

RMFT FASTCopy Administrator's Guide

RMFT Version 2.4.3

February 7, 2010

February 7, 2010
Copyright © 2000-2010 by RepliWeb, Inc.

This product includes cryptographic software written by Eric Young
(eay@cryptsoft.com)

The information in this document has been compiled with care, but RepliWeb makes no warranties as to accurateness or completeness, as the software described herein may be changed or enhanced from time to time. This information does not constitute commitments or representations by RepliWeb, and is subject to change without notice. The software described in this document is furnished under license and may be used or copied only in accordance with the terms of this license.

No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written consent of RepliWeb, Inc.

Any trademarks, trade names, service marks, or service names owned or registered by any other company and used in this document are proprietary to that company.

Please direct correspondence or inquiries to:

RepliWeb, Inc.
6441 Lyons Road
Coconut Creek
FL 33073

Phone: (954) 946-2274

Fax: (954) 337-6424

E-Mail: info@repliweb.com,

Support: <http://support.repliweb.com/>

Web Site: <http://www.repliweb.com/>

Table of Contents

Introduction	5
1. Security and Administration	6
Security Roles	7
Administrative Roles.....	8
The Security and Administration File Hierarchy	8
Security and Administration File Structure	12
The Requester Verb	15
The Criteria Verb.....	16
The Actions Verb	23
The Ifnot Verb	30
The Search Verb.....	31
The On Verb	33
Using Variables in Records	34
Implementing the Proxy Security Mechanism	36
Different Security Checks	36
The Login Security Check.....	36
Operations within a Closed Network.....	38
Data Transfer Firewall	39
Proxy Login with Logical Passwords	39
When the Network is Not Trusted.....	40
Securing Specific Resources.....	40
Control Of Outgoing Operations	41
Working with Groups	41
The sl_passwd Utility.....	43
SSL 3.0 Authentication	46
2. RMFT FASTCopy Receiving Agent Manager	49
Opening RMFT FASTCopy Receiving Agent Manager.....	50
Configuring RMFT FASTCopy Receiving Agent	51
General Tab	51
Security Tab.....	53
SSL Tab	55
Trace Tab.....	60

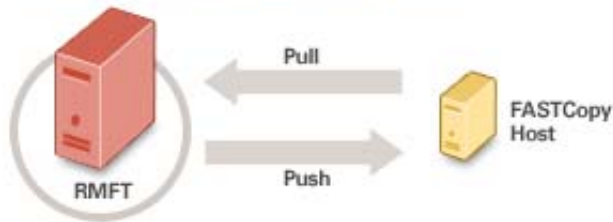
Managing RMFT FASTCopy Receiving Agent Nodes	61
Adding RMFT FASTCopy Receiving Agent Nodes	61
Secure Login Options	64
3. Starting RMFT FASTCopy Receiving Agent on UNIX	68
Starting fcopyd on UNIX	68
Enabling fcopyd through inetd	68
Starting fcopyd Directly as a Standalone Daemon	69
xinetd Support for RedHat Linux 7.x.....	69
Example of RedHat 7.x FASTCopy Installation Prompts	69
<i>fcopyd</i> Qualifiers	70
A. Loading Certificates on UNIX/Linux Platforms	71
B. Importing MSCAPI Server Certificates	75
Importing the Server Certificate	75
Importing the CA Certificate.....	82
INDEX	84

Introduction

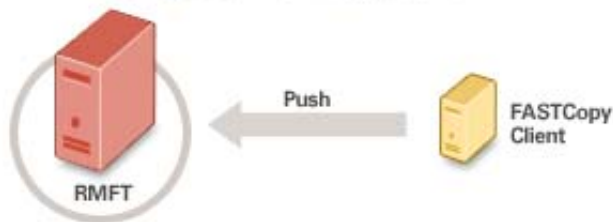
RMFT's EFT FASTCopy protocol is engineered for WAN transfers, where guaranteed delivery and file integrity are business-critical requirements. FASTCopy-enabled transfers offer encryption, block-level recoverability, data integrity assurance, and CRC Checksums. Full bandwidth control, compression and pre-and-post-processing make sure that transfers are accomplished as quickly and efficiently as possible.

In a full installation of RMFT Server, both RMFT FASTCopy Sending Agent and RMFT FASTCopy Receiving Agent are installed, enabling RMFT Server to act as both a FASTCopy Client and a FASTCopy Server. In other words, in addition to being able to pull and push files from/to Hosts (i.e. initiate FASTCopy transfers), RMFT Server can also receive files sent by FASTCopy Clients using the FASTCopy command line.

