

Unattended file transfer - ectrans

- [ECtrans Association](#)
 - [The genericFtp Options](#)
 - [The genericSftp Options](#)
 - [The genericExec Options](#)
- [ectrans command](#)
 - [Transfer to a Member State host via gateway](#)
 - [Transfer from a Member State host via gateway](#)

The ectrans command allows you to transfer files securely between ECMWF and remote sites. Like the UNIX "rcp" command, ectrans requires no password to be specified for the remote host: the ECaccess gateway performs the security checking. Unlike standard FTP, ectrans is suitable for unattended file transfers in scripts, cron jobs, etc., as it avoids the problems inherent in storing passwords in text files and sending passwords across networks. Even if you don't have a local gateway installed, you can benefit from the ectrans command by using the ECMWF ECaccess gateway. Please note that in this case the transfer is not as secured as when a Member State ECaccess gateway is used.

ECtrans Association

Users who wish to transfer files between ECMWF and Member State servers using ectrans need to declare one or more ectrans associations for the storage/retrieval of the remote file. This can be done either through the Web Toolkit command ecaccess-association-put (see [Web Toolkit - The full featured client](#)) or through the ECaccess Web interface of the target gateway (see [The Web server](#)). For every association (previously known as "msuser"), the hostname, login username and password, target path and transfer protocol need to be specified. Target directories can be located on:

- Member State servers running a standard FTP/SFTP service accessible from the ECaccess gateway. This is known as a "genericFtp"/"genericSftp" protocol and is the most convenient way of getting the files to the system you want, under the specified user ID.
- The server running the ECaccess gateway. This is known as a "genericFile" protocol. All users will share in a common directory the files transferred using this protocol.
- Member State servers running a proprietary application. The administrator provides ectrans with the implementation of the access protocol. The administrator can also use more complex rules to define special target locations for ECMWF users, Member State users or groups of Member State users.

Additional parameters, related to the chosen transfer protocol can be specified in the ECtrans Association.

The protocols available are the following:

- **genericFtp**: standard FTP protocol (ftp options).
- **genericSftp**: standard SFTP protocol (sftp options).
- **genericFile**: transmit a file on the target Gateway file system.
- **genericExec**: transmit and execute a file on the target Gateway file system (exec options).

The "genericFile" and "genericExec" protocols are only available on remote Gateways.

The various options available for each protocol are described in the following tables (please note the "genericFile" protocol has no option).

The "debug" option works for each protocol and creates debugging information in the logs of the ECaccess Gateway which can be collected by the administrator for investigating problems related to Ectrans.

WARNING: please note option names are case-sensitive and unknown option names are ignored by Ectrans.

WARNING: if you don't have the latest version of the software some options might be missing.

The genericFtp Options

Name	Default	Type /Unit	Comment
ftp.passive	"false"	boolean	Allow selecting the Passive or Active FTP mode. In Passive mode, the FTP client initiates both connections to the server, solving the problem of firewalls filtering the incoming data port connection to the client from the server. In Active mode, the FTP client connects from a random unprivileged port (N > 1023) to the FTP server's command PORT, usually port 21. Then, the client starts listening to port N+1 and sends the FTP command <code>PORT N+1</code> to the FTP server. The server will then connect back to the client's specified data port from its local data port, which is usually port 20.
ftp.extended	"false"	boolean	Its use is required when connecting to a remote IPv6 server (EPSV/EPRT commands are issued instead of PASV/PORT commands when opening data channels).
ftp.lowPort	"false"	boolean	Allow forcing the FTP client to bind a privileged port ($500 \leq N \leq 1023$) instead of an unprivileged port ($N > 1023$) when using the Active FTP mode.
ftp.commandTimeout	"60000"	milliseconds	Enable/disable a timeout when opening the control channel. The value must be > 0. A timeout of zero is interpreted as an infinite timeout.
ftp.dataTimeout	"60000"	milliseconds	Enable/disable a timeout when reading from the data channel. The value must be > 0. A timeout of zero is interpreted as an infinite timeout.

