IPv6 environment.

- `PORT` and `PASV` Data Connection

  This feature enables the site administrator to selectively allow `PORT` and `PASV` data connections. Usually a connection is not established if the remote IP address of the data connection does not match the remote IP address of the control connection data. You can specify multiple passive addresses to handle complex or multi-gateway networks.

  The syntax for selectively allowing PORT and PASV data connections is as follows:

  ```
  pasv-allow <class> [ addrglob ...]
  port-allow <class> [ addrglob ...]
  ```

  NOTE: You cannot selectively allow `PORT` and `PASV` data connections in an IPv6 environment.

- The `keepalive` Clause

  The `keepalive` clause allows you to control network disconnect by setting the TCP `SO_ALIVE` option for data sockets. You can specify `yes` to set the TCP option, or `no` to use the system default settings, which is usually off. HP recommends that you set the `keepalive` clause to `yes` to retain the network traffic connected.

  The syntax for `keepalive` clause is as follows:

  ```
  keepalive yes no
  ```

- The `/etc/ftpd/ftpaccess log` Clause

  The `log` clause is changed to allow logging transfers to both the `/var/adm/syslog/syslog` and `/var/adm/syslog/xferlog` files. This option enables you to redirect the logging messages for incoming and outgoing transfers to the `/var/adm/syslog/syslog` file. If you do not specify this option, the messages are written to the `/var/adm/syslog/xferlog` file.

  The general syntax to redirect messages is as follows:

  ```
  log sysloglog syslog+xferlog
  ```

- File Retrieval

  You can specify certain clauses to control whether a real or guest user is allowed access to areas on the FTP site other than their home directories.

  The syntax for the clauses that control access to areas on the FTP site is as follows:

  ```
  restricted-uid <uid-range>[...]
  restricted-gid <gid-range>[...]
  unrestricted-uid <uid-range>[...]
  unrestricted-gid <gid-range>[...]
  ```

NOTE: For all these clauses, you must copy the libraries /usr/lib/libnss_files.1 and /usr/lib/libdld.2 to the /usr/lib directory of the current environment.

- Virtual Server

  You can use the virtual server clauses to restrict user access to both the virtual and non-virtual domains. Additionally, you can use the options specified in the virtual clause to display the virtual host name.

  The syntax for the virtual clause is as follows:

```
virtual <address> allow <username> [ username ...]
virtual <address> deny <username> [ username ...]
virtual <address> private
virtual <address> hostname email string
defaultserver deny <username> [ username ...]
defaultserver allow <username> [ username ...]
defaultserver private
```

  Table 2 specifies different virtual clause examples.

Table 2 The virtual Clause Options

The virtual Clause Option	Description
virtual xx.xx.xx.xx allow root	Allows the root user to start an FTP session on the system xx.xx.xx.xx. By default, real and guest users are not allowed to log in to the virtual server unless they are guests and have changed their directory to the virtual root directory. This is applicable only for virtual FTP servers.
virtual xx.xx.xx.xx allow * virtual xx.xx.xx.xx deny root	Denies root users and allows other users to start the FTP session.
virtual xx.xx.xx.xx private	Denies service to anonymous FTP users.
virtual xx.xx.xx.xx hostname telnet2.abc	Prints the string (telnet2.abc) instead of the actual host name in the greeting message and STAT command.
defaultserver deny root	Denies ftp on the default FTP server for the root user. The message FTP LOGIN REFUSEDis logged in the /var/adm/syslog file.
defaultserver private	Denies anonymous ftp connection to the default server. The message FTP LOGIN REFUSED is logged in the /var/adm/syslog file.

- Default Host Name

  This feature defines the default host name of the FTP server that is displayed in the greeting message. If you do not specify this clause, the default host name of the local machine is used.

  The syntax for the specifying the default host name is as follows:

  ```
  hostname <some.host.name>
  ```

  Example 3 The `hostname` Clause

  An example of the `hostname` clause is as follows:

  ```
  hostname telnet2.123.com
  ```

  This clause displays the default host name (`telnet2.123.com`) instead of the actual host name in the greeting message.

- Control Information

  This feature allows you to control the information specified in the greeting message before a remote user logs in. For the greeting message, you can specify the host name and the daemon version, only the host name, or only the message `FTP server ready`. The default `greeting` clause is `greeting full`.