RMFT Acting as a FASTCopy Client



RMFT Acting as a FASTCopy Server



To enable SSL transfers, you will need to configure RMFT FASTCopy Receiving Agent to load the appropriate certificates by performing the procedure(s) described in [SSL Tab](#).

Note: For instructions on installing RMFT FASTCopy Receiving Agents, please refer to the *RMFT FASTCopy Installation Guide (Windows)* and the *FASTCopy Installation Guide for UNIX*.

1. Security and Administration

Administrators can control RMFT FASTCopy operations through RMFT FASTCopy's **Proxy Security Mechanism**. Whenever a FASTCopy *operation* is initiated, it activates a security check which compares the details of the requested operation against a rule base stored in one or more *Security and Administration Files* ("security files" for short).

By default, these files are located in:

```
~\RepliWeb\RMFT\security
```

Based on the administrator-defined rules recorded in this file, the operation is approved, refused or modified.

A security check is carried out in the following cases:

- When a local user or application issues a FASTCopy command.
- Whenever RMFT FASTCopy Receiving Agent receives a remote request for a FASTCopy operation involving the local node.
- Whenever the *fmonitor* command is issued.
- Whenever a monitor message is received from a monitored FASTCopy operation.
- Whenever the *fcmon* command is issued.

Note: In most cases, the operation must pass two security checks - one on the node of origin and one on the peer node; if it fails to pass either one, it will be aborted.

FASTCopy's Proxy Security Mechanism plays two major roles:

1. It is an administrative tool, allowing you to control how users use FASTCopy and its related programs on your local system.
2. It is a security tool, allowing you to control all incoming FASTCopy requests from outside your system. The Proxy Security Mechanism serves here as a "data transfer firewall", allowing remote users access without exposing your system.

To understand how the Proxy Security Mechanism carries out these functions, it is useful to know how it performs the security check.

For each requested operation, a number of parameters are examined during a security check. These include the requester's node, user name and network address, the local user name and password given for the operation, the type of operation or command, the files and directories affected, the time of day, and any special codes or identifying signs used in the operation. Any of these properties can be used as criteria to determine if a given rule should apply to the operation. A matching rule can instruct the checking application to refuse or approve the requested operation.

The rules can also tell the application to modify the operation's properties, such as changing the local user account the operation is carried under, the directories the operation acts on, or a transfer's bandwidth limit.

Note: The “checking application” referred to above can be FASTCopy (for FASTCopy operations initiated by local users), RMFT FASTCopy Receiving Agent *fcopyd* (for remote operations involving the local node) or the batch daemon *flogicd* (when a batch operation is submitted or an *fmonitor* command is used to examine or change a batch job's status).

Security Roles

The Proxy Security Mechanism operates on the principal of Active Security: Instead of passively relying on the system's built-in security measures, it actively takes any information it has on the requested operation and decides if it should be refused, approved or modified, based on any relevant rules defined in one or more security files. Because of this, security is not based on what the remote user knows about a system (like *ftp* or *rcp* passive security), but on what the administrator actively allows to be done on the system through the security rule base.

This means that sensitive system information such as user account names or passwords does not have to be given to remote users or passed over the network. The administrator decides exactly who is allowed to connect from where, what authentication process they must pass, and what exactly they will or will not be allowed to do.

Active Security can be used to limit access based on remote user's name, remote node's name, network address, and other identifiers. Complete sub-networks and user groups can be identified and restricted in a single rule.

Data transfer to or from a node can be restricted with regards to read and write access to specific directories and sub-directories, the size of files received, permitted time of day for upload or download and remote activation of local images (using FASTCopy's post-transfer processing features). The administrator can use it to divert remote users to specific local accounts, regardless of information they might possess. Logical passwords and user names can be created, which can then

be diverted to real system user accounts. In addition, remote users can be required to use a message authentication protocol, which verifies the identity of both sides and “signs” each transferred packet to prevent tampering, and a data-encryption method, which hides the content of files as they pass over the network.

FASTCopy also provides file and directory level security files, which can restrict or allow remote user access to specific resources in the file system. These security files are described in the next section.