ftp.portTimeout	"60000"	milliseconds	Enable/disable a timeout when opening the data channel specified by a PORT command. The value must be > 0. A timeout of zero is interpreted as an infinite timeout.
ftp.listenAddress	-	string	Allow specifying the listen address used by the FTP client (e.g. when waiting for an incoming data connection). By default, the system will pick up a valid local address. A value of "0.0.0.0" will cause the binding of all the valid network interfaces.
ftp.sendBufferSize	"-1"	integer	Allow specifying a buffer size when sending data on the data channel.
ftp.receiveBufferSize	"-1"	integer	Allow specifying a buffer size when receiving data from the data channel.
ftp.setNoop	"NOOP"	string	Allow specifying the "No-Operation" command. This command does not affect anything at all. It performs no action other than having the server send an OK reply. This command is used to keep connections with the FTP servers "alive" (connected) while nothing is being done.
ftp.port	"21"	short	Allow specifying the port number of the remote FTP server to connect to (used to open the control channel).
ftp.usecleanpath	"false"	boolean	When using the "mkdir" command this option allow forcing the FTP client to fix directory pathnames (e.g. remove multiple instances of /).
ftp.usenlist	"false"	boolean	Force using the NLIST command on the remote FTP server when doing a directory listing instead of the standard LIST. This options is required for some FTP servers which do not allow (or do not implement) the LIST command.
ftp.usetmp	"true"	boolean	Force using a temporary name when transmitting a file to the remote FTP server. Once the transmission has completed successfully, the file is renamed with its final target name. The temporary file name is by default the file name with the ".tmp" suffix concatenated to it, however this behaviour can be customised with the "mksuffix", "usesuffix", "prefix" and "suffix" options.
ftp.mkdirs	"true"	boolean	Allow creating the directory named by the target pathname, including any necessary but non-existent parent directories.
ftp.prefix	-	string	When using the "usetmp" option, this option allow setting the prefix to use for the temporary file name (e.g. ".").
ftp.suffix	-	string	When using the "usetmp" option, this option allow setting the suffix to use for the temporary file name (e.g. ".tmp").
ftp.postConnectCmd	-	string	Allow issuing an FTP command on the remote FTP server right after the control connection is established (multiple commands can be specified using the semi columns separator).
ftp.preCloseCmd	-	string	Allow issuing an FTP command on the remote FTP server right before closing the control connection (multiple commands can be specified using the semi columns separator).
ftp.preGetCmd	-	string	Allow issuing an FTP command on the remote FTP server right before getting a file (multiple commands can be specified using the semi columns separator).
ftp.prePutCmd	-	string	Allow issuing an FTP command on the remote FTP server right before putting a file (multiple commands can be specified using the semi columns separator).
ftp.postPutCmd	-	string	Allow issuing an FTP command on the remote FTP server right after putting a file (multiple commands can be specified using the semi columns separator). The parameter "\$filename" can be introduced in the commands and it gives the name of the file which was just transmitted. If the "usetmp" option is used then it will give the temporary name. e.g. ftp.postPutCmd="SITE CHMOD 0644 \$filename"
ftp.mkdirCmdIndex	"0"	integer	When using the "preMkdirsCmd" and "postMkdirsCmd" options, this index allow selecting the directories for which a command should be triggered. For example, if the directory is "/home/uid/test/data/out/bin", an index of 3 will make sure the FTP client start the "preMkdirsCmd" and/or "postMkdirsCmd" commands for the "home", "uid" and "test" directories. If the index is negative then the selection start from the end of the path instead of the beginning. In the previous example it would make sure the FTP client start the "preMkdirsCmd" and/or "postMkdirsCmd" commands for the "data", "out" and "bin" directories. e.g. ftp.mkdirCmdIndex="-3"
ftp.preMkdirsCmd	-	string	Allow issuing an FTP command on the remote FTP server right before creating a directory (multiple commands can be specified using the semi columns separator). The parameter "\$dirname" can be introduced in the commands and it gives the name of the directory to create (including all the parent directories).
ftp.postMkdirsCmd	-	string	Allow issuing an FTP command on the remote FTP server right after creating a directory (multiple commands can be specified using the semi columns separator). The parameter "\$dirname" can be introduced in the commands and it gives the name of the directory just created (including all the parent directories). e.g. ftp.postMkdirsCmd="SITE CHMOD 0755 \$dirname"
ftp.keepAlive	"0"	milliseconds	Allow keeping FTP control connections alive in a pool. If the connection is unused for longer than the duration specified by this option then the connection is closed and removed from the pool. A duration of zero is interpreted as no pooling.
ftp.useNoop	"0"	milliseconds	When the "keepAlive" option is configured, then this option allow sending NOOPS commands to the remote server in order to keep the connection alive. The value of this option specify the interval between 2 sending of the NOOPS command. A value of 0 deactivate the sending of the NOOPS command. A custom NOOPS command can be specified with the "setNoop" option.
ftp.ignoreCheck	"true"	boolean	If set, the remote size of the file is not checked after an upload has been completed. If not set, the remote size of the uploaded file is checked and compared against the size of the source file.
ftp.ignoreDelete	"true"	boolean	If not set, the FTP client tries to delete the target file before the upload is processed.
ftp.deleteOnRename	"true"	boolean	If set, the FTP client tries to delete the target file before doing a rename (e.g. when using temporary file names with the "usetmp" option).