  The syntax for the `greeting` clause is as follows:

  ```
  greeting full brief terse
  greeting text <message>
  ```

  Using the `greeting text <message>` clause, you can print a message different from the standard greeting message.

  Example 4 The `greeting` Clause

  An example for the `greeting` clause is as follows:

  ```
  greeting text Hi!!! Welcome to FTP Server
  ```

  This clause displays the message `Hi!!! Welcome to FTP server` as the greeting message.

- Session Time Limit

  This feature allows you to limit the total time for a session. By default, a limit is not set. Real users are never limited.

  The syntax for limiting the total time of a session is as follows:

  ```
  limit-time {* anonymous guest} <minutes>
  ```

- Treatment of UIDs and GIDs as Guests

  This feature allows you to force the user IDs (UIDs) and group IDs (GIDs) in a range to be treated as guests.

  The syntax for treating UIDs and GIDs as guests is as follows:

  ```
  guestuser <username> [ username ... ]
  realgroup <groupname> [ groupname ... ]
  realuser <username> [ username ... ]
  ```

- Upload and Download Ratios

  You can set the upload and download ratio to limit the user's ability to upload and download files. By default, a ratio is not set.

  The syntax for setting the upload and download ratio is as follows:

  ```
  ul-dl-rate <rate> [ class ...]
  dl-free <filename> [ class ...]
  dl-free-dir <dirname> [ class ...]
  ```

  Example 5 The `ul-dl-rate` Clause

  An example for the `ul-dl-rate` clause is as follows:

  ```
  ul-dl-rate 2
  ```

  For every 1 byte of data that is uploaded, the `ftp` server allows 2 bytes of data to be downloaded.

- The `nice` Clause

  The `nice` clause allows you to modify the `nice` value of the FTP server if the remote user is a member of the named class. If you do not specify the class, use `nice-delta` as the default adjustment to the `nice` value of the FTP server process. The default `nice` value adjustment is used to adjust the `nice` value of the server process. You can use the adjustment only for users who do not belong to any class for which a class-specific `nice` directive exists in the `/etc/ftpd/ftpaccess` file.

  The syntax for the `nice` clause is as follows:

  ```
  nice <nice-delta> [ class ]
  ```

  NOTE: You can specify only negative values for `nice-delta`. Positive values or 0 are ignored.

- The `defumask` Clause

  The `defumask` clause allows you to set `umask` for a file created by the FTP daemon if the remote user is a member of the named class. You can enter multiple `defumask` entries in the `/etc/ftpd/ftpaccess` file. If you do not specify a class for a `defumask` entry, use `umask` as the default for classes that do not have a `defumask` entry.

The syntax for the `defumask` clause is as follows:

```
defumask umask [ class ]
```

Example 6 The `defumask` Clause

The following are some examples for the `defumask` clause:

```
defumask 0177
```

```
defumask 0133 ClassA
```

This clause creates files with the permission `-rw-r--r--` for a user of `ClassA`. For other users, files are created with the permission `-rw-------`.

- Limitations on the Number of Lines of Output

  This feature allows you to limit the number of lines of output that can be sent to the remote client. By default, the limit is set to 20.

  The syntax for controlling the maximum number of lines of output is as follows:

  ```
  site-exec-max-lines <number> [ class ...]
  ```

  Example 7 The `site-exec-max-lines` Clause

  The following are some examples for the `site-exec-max-lines` clause:

  ```
  site-exec-max-lines 200 remote
  site-exec-max-lines 0 local
  site-exec-max-lines 25
  ```

  Example 7 contains three example statements for the `site-exec-max-lines` clause. The first example limits the output from `SITE EXEC` (therefore `SITE INDEX`) to 200 lines for remote users. The second example specifies no limit for local users. The third example sets a limit of 25 lines for all other users.

- Root Directory Specification

  This feature specifies the root directory when a user logs in as an anonymous or guest user.