Administrative Roles

Since the security check is also performed on local FASTCopy activities, any of the above restrictions can be placed on local as well as remote users. The same mechanism can be used for administration as well as security purposes, such as improving the organization of distributed FASTCopy activities within a networking environment. In addition, the security mechanism can be used to enforce certain requirements or restrictions on outgoing operations. For example, you can limit the bandwidth of any locally-initiated FASTCopy operation, cause it to use a specific encryption method on any files transferred or automatically generate reports to a central monitoring point.

The following sections detail the different types of security files that can be created, how such a file is constructed and how a *Security And Administration File* is used to control different kinds of file transfer activities.

The Security and Administration File Hierarchy

The FASTCopy Proxy Security Mechanism's rule base is stored in one or more security and administration files. Each node has at least one such file called **softlink.security**, setting general file-transfer permissions for that node. On UNIX systems, this file is located in the `/etc` directory. The default location of this file on the RMFT Server machine or on RMFT FASTCopy Agent machines (Windows) is:

```
~\RepliWeb\RMFT\ssl
```

In this document, the **softlink.security** file is referred to as the *Security and Administration File*. This file should be readable to all users (since it is examined by FASTCopy processes executed from different local user accounts) but only the administrator or Super user (root) should have write privileges to this file.

The Proxy Security Mechanism also searches for security and administration rules in additional files (referred to in this documentation simply as “security files”).

There are two kinds of auxiliary security and administration files:

1. A “login” security file for the node. This file contains rules that are used to determine if there is a local user under which a remotely initiated operation can be carried out. This file is examined only once during an operation, when the operation's local context is set. During this check, *RMFT FASTCopy Receiving Agent* uses only the rules in this file and the main Security and Administration File. A login file is especially important if remote users connect to the node from outside the trusted environment, for example over the Internet rather than within a closed network. It should contain all the sensitive information used to set up a secure session, such as encryption keys, mutually agreed upon codes and logical passwords. The login security file is called **softlink.login**, and is located in the same directory as the main Security and Administration File. This file must be readable only by the administrator or super user (mode 600 on UNIX), or FASTCopy will issue an error message.
2. A security and administration file can be created for any given file or directory. This security file contains rules which govern the transfer of the matching file or directory. Whenever *RMFT FASTCopy Receiving Agent* responds to a remote request to pull a local file or push files into a local directory, it applies the relevant rules in the file or directory's associated security file, and in the security files of all the directories in the path, as well as any appropriate rules in the main *Security and Administration File*. Using file and directory level security files provides more finely-grained control over file-transfer operations, and can be used to delegate some specific file transfer security decisions to users, rather than leaving them all in the hands of the administrator.

A file's corresponding security file must be in the same directory as the file.

On UNIX and Windows systems, file-level security files are called *.datafilename.sl_sec*, where *datafilename* is the name of the data file that the security file is associated with, including any extensions.

A directory-level security file is a file inside the relevant directory called *.sl_sec*.

Note: A directory can contain both a file-level security file inside the parent directory, and a directory-level security file, inside the directory itself. For example, the UNIX directory `/usr/joe/docs` can contain both a file-level security file called `/usr/joe/.docs.sl_sec`, and a directory-level security file called `/usr/joe/docs/.sl_sec`.

Both file and directory level security files should be readable from any local user account.

```
linux:/usr/joe%>ls -laR
total 13
drwxr-xr-x  3  joe  agroup 2048 Mar 24 11:37 ./
drwxr-xr-x 25  root  root   1024 Mar 18 11:34 ../
-rw-r.r.r.  1  root  root    42 Mar 24 11:37
.docs.sl_sec
-rwx.x.x  1  joe  agroup 5882 Mar 12 12:22 b.b*
drwxr-xr-x  2  root  root   1024 Mar 24 11:38 docs/
docs:
total 4
drwxr-xr-x  2  root  root   1024 Mar 24 11:38 ./
drwxr-xr-x  3  joe  agroup 2048 Mar 24 11:37 ../
-rw-r.r.r.  1  root  root    17 Mar 24 11:38 .sl_sec
linux:/usr/joe%>
```

Whenever a file is transferred, FASTCopy takes into account rules in all the security files relevant to that file. These include the main *Security and Administration File*, the transferred file's own security file, and any security files associated with the directories in its path. In general, specific rules (for example those in the security file associated with the file itself) will override more generic ones (for example, those defined in the directory's security file or in the main *Security and Administration File*). However, a "general" file can instruct the checking application to ignore the rules in more "specific" files. This instruction will cause the checking application to ignore all the rules in more specific files, EXCEPT for those that tell it to refuse an operation that was approved in the more general file.