ftp.retryAfterTimeoutOnCheck	"false"	boolean	If set, the FTP client will try to reconnect if the control connection has timed-out during the data transmission.
ftp.keepControlConnectionAlive	"false"	boolean	If set, the FTP client will keep sending NOOPS commands through the control channel while waiting for a data transmission to complete.
ftp.mksuffix	"false"	boolean	If set, the FTP client will generate a random suffix for the "usetmp" option with 3 characters (e.g. ".3te"). If this option is used then the "prefix" and "suffix" options are ignored. The "usesuffix" option can be used to customize the created suffix.
ftp.usesuffix	"false"	boolean	If set and the "mksuffix" is set as well then the suffix specified in the "suffix" option will be concatenated to the generated suffix (e.g. ".3te.tmp").
ftp.cwd	-	string	Allow changing directory just after the login (a "cd" command is issued with the specified directory).
ftp.nopassword	"false"	boolean	Allow connecting to a remote FTP server with no password required (e.g. with anonymous users).

The genericSftp Options

Name	Default	Type /Unit	Comment
sftp.usetmp	"true"	boolean	Force using a temporary name when transmitting a file to the remote SFTP server. Once the transmission has completed successfully, the file is renamed with its final target name. The temporary file name is by default the file name with the ".tmp" suffix concatenated to it, however this behaviour can be customised with the "mksuffix", "prefix" and "suffix" options.
sftp.mkdirs	"true"	boolean	Allow creating the directory named by the target pathname, including any necessary but non-existent parent directories.
sftp.prefix	-	string	When using the "usetmp" option, this option allow setting the prefix to use for the temporary file name (e.g. ".").
sftp.suffix	-	string	When using the "usetmp" option, this option allow setting the suffix to use for the temporary file name (e.g. ".tmp").
sftp.chmod	-	integer	If set, the file mode will be changed according to the value of this option once it has uploaded a file successfully (e.g. "640").
sftp.ignoreCheck	"false"	boolean	If set, the remote size of the file is not checked after an upload has been completed. If not set, the remote size of the uploaded file is checked and compared against the size of the source file.
sftp.execCode	"0"	integer	Allow specifying the expected exit code of the command (or list of commands) started with the "execCmd" option. If the exit code is different from the one specified in this parameter then the transmission is tagged as failed.
sftp.execCmd	-	string	Allow specifying a command (or list of commands) to execute on the SSH server once a file has been transmitted (multiple commands can be specified using the semi columns separator). The parameter "\$filename" can be introduced in the commands and it gives the name of the file transmitted (source or target depending if the transmission was a upload or a download).
sftp.kex	-	string	Allow specifying the key exchange algorithms to allow (e.g. "ecdh-sha2-nistp256,ecdh-sha2-nistp384"). The list of valid algorithms can be found at http://www.jcraft.com/jsch/README (multiple algorithms can be specified using the columns separator). By default all supported algorithms are allowed.
sftp.serverHostKey	-	string	Allow specifying the host key type to allow (e.g. "ssh-dss,ssh-rsa,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521"). The list of valid key types can be found at http://www.jcraft.com/jsch/README (multiple algorithms can be specified using the columns separator). By default all supported key types are allowed.
sftp.cipher	-	string	Allow specifying the cipher algorithms to allow (e.g. "aes128-cbc,3des-cbc,blowfish-cbc"). The list of valid algorithms can be found at http://www.jcraft.com/jsch/README (multiple algorithms can be specified using the columns separator). By default all supported algorithms are allowed.
sftp.mac	-	string	Allow specifying the Message Authentication Code (MAC) algorithms to allow (e.g. "hmac-md5,hmac-md5-96,hmac-sha1,hmac-sha1-96"). The list of valid algorithms can be found at http://www.jcraft.com/jsch/README (multiple algorithms can be specified using the columns separator). By default all supported algorithms are allowed.
sftp.compression	-	string	Allow specifying the compression algorithms to allow (e.g. "zlib,none"). The list of valid algorithms can be found at http://www.jcraft.com/jsch/README (multiple algorithms can be specified using the columns separator). By default all supported algorithms are allowed.
sftp.fingerPrint	-	string	Allow specifying a key's fingerprint. It is a lower-case hexadecimal representation of the MD5 of a key (e.g. "22:fb:ee:fe:18:cd:aa:9a:9c:78:89:9f:b4:78:75:b4"). If a key's fingerprint is defined then it is compared against the key's fingerprint of the remote SFTP server when connecting. If it does not match then the connection is rejected.
sftp.preferredAuthentications	-	string	Allow specifying the list of supported authentication methods (e.g. password,publickey,keyboard-interactive).
sftp.passPhrase	-	string	The passphrase necessary to access the private key when the "privateKey" or "privateKeyFile" option is used. This option is only required if the private key is protected.