  The syntax for specifying the root directory is as follows:

  ```
  anonymous-root <root-dir> [ class ]
  ```

  ```
  guest-root <root-dir> [ uid-range ]
  ```

Example 8 The `anonymous-root` Clause

The following are examples of the `anonymous-root` clause:

```
anonymous-root /home/ftp
anonymous-root /home/localftp localnet
```

The first example changes the root directory of all the anonymous users to the directory `/home/ftp`, the anonymous user's current working directory being the home directory. If an FTP user exists in the `/home/ftp/etc/passwd` file, the user's current working directory is the home directory. In the second example, the root directory of all the anonymous users in the class `localnet` is changed to the directory`/home/localftp`, and the FTP user's home directory in `/home/localftp/etc/passwd` specifies the initial current working directory.

Example 9 The `guest-root` Clause

An example of the `guest-root` clause is as follows:

```
guest-root /home/users guest-root /home/staff %100-999 sally
```

The example changes the root directory of all the guest users to the `/home/users` directory. The directory of users in the range 100 through 999 and user `sally` is changed to the `/home/staff` directory, and the current working directory is obtained from their entries in the `/home/staff/etc/passwd` file.

- Server Listening Clause

  This clause enables the server to listen on any address. If you do not set this value, the server listens for connections on all the IP addresses. HP recommends not to use this clause because it breaks virtual hosting.

  NOTE: This option works only when `ftpd` is running in a standalone mode. For more information, see *ftpd*(1M).

  The syntax for enabling the server to listen is as follows:

  ```
  daemonaddress <address>
  ```

  For detailed information on all the clauses in the `/etc/ftpd/ftpaccess` utility, type `man 4 ftpaccess` at the HP-UX prompt.

Enabling the Identification Protocol (RFC 1413)

The Identification Protocol, `/usr/bin/ident`, enables you to determine the identity of a user of a particular TCP connection. For a particular TCP port number pair, `identd` returns a character string that identifies the owner of that connection on the system of the server. You can use the `-I` daemon option to enable RFC 1413-based authentication. By default, this authentication is disabled.

New Feature Related to Data Transfer

The following lists the data transfer features:

- For statistical purposes, you can track the total bytes of data transferred. Also, you can limit the number of data bytes that a user, in any given class, can transfer. You can specify a directive in the `/etc/ftpd/ftpaccess` file to limit the number of bytes incoming, outgoing, or both.

  The syntax for the directive is as follows:

  ```
  data-limit [raw] in out total count [class]
  ```

  A default limit is specified to all the classes for which you have not specified a limit. When the FTP session logs off, this directive prints the number of files and the number of bytes transferred.

- You can limit the number of data files that a user, in the given class, can transfer in a session. You can specify a directive in the `/etc/ftpd/ftpaccess` file to limit the number of incoming files, outgoing files, or both.

  The syntax for the directive is as follows:

  ```
  file-limit [raw] in out total count [class]
  ```

  If you do not specify a class, a default limit is specified to all the classes for which a limit is not specified.

For more information, type `man 4 ftpaccess` at the HP-UX prompt.

Field Added to the `/var/adm/syslog/xferlog` File

A new field is added to the `/var/adm/syslog/xferlog` file to indicate the completion status of the data transfer. A field value of `c` indicates complete transfer.

For more information on the new fields in the `/var/adm/syslog/xferlog` file, see *xferlog*(5).

Command-Line Options

Table 3 describes the new options in WU-FTPD 2.6.1.

Table 3 New Options in WU-FTPD 2.6.1

Option	Description
`-q` and `-Q`	These options determine whether WU-FTPD 2.6.1 uses the PID files.
`-rroot dir`	This option instructs the daemon to change the root directory to the specified root directory immediately after loading.
`-V`	This option causes the program to display copyright and version information and then terminate.

Table 3 New Options in WU-FTPD 2.6.1 *(continued)*

Option	Description
`-w` and `-W`	This option determines if user logins must be recorded in the `/var/adm/wtmp` and `/var/adm/btmp files`.
`-X`	This option does not save the output created by the `-i` and `-o` options to the `/var/adm/syslog/xferlog` file but writes to the `/var/adm/syslog/syslog.log` file.
`-I`	This option enables the use of Identification Protocol (RFC1413) to attempt to determine the username on the client.
`-s` and `-S`	These options run the daemon in standalone operation mode.
`-c<ctrl port>` and `-C<data port>`	These options override the control and the data port numbers that is used by the daemon.
`-U`	For the HP-UX 11i v1 operating system, this option replaces the `sendfiletransfer` option in the `/etc/ftpd/ftpaccess` configuration file.
`-V`	This option prints the copyright and the version information for all utilities (`ftpcount`, `ftprestart`, `ckconfig`, `ftpwho`, `privatepw` and `ftpshut`)

For more details on the new command-line options in WU-FTPD 2.6.1, type man 1M ftpd at the HP-UX prompt.