For example, during a FASTCopy operation that transfers a file called **/usr/joe/docs/fcopy_manuals.ps.tar.Z** (on a UNIX system), a security check will be carried out that can potentially include rules in any of the following files (ordered here from the most general to the most specific):

File	Description
/etc/softlink.security	The main Security and Administration File
/.sl_sec	Directory-level security file
/.usr.sl_sec	File-level security file for a directory
/usr/.sl_sec	Directory-level security file
/usr/.joe.sl_sec	File-level security file for a directory
/usr/joe/.sl_sec	Directory-level security file
/usr/joe/docs.sl_sec	File-level security file for a directory
/usr/joe/docs/.sl_sec	Directory-level security file

<code>/usr/joe/docs/fcopy_manualse.ps.tar.Z.sl_sec</code>	File-level security file for the file itself
---	--

In the above example, rules in the file `/usr/joe/docs/.sl_sec` can override rules in the file `/usr/joe.sel_sec`, and in turn be overridden by rules in the file `/usr/joe/docs/fcopy_manualse.ps.tar.Z.sl_sec`. A security file higher in the file hierarchy can include a rule that will cause FASTCopy to disregard security files below it.

For example, if the file `/usr/.sl_sec` includes such a rule, FASTCopy will ignore the rules in the following files:

```
/usr/.joe.sl_sec, /usr/joe/.sl_sec,  
/usr/joe/docs.sl_sec,  
/usr/joe/docs/.sl_sec  
/usr/joe/docs/fcopy_manualse.ps.tar.Z.sl_sec.
```

It will only act upon a rule in any of these files if it specifies that an operation that would otherwise be approved should be refused.

A further discussion of the auxiliary security and administration files can be found after the next section, which describes the contents of a security file, and how security and administration rules are constructed.

Security and Administration File Structure

General Syntax

Both the main and the auxiliary *Security and Administration Files* share a common structure and syntax. Each file is made up of records, which serve as a set of rules.

Each record consists of several lines describing how to respond to a request matching the details of the record. A separator line, which must begin with a “=” symbol, is used to separate a record from the one following it. As RMFT FASTCopy Receiving Agent reads the security file, it uses the separator lines to identify all the text before them (and after the previous separator line) as a record it must examine. It ignores the entire separator line, so the line can be used for comments, for example identifying the next record. Comments can also be placed in lines beginning with a mark, which FASTCopy also ignores.

For example, here's a simple security file, which contains two records, a separator line, and a comment:

```
#On this node you can send files to other nodes:
requester
criteria -operation=data_out
actions -approve
== But you can receive files only from Bill at the "server"
node: ==
requester -node=server -user=Bill
criteria -operation=data_in
actions -approve
```

Note: The last record in a file does not have to be followed by a separator line.

Record Structure

Each line in a record begins with a verb, which is usually accompanied by one or more qualifiers. Each qualifier is preceded by a dash (-) and followed by a value. The verb provides the line's context within the record, while the qualifiers and their values provide the line's actual content. You may substitute any non-ambiguous abbreviation for any verb or qualifier. Any value string which is broken up by spaces or commas (,) must be enclosed in quotation marks (“”).

The example above contained two records. In both, the first line begins with the verb **requester**, the second line begins with the verb **criteria** and the last line starts with the verb **actions**. Both records act in an identical manner: They specify that if the request comes from someone identified in the **requester** line and matches the specified **criteria**, FASTCopy should act according to what is in the **actions** line.

These are the verbs which can be used in a security file record:

- **requester** - The details in this line identify the initiator of the request. If it is used without identifying qualifiers, the requester can be anyone.
- **criteria** - Lines with this verb are used to filter requests according to their particulars. If a record includes no criteria lines, it applies to any request matching the requester's details.
- **actions** - This verb specifies how the request should be responded to if it matches the specified criteria, and if it should be approved or refused.
- **ifnot** - This verb is similar to the actions verb, in that it specifies a response to a request, but the actions it specifies are to be taken only if the specified criteria are not met.
- **search** - This verb is used to provide instructions on how to continue reading the security file after finding that this record matches the request.
- **on** - This verb specifies on which node the rule should apply.

A record must include at least one **requester** and one **actions** line. It can include multiple lines of **requester**, **criteria** or **actions**. All the actions specified in the **actions** line of a record will apply to a request if it matches one of the record's **requester** lines and one of the **criteria** lines.