sftp.privateKeyFile	-	string	<p>Adds a private key to be used for public-key authentication. The private key should be in the PEM format in a file accessible from the ECaccess Gateway (the file permissions should be set accordingly). This option contains the name of the file (e.g. "/home/uid/.ssh/mykey.pem").</p> <p>If both options, "privateKeyFile" and "privateKey" are defined, then the latest will be ignored.</p>
sftp.privateKey	-	string	<p>Adds a private key to be used for public-key authentication. The private key should be in the PEM format.</p> <pre>e.g. sftp.privateKey=" -----BEGIN DSA PRIVATE KEY----- Proc-Type: 4,ENCRYPTED DEK-Info: DES-EDE3-CBC,2AEF2CF4618CA866 HFYVuVUSTdJacwFiaTcY3I5jUHaviYdDQHIzarzox/ikWC6rN7uJbwPrZZPH+WV1 PfwX9yC7jVlUSgPu7o8fSdEhBaPZQ0NZ7Gq3cOpsRYicRa3F4GebK748m8VZLXLe (...) b4DxuN8evYgV/M9V76JoW5TDiizgst86w915bjM/gl+mVPPH/XitSOpaAdtHSpDv ph1kgY1fjwT51e+i66AnLmbJ4cLLAMz3 -----END DSA PRIVATE KEY----- "</pre> <p>As shown in the example above, the first line should always be sftp.privateKey= and the last line should always be " with the content of the private key in between.</p>
sftp.listenAddress	-	string	<p>Allow specifying the source address used by the SFTP client when connecting to the remote SFTP server. By default, the system will pick up a valid local address.</p>
sftp.sessionTimeout	"60000"	milliseconds	<p>This value is used as the socket timeout parameter, and also as the default connection timeout. The value must be > 0. A timeout of zero is interpreted as an infinite timeout.</p>
sftp.connectTimeout	-	milliseconds	<p>This value is used as the connection timeout. The value must be > 0. A timeout of zero is interpreted as an infinite timeout. The default value is the same as the "sessionTimeout" value.</p>
sftp.mkssuffix	"false"	boolean	<p>If set, the SFTP client will generate a random suffix for the "usetmp" option with 3 characters (e.g. ".3te"). If this option is used then the "prefix" and "suffix" options are ignored.</p>
sftp.port	"22"	short	<p>Allow specifying the port number of the remote SFTP server to connect to.</p>

The genericExec Options

The purpose of this ECtrans Module is to transmit a file on the file system of the ECaccess Gateway and start a script after the transmission has completed successfully. However, this module has several limitations as the script can only be started by the user-identifier used to run the ECaccess Gateway. If a user is required to start a script after a file has transferred successfully using its own user-identifier, then the "genericSftp" module should be used instead, with the help of the "execCmd" option. An alternative would be to perform a "su - \$ecuser" in the script body (providing the "\$ecuser" uid exists on the system, otherwise a mapping between ECMWF users and local users could be implemented in the script).

Most of the options can only be set in the configuration file of the ECaccess Gateway, the "ecmwf.properties" file. Indeed, anyone with an ECtrans Association could virtually start any command on the server hosting the ECaccess Gateway, which could lead to a security breach.