IPv6 Support

To support IPv6 functionality, you must modify the `/etc/inetd.conf` file as follows:

```
ftp stream tcp6 nowait root /usr/bin/ftpd ftpd -l
```

However, if you specify `tcp` instead of `tcp6`, FTP operates in the IPv4 mode.

Following are the features that support IPv6:

- Implementation of RFC 2428 (FTP Extensions for IPv6 and NATs)

  This RFC specifies a method by which FTP clients and server exchange data connection information, such as port, host address, and type of protocol family, for both IPv4 and IPv6 addresses.

  FTP uses `EPRT`and `EPSV` instead of `PORT`and `PASV`, respectively, for IPv6 connections.

  - `EPRT` - Extended Port

    This command specifies a host port for both IPv4 and IPv6 connections.

Example 10 ERPT Command Output for IPv6 and IPv6 Connections

The following displays the output for the `EPRT` command for both IPv6 and IPv6 connections.

For IPv4:

```
------> EPRT  1 132.235.1.2 50934
```

For IPv6:

```
------> EPRT  2 fe80::260:b0ff:fec1:7b2f 50934
```

  - `EPSV` - Extended Passive

    This command requests a server to listen on a data port and wait for a connection. The response to this command includes only the TCP port number of the listening connection.

Example 11 EPASV Command Output

An example for the `EPASV` command is as follows:

```
ftp> passive
Passive mode on.
------> EPSV
229 Entering Extended Passive Mode (   9495 ).
```

- Implementation of RFC 1639 (*FTP Operation Over Big Address Records (FOOBAR)*)

  This RFC describes a convention for specifying an address other than the default data port for the connection over which data is transferred.

  The commands to accommodate FTP operations over network and transport protocols are specified as follows:

  - `LPRT`

    This command enables you to specify a long address for the transport connection.

Example 12 LPRT Command Output

The following displays the output for the `LPRT` command:

```
------> LPRT6,16,254,128,0,0,0,0,0,0,2,96,176,255,254,193,123,47,
2,198,244200 LPRT command successful
```

  - `LPSV`

    This command requests a server to listen on a data port other than its default port and to wait for a connection rather than initiate one on the receipt of a transfer command.

Example 13 LPASV Command Output

The following displays the output for the `LPASV` command:

```
ftp> passive
Passive mode on.
-------> LPSV
228 Entering Long Passive Mode (6,16,254,128,0,0,0,0,0,0,
2,96,176,255,254,193,123,47,2,134,7)
```

NOTE: The FTP client must use the `-l` option to use the `LPSV` and `LPRT` commands.

The FTP session command `longaddr` toggles the use of the `LPRT` (extended port) and `LPSV` (extended passive) commands. For more information on the `-l` option, type `man 1 ftp` at the HP-UX prompt.

HP-Specific Features

HP has introduced the following features in WU-FTPD 2.6.1:

- Command-Line Options

  Following are the options included in WU-FTPD 2.6.1:

  - `-m number_of_tries`

    Specifies the number of tries for a `bind()` socket call.

  - `-n nice_value`

    Sets the nice value for an WU-FTPD process. When using this option, ensure that the nice clause in the `/etc/ftpd/ftpaccess` file (see *ftpaccess*(4)) is not set.

  - `-B`

    Sets the buffer size of the data socket to blocks of 1024 bytes. The valid range for size is from 1 to 64 (default is 56).

    NOTE: A large buffer size improves the performance of WU-FTPD 2.6.1 on fast links (for example, FDDI) but may cause long connection times on slow links (for example, X.25).

  - `-p` and `-P`

    The `-p` option is used to allow private port access to the client. The `-P` option is used to allow third party access and private port access.

    These options are also available in WU-FTPD 2.4. These options do not exist in the open-sourced version of WU-FTPD 2.6.1 but have been incorporated in HP's port of WU-FTPD for backward compatibility.

Other Features

In addition to the features discussed in the previous sections, WU-FTPD 2.6.1 supports the following features:

- Files greater than 2 GB
- Large user IDs (UIDs) and group IDs (GIDs)
- Trusted system features

Changed and Removed Features

There are no changed or removed features in WU-FTPD 2.6.1.

Compatibility and Installation Information

This section describes the compatibility and installation requirements.

Compatibility Information

Customers currently using WU-FTPD 2.4 do not need to modify their configuration file. WU-FTPD 2.4 is compatible with this release of WU-FTPD. However, HP recommends you to use the WU-FTPD 2.6.1 configuration file delivered with this release to effectively use the new features and changes incorporated in WU-FTPD 2.6.1.

You must modify your configuration settings only for the following instances:

- If you are upgrading to WU-FTPD 2.6.1 on an HP-UX 11i v1 operating system, you must consider the following:
  - The `sendfiletransfer` option in the `/etc/ftpd/ftpaccess` configuration file is replaced with the `-U` option in WU-FTPD 2.6.1.
  - The `suppresshostname` and `suppressversion` options in WU-FTPD 2.4 are replaced with the new `greeting` option in WU-FTPD 2.6.1.

    For more information on WU-FTPD 2.4, see the *WU-FTPD 2.4 Release Notes* at www.hp.com/go/hpux-networking-docs.

    NOTE: WU-FTPD 2.6.1 does not support sublogins for an anonymous FTP user.

    The `suppresshostname` and `suppressversion` options that were present in the previous version of WU-FPTD are removed from the `/etc/ftpd/ftpaccess` file because features provided by these options are already present in the new `greeting` option. If you are using any of these options, you need to modify your `/etc/ftpd/ftpaccess` configuration file accordingly. For more information on these options, type `man 4 ftpaccess` at the HP-UX prompt.

    The option `sendfiletransfer` in the `ftpaccess` configuration file are replaced with `-U` option in `ftpd`.

Installing WU-FTPD 2.6.1

To install WU-FTPD 2.6.1, run the following command at the HP-UX prompt:

```
$ swinstall -s <source>
```

where *<source>* is the location to which you downloaded the WU-FTPD depot.

NOTE: The installation of this upgrade kills all the instances of *ftpd*(1M) and it does not restart *ftpd*(1M) because *ftpd*(1M) is usually invoked by *inetd*(1M). If *ftpd*(1M) is running in standalone mode (using the `-S` or `-s` option), *ftpd*(1M) must be restarted manually after installing this upgrade.

The WU-FTPD 2.6.1 files are installed in the `/usr/contrib/wuftpd` directory. During installation, the `/usr/bin/enable_inet` script backs up the existing WU-FTPD files in the /usr/contrib/wuftpd/save_custom/backup directory and enables the higher version of WU-FTPD by linking the new files to existing file locations.

The enable_inet -r wuftpd command enables you to revert to the previous version of WU-FTPD. To enable the newer version of WU-FTPD, you must run the enable_inet wuftpd command on the HP-UX prompt. The enable_inet status wuftpd command displays the currently active version of WU-FTPD.

If you want to install a general release (GR) patch, you must disable WU-FTPD 2.6.1 by running the following command at the HP-UX prompt before installing the GR patch in an HP-UX 11i v1 system:

```
/usr/bin/enable_inet -r wuftpd
```

This command reverts the system to the base version of FTPD (WU-FTPD 2.4) that is delivered with the core HP-UX 11i v1 operating system.

If you wish to upgrade your operating system and if the Web upgrade version of WU-FTPD is enabled on the existing operating system, you must revert to the previous version of WU-FTPD using the enable_inet -r wuftpd command before upgrading the operating system.

If you want to reinstall the Web upgrade, which is already enabled on the operating system, revert to the previous version of WU-FTPD using the enable_inet -r wuftpd command, before reinstalling the Web upgrade.