The only one parameter which can be customized through the ECtrans Association is the following:

Name	Default	Type /Unit	Comment
returnCode	"0"	integer	<p>Allow setting the expected exit code of the script started with the "script" option specified in the ECaccess Gateway configuration file. If the exit code is different from the one specified in this parameter then the transmission is tagged as failed. This option is only used in the "synchronous" mode (see the table below), otherwise the script is started in the background and ECtrans does not wait for the exit code to return.</p>

The following options can be set in the "ecmwf.properties" file, in the "ExecModule" group of parameters:

Name	Default	Type /Unit	Comment
returnCode	"0"	integer	<p>Allow specifying the expected exit code of the command started with the "script" option. If the exit code is different from the one specified in this parameter then the transmission is tagged as failed. This value can be overwritten in the ECtrans Association. This option is only used when the "synchronous" option is set to "true", otherwise the script is started in the background and ECtrans does not wait for the exit code to return.</p>
script	\${ecmwf.dir}/gateway/bin/ectrans	string	<p>Allow specifying the script to execute on the ECaccess Gateway once a file has been transmitted.</p>
dir	-	string	<p>Sets the working directory for the data transmission.</p>
synchronous	"true"	boolean	<p>If set then the script is started and ECtrans wait for the exit code to return, If not set then the script is started in the background and ECtrans does not wait for the exit code to return.</p>

usePipe	"false"	boolean	If set then the arguments (parameters) of the script are sent one per line to the standard input stream of the process running the script. If not set then the arguments (parameters) are passed to the script through the command-line. This option can be useful to hide the arguments passed to the script, which could be seen by a "ps" command started on the system.
args	-	string	<p>This is the string which is passed to the script for execution.</p> <p>e.g. <code>args="-user \$ecuser -filename \$target"</code></p> <p>The args string can take the following parameters:</p> <ul style="list-style-type: none"> • <code>\$msuser[name]</code>: name of the association. • <code>\$msuser[comment]</code>: comment. • <code>\$msuser[dir]</code>: target directory. • <code>\$msuser[host]</code>: target host. • <code>\$msuser[login]</code>: login identifier. • <code>\$msuser[passwd]</code>: password. • <code>\$msuser[password]</code>: alias to <code>\$msuser[passwd]</code>. • <code>\$ecuser[name]</code>: ECMWF user-name. • <code>\$ecuser[uid]</code>: ECMWF user-identifier. • <code>\$ecuser[gid]</code>: ECMWF group-identifier. • <code>\$ecuser[dir]</code>: ECMWF home directory. • <code>\$ecuser[shell]</code>: ECMWF login shell. • <code>\$ecuser[comment]</code>: ECMWF comment. • <code>\$ecuser</code>: alias to <code>\$ecuser[name]</code>. • <code>\$target</code>: target file name (absolute path). • <code>\$dir</code>: working directory as defined in the "dir" option. • <code>\$location</code>: alias to <code>\$dir</code>. • <code>\$password</code>: alias to <code>\$msuser[passwd]</code>.
mkdirs	"true"	boolean	Allow creating the directory named by the target pathname, including any necessary but non-existent parent directories.
usetmp	"true"	boolean	Force using a temporary name when transmitting a file to the ECaccess Gateway. Once the transmission has completed successfully, and before the script is started, the file is renamed with its final target name. The temporary file name is by default the file name with the ".tmp" suffix concatenated to it, however this behaviour can be customised with the "prefix" and "suffix" options.
prefix	-	string	When using the "usetmp" option, this option allow setting the prefix to use for the temporary file name (e.g. ".").
suffix	-	string	When using the "usetmp" option, this option allow setting the suffix to use for the temporary file name (e.g. ".tmp").

These options can only be modified by the administrator of the ECaccess Gateway.