Verifying the WU-FTPD 2.6.1 Installation

To verify whether the WU-FTPD 2.6.1 depot is installed successfully on your system, enter the following command at the HP-UX prompt:

```
# swlist -l bundle | grep bundle_name
```

If WU-FTPD 2.6.1 is installed properly, the following output is displayed:

- On an HP-UX 11i v1 operating system

  ```
  WU-FTP-261 B.11.11.01.014 WU-FTPD-2.6.1 special release upgrade
  ```

- On an HP-UX 11i v2 operating system

  ```
  ftp-ssl-ncf   B.11.23.01.001     ftp-ssl-ncf web release
  ```

- On an HP-UX 11i v3 operating system

  ```
  HPUX-FTPServer C.2.6.1.5.0 HPUX FTP Server
  ```

Known Problems and Limitations

WU-FTPD 2.6.1 does not have any known problems and limitations.

Related Information

The following sections discuss the documentation available for WU-FTPD 2.6.1.

Manpages

Table 4 describes the manpages distributed with the WU-FTPD 2.6.1 depot.

Table 4 WU-FTPD 2.6.1 Manpages

Manpage	Description
ftp(1)	User interface to the file transfer program
ftpd(1M)	Server for the Defense Advanced Research Project Agency (DARPA) Internet file transfer protocol.
ckconfig(1)	Utility to verify the path names of the FTP configuration files
ftprestart(1)	Command to remove all the shutdown message files from the real, anonymous, and virtual user accounts
ftpwho(1)	Command that shows the current process information for each user logged into the FTP server
ftpcount(1)	Command that shows the current number of users (and the limit) for each class defined in the `/etc/ftpd/ftpaccess` file
ftpshut(1)	Command that provides an automated shutdown procedure that a superuser can use to notify FTP users when the FTP server is shutting down
privatepw(1)	Utility used to add, delete, and list enhanced access group information in the group access file (`/etc/ftpd/ftpgroups`)
ftpaccess(4)	File used to configure the operation of *ftpd*(1M)
ftpgroups(4)	Group password file for use with the `SITE GROUP` and `SITE GPASS` commands
ftpservers(4)	File that contains the set of virtual domain configuration files, which the *ftpd*(1M) server must use
ftpconversions(4)	*ftpd*(1M) conversion database.
ftpusers(4)	File that contains the local user accounts to which remote logins are rejected by *ftpd*(1M)
ftphosts(4)	File that allows or denies access to certain accounts from various hosts
xferlog(5)	File that contains logging information from the FTP server daemon

Product Documentation

For more information on configuring and administering FTP, see the *HP-UX Remote Access Services Administrator's Guide* at:

www.hp.com/go/hpux-networking-docs

The README files for WU-FTPD 2.6.1 are available in the `/usr/share/doc` directory.

Defects Fixed in This Release

This section describes the WU-FTPD 2.6.1 defects fixed in the HP-UX 11i v1 and 11i v3 operating systems.

It addresses the following topics:

- "Defects Fixed in the HP-UX 11i v1 Operating System" (page 45)
- "Defects Fixed in the HP-UX 11i v3 Operating System" (page 49)

Defects Fixed in the HP-UX 11i v1 Operating System

Table 5 describes the defects fixed in the HP-UX 11i v1 operating system.

Table 5 Defects Fixed in the HP-UX 11i v1 Operating System

Identifier	Description
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.001`)	
JAGad39666	Porting of WU-FTPD 2.6.1.
JAGad68796	IPv6 changes for WU-FTPD 2.6.
JAGad39650	IPv6 changes for FTP client.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.002`)	
JAGad96997	WU-FTPD is not working as expected.
JAGad99478	WU-FTPD is not checking NULL hostname.
JAGad88782	The previous version of WU-FTPD released as a Web upgrade generates `swverify` error when installed above PHNE_23950.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.003`)	
JAGae85593/ QXCR1000730078	Under certain conditions *ftpd*(1M) does not work properly.
JAGae53898/ QXCR1000519011	Enhancement request to log the client IP address along with other information in the `/var/adm/wtmp` file for successful login and to log unsuccessful login attempts to the `/var/adm/btmp` file.
JAGae69021/ QXCR1000522851	*ftp*(1) generates an incorrect transfer report while storing files of size more than 2 GB.
JAGae58493/ QXCR1000729331	`get` command of *ftp*(1) does not function properly.