ectrans command

With the ectrans command, Member State users can initiate secure file transfers between ECMWF (ecgate or HPCF systems) and a Member State server, as defined in the ectrans association. When ectrans is used to put a file (from ECMWF to a remote server), the ECaccess Server will spool the file in the user's "ectrans" transfer queue. If the connection between the ECMWF and the remote gateway is down or if any error occurs, the file will be kept in the spool area at ECMWF and you can resume the transfer either through the web interface or with the Web Toolkit command `ecaccess-ectrans-restart`. When ectrans is used to get a file (from a remote server to ECMWF) the transfer will fail by default, if the connection between the ECMWF and the remote gateway is down. A retry mechanism is available for all types of transfers. To show the ectrans usage:

```
> ectrans -help
usage: ectrans [-gateway name] -remote association@[protocol] \
    [-get|-put] -source [ec:[ectmp:]filename [args ...]] (*)
    ectrans -check requestID (*)

-gateway {arg} - access gateway name (default (**): boaccess.ecmwf.int)
-remote {arg} - association and (optional) protocol (default (**): *none*)
-source {arg} - source file name
-target {arg} - target file name (default: same as -source)
-mailto {arg} - target email address (default: current user)
-lifetime {arg} - lifetime of the file in the spool (default: 1w) (***) (****)
-delay {arg} - transmission delay (default: immediate transfer) (***) (****)
-at {arg} - transmission date (default: immediate transfer) (****)
-format {arg} - define the date format as used with -at (default: yyyyMMddHHmmss)
-retryCnt {arg} - define the number of retries (default: async=144, sync=0)
-retryFrq {arg} - define the frequency of retries (default: async=10m, sync=1m) (****)
-maxTime {arg} - define the maximum transfer duration (default: 12h) (****)
-priority {arg} - transmission priority 0-99 (default: 99) (****)
-put - interactive/synchronous transfer (no spool)
-get - interactive/synchronous pull (rather than push) file
-onsuccess - mail sent on successful transfer
-onfailure - mail sent when transfer has failed
-onretry - mail sent when transfer is retried
-keep - keep the request in the spool till expiration (****) (*****)
-remove - always remove the request from the spool (****) (*****)
-reject - if existing target file (default)
-append - if existing target file
-resume - if existing target file
-overwrite - if existing target file
-verbose - verbose mode on
-version - print version number
-help - this message

(*) If successful, a requestID is returned, which can be used in
    check requests. Exit code is 0 on success and >0 otherwise.
(**) The default values depend on the GATEWAY or REMOTE environment
    variables.
(***) Duration in weeks, days, hours, minutes or seconds (e.g. 1w|2d).
(****) These options are only relevant when the spool is used. The spool
    is no used during interactive transfers (-get and -put options).
(*****) By default, successful requests are removed from the spool and
    failed requests are kept in the spool till expiration.
```

The "reject", "append", "resume" and "overwrite" options are mutually exclusive and determine what to do if there is an existing target file. The "mailto" option specifies an email address to be notified in case of a successful (option "onsuccess") and/or a failed transfer (option "onfailure"). The "check" option prints the status of the specified request on the standard output. The transfer status, which can be checked with the `ecaccess-ectrans-list` command or the Web interface, can takes values as listed in table 1. Table 1: Transfer status.

Status	Meaning
INIT	Files are being transferred to the spool
COPY	Files are being transferred to the remote site
WAIT	Files are scheduled and waiting to be started
RETR	File transfer will be retried
STOP	Files have NOT been successfully transferred (error)
DONE	Files have been successfully transferred

Transfer to a Member State host via gateway

To transfer file "fff" from the current working directory on "ecgate" to the "genericFtp" protocol of the use "myUser" on the ECaccess gateway "[ecaccess.meteo.ms](#)":

```

> ectrans -gateway ecaccess.meteo.ms \
          -remote myUser@genericFtp
          -source fff \
          -verbose
verbose: gateway=ecaccess.meteo.ms
verbose: echost=ecgate.ecmwf.int
verbose: eport=644
verbose: action=spool
verbose: ecuser=uid
verbose: source=fff
verbose: target=fff
verbose: keep=false
verbose: remove=false
verbose: option=reject
verbose: lifetime=1w
verbose: delay=(none)
verbose: at=(now)
verbose: format=yyyyMMddHHmmss
verbose: retryCnt=144
verbose: retryFrq=10m
File to upload (5140480 bytes)
9442903031

```

When a request has been spooled successfully, a requestID is returned immediately. ectrans will then return the exit code 0. The requestID can be used to reference the transfer, using the interface described in section 8 or with the command ecaccess-ectrans-list. If the file is not successfully spooled, an error message is printed and the ectrans return code is -1.

Transfer from a Member State host via gateway

To transfer file "fff" at the "genericFtp" protocol of the "myUser" association of the ECaccess gateway "[ecaccess.meteo.ms](#)" to the current directory at ECMWF:

```

> ectrans gateway ecaccess.meteo.ms \
          -remote myUser@genericFtp \
          -get -source fff \
          -verbose
gateway: boaccess.ecmwf.int
echost: ecgate.ecmwf.int
eport: 644
action: get
ecuser: uid
target: fff
source: fff
keep : false
option: reject
File to download (0 bytes)
5140480 bytes to download

```

When the request has been carried out successfully, the result is returned immediately. Transfers from a Members State to ECMWF are not spooled; they are carried out synchronously. The ectrans return code is 0 if the file has been transferred successfully or -1 if the file has not been transferred successfully.