Table 5 Defects Fixed in the HP-UX 11i v1 Operating System *(continued)*

Identifier	Description
JAGae21322	In an FTP session, when the command `ls` is executed with the pathname of any file followed by `/.`, FTP displays the long listing of the file instead of displaying the error message `not found`. For instance, when the `ls /etc/passwd/.` command is issued in an FTP session, the long listing of the file `/etc/passwd` is displayed.
JAGae12022/ QXCR1000512388	In WU-FTPD 2.6.1, user access cannot be limited through the `/etc/ftpd/ftpusers` file.
JAGae62972/ QXCR1000521254	*ftpd*(1M) fails to do user authentication using *identd*(1M) in IPv6-enabled systems.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.004`)	
JAGaf20839/ QXCR1000535121	*ftpd*(1M) does not work properly under certain situations.
JAGaf11467/ QXCR1000532398	*ftpd*(1M) does not work correctly with certain group IDs.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.005`)	
JAGaf08674/ QXCR1000531669	In some situations, there is a delay in an FTP connection after the FTP client displays the `Connected to` message.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.006`)	
JAGaf62718/ QXCR1000548386	*ftpd*(1M) has a problem when a failure occurs in establishing a data connection with the client.
JAGaf54890/ QXCR1000545542	*ftpd*(1) is unable to log into the virtual domain when the FTP server invoked by *inetd*(1M) uses the *ftpservers*(4) configuration file to enable the virtual hosting feature.
JAGaf40126/ QXCR1000540528	An anonymous user cannot log into a Trusted system.
JAGaf39590/ QXCR1000540388	In an IPv6 environment, a delay may be observed in the FTP connection.
JAGaf39331/ QXCR1000540315	No customer-visible symptoms in most cases.

Table 5 Defects Fixed in the HP-UX 11i v1 Operating System *(continued)*

Identifier	Description
JAGaf35480/ QXCR1000539305	*ftpd*(1M) always uses the primary interface address of the system for the data connection instead of using the address on which the control connection request is received.
JAGaf33866/ QXCR1000538860	In an NFS-mounted file system, which is full, the *ftp*(1) `get` or `mget` command fails without displaying any error message. Also, in some cases in a non-NFS mounted system, which is full, *ftp*(1) `get` or `mget` command fails without displaying any error message. As a result, unreported data loss may occur.
JAGaf32059/ QXCR1000538330	The `restart` command in *ftp*(1M) does not work properly when the restart marker is set to a value greater than or equal to 2 GB.
JAGae79698/ QXCR1000525558	When *ftp*(1M) tries to transfer a file to an NFS-mounted directory in a system where the disk space is full, *ftpd*(1M) displays the following error message, even though transfer operation has failed: `226 Transfer complete`
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.007`)	
JAGaf82539/ QXCR1000555915	*ftpd*(1M) does not correctly process certain configuration information.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.008`)	
JAGaf89900/ QXCR1000558838	*ftpd*(1M) has problem in globbing patterns.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.009`)	
JAGaf87174/ QXCR1000557780	In passive mode, *ftpd*(1M) may assign the same port number for consecutive `PASV` requests for data connections.
JAGaf87739/ QXCR1000558009	*ftpd*(1M) takes long time to transfer files in ASCII mode.
JAGaf91558/ QXCR1000559521	*ftp*(1) has problem in globbing patterns.
JAGaf86407/ QXCR1000557449	*ftpd*(1M) configured with upload directive in *ftpaccess*(4) does not send an error message to the FTP client when a file transfer fails in an NFS-mounted file system.
JAGaf80981/[1] QXCR1000555243	The *ftpd*(1M) man page does not mention the usage of the `/usr/bin/ls` command to support directory listing by *ftpd*(1M) in an anonymous FTP setup.

Table 5 Defects Fixed in the HP-UX 11i v1 Operating System *(continued)*

Identifier	Description
JAGae22345/ QXCR1000513734	*ftpd*(1M) does not clean up certain environment variables when started in stand-alone mode.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.010`)	
JAGag20313/ QXCR1000572236	The directives related to virtual hosting feature are not documented properly in the documentation available for *ftpaccess*(4).
JAGag03440/ QXCR1000734472	*ftpd*(1M) has problem with the `guestserver` clause.
JAGaf85093/ QXCR1000733531	*ftpd*(1M) has problem with certain clauses.
JAGaf78174/ QXCR1000554076	The virtual hosting feature in *ftpd*(1M) does not work when *ftpd*(1M) is started in the stand-alone mode.
JAGae30158/ QXCR1000514723	If the IPv6 feature does not exist in a system, *ftpd*(1M) does not send an error message when an FTP client issues an `EPRT` command.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.011`)	
JAGag46940/ QXCR1000591157	In an IPv6 environment, *ftpd*(1M) does not function properly for certain directives in the *ftpaccess*(4) file.
JAGaf91258/ QXCR1000559419	Certain inputs to *ftpd*(1M) can cause huge delay in the response.
JAGaf71500/ QXCR1000551539	*ftpd*(1M) does not list all the files when a file name glob is used against a directory listing command and the number of files passing the file name glob is more than 1000.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.012`)	
JAGaf91565/ QXCR1000559524	Certain inputs to *ftp*(1) can cause huge delay in response.
QXCR1000774466	A user is restricted login from all hosts, though the deny clause for the user applies to particular hosts only.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.013`)	
QXCR1000927648	*ftp* does not handle some multi-byte character set filenames correctly.
QXCR1000973381	The *ftpd* process does not have the informational command string.

Table 5 Defects Fixed in the HP-UX 11i v1 Operating System *(continued)*

Identifier	Description
QXCR1000965335	When the *ftpwho*(1) command is run, it does not return the expected process information for each connected ftp session.
QXCR1000576150	The default umask for *ftpd*(1M) is set to 022 instead of 027 as mentioned in the manpage. Due to this *ftpd*(1M) does not behave as expected in certain account configurations.
Defects fixed in WU-FTPD 2.6.1 (`B.11.11.01.014`)	
QXCR1000962860	FTP client does not handle some multi-byte character set filenames correctly.
QXCR1001052936	The latest ftpd(1M) man page does not display after installation of FTP v1.013 Web Upgrade depot on HP-UX 11i v1.

1 In an anonymous FTP setup, if you want *ftpd*(1M) to use the `/usr/bin/ls` command, instead of the `/sbin/ls` command, to support directory listing, copy the following library files to the `~ftp/usr/lib/` directory:

- `/usr/lib/libc.2`
- `/usr/lib/libcurses.1`
- `/usr/lib/dld.sl`
- `/usr/lib/libdld.2`

Defects Fixed in the HP-UX 11i v3 Operating System

Table 6 lists the defects fixed in the HP-UX 11i v3 operating system.

Table 6 Defects Fixed in the HP-UX 11i v3 Operating System

Identifier	Description
Defects fixed in WU-FTPD 2.6.1 (`C.2.6.1.3.0`)	
JAGag46940/ QXCR1000591157	In an IPv6 environment, *ftpd*(1M) does not function properly for certain directives in the *ftpaccess*(4) file.
JAGaf91258/ QXCR1000559419	Certain inputs to *ftpd*(1M) can cause huge delay in response.
Defects fixed in WU-FTPD 2.6.1 (`C.2.6.1.5.0`)	
QXCR1000927648	The *ftp* client does not handle some multi-byte character set filenames correctly.
QXCR1000508767	The *ftp* client and *ftpd* daemon do not support secure data transfer.

Table 6 Defects Fixed in the HP-UX 11i v3 Operating System *(continued)*

Identifier	Description
QXCR1000545220	When the *ftpd* daemon logs file transfers in the `/var/adm/syslog/xferlog.log` file, filenames containing 8-bit ASCII characters may be incorrectly logged.
QXCR1000867024	The exceptions in handling file names logged in the `/var/adm/syslog/xferlog.log` file are not documented in the *xferlog*(5) man page.
QXCR1000895696	WU-FTPD 4.0 of 2.6.1 does not create the `/etc/ftpd` directory on installation.
QXCR1000576150	The default umask for the *ftpd* daemon is set to 022 instead of 027 as mentioned in the manpage. Due to this the *ftpd* daemon does not behave as expected in certain account configurations.
QXCR1000965335	When the *ftpwho* command is run, it does not return the expected process information for each connected ftp session.
QXCR1000812507	The *ftpd* daemon does not behave as expected in certain account configurations.
QXCR1000973381	The *ftpd* daemon does not have the informational command string.
Defects fixed in WU-FTPD 2.6.1 (`C.2.6.2.6.0`)	
QXCR1001079010	The `ftp://FTP.FTP-SSL-LIBS` fileset does not have a packaging attribute.