Parsing the Security File

Whenever a request causes a security check to be run, the security module goes through the security file and compares the details of the request to the requester and criteria lines in the different records. When the security module finds a match between the request and a record, it stores all the actions detailed in that record and continues on to the next record in the file. In this way, it checks the entire file and collects all the actions appropriate for the request.

When two records conflict, the later one takes precedence. For example:

```
requester -node=alf
actions -noapprove
==
requester -node=alf -user=joe
actions -approve
```

Even though the first record instructs FASTCopy to refuse all requests from the "alf" node, the second record overrides the first with regard to the user "joe".

However, if (as shown below) the order of the records was different, requests it from user "joe" on the "alf" node would be refused, since the second record, instructing FASTCopy to refuse all requests from the "alf" node, takes precedence.

```
requester -node=alf -user=joe
actions -approve
==
requester -node=alf
actions -noapprove
```

For this reason, when constructing a security file, general defaults should be listed first, with exceptions to these defaults listed below them.

You can change the way the security file is analyzed by the security module using the `search` verb, as explained under that verb's description.

The Requester Verb

Each record must contain at least one **requester** line. This line is used to identify the initiator of a FASTCopy request. A requester can be identified by node, user name, user group, or any combination of the three, using these qualifiers:

- **-node=node_name** - The node where the request was initiated (in some cases this may not be the node from which the request arrives). If you want to specify the local node, use the string “localhost” rather than the full node name.
- **-user=user_name** - The name of the user issuing the request.
- **-group=group_name** - If an operation is initiated with the FASTCopy qualifier `-group`, the group name specified there is compared with the one specified here. See [Working with Groups](#) (41) below for details. The default group is users.

If **requester** is specified without any identifying qualifiers (as in the first record above), the record will apply to all requesters.

You can specify multiple **requester** lines in a record. For the **actions** specified in a record to apply to a given request, the request must match all the conditions in at least one of the requester lines.

EXAMPLE

The record shown below will carry out the specified action if the requester is anyone from the node “sunny”, or if the requester is the user “Bill” or the user “Sam” from the “central” node.

```
requester -node=sunny
requester -node=central -user=Bill
requester -node=central -user=Sam
actions -approve
```

The Criteria Verb

Lines beginning with the **criteria** verb are used to further describe the FASTCopy requests to which a given record applies, beyond the details given in the **requester** line(s). Criteria used to filter requests might include the type of operation attempted, such as incoming data transfer or remote activation of a local executable, the local user name given, the node or network the request is coming from (which in some cases may not match the requester's node), and whether the request has successfully passed an authentication procedure.

The **criteria** verb can be used with the following qualifiers:

-class=class

Can be used only in the main Security and Administration File or in file and directory-level security files; matches incoming operations that were assigned the specified `class` name during the login security check, by a record in the login security file that specified that name with the `actions` qualifier `class`. This criteria can be used to differentiate in later security checks between different classes of operation that were approved during the login security check.

-code=code

Matches operations that include the particular code phrase in their command line, specified with the FASTCopy qualifier `code`. The code is received at the target node after being scrambled with a one-way encryption function. During a security check, the code phrase specified in the security file is scrambled using the same function and the resulting value is compared to the code received from the requesting process.

-control

This qualifier matches operations where the requester wants to control local batch operations. This matches *fmonitor* commands that alter the status of batch jobs, such as `hold`, `resume`, `kill`, etc.

-hook=program

Specifies an external program that should be used to carry out additional security checks on the request. The operation matches the criteria if the external program returns an exit status of 1, and does not match it if the program returns an exit status of 0. The complete path to the program should be specified as the value of this qualifier. When a security check activates an external hook program, it sends it a set of parameters as command line arguments. These parameters are pairs of the form `key=value` and describe the requesting operation.

The parameters (which are the equivalents of other `criteria` qualifiers) include:

Parameter	Description
<code>l_node</code>	The local node that received the request.
<code>l_address</code>	The local node's address.
<code>r_node</code>	The requester's node.
<code>r_user</code>	The requester's user name (on the requester's node).
<code>p_node</code>	The peer node's name.
<code>p_address</code>	The peer node's address.
<code>t_node</code>	The operation's target node.
<code>t_user</code>	The operation's target user.
<code>t_group</code>	The operation's target group.
<code>l_user</code>	The local user name given by the requester.
<code>l_password</code>	The password given by the requester for the local node, scrambled by a one-way encryption function.
<code>l_appl</code>	The local application that received the request, usually FASTCopy.
<code>r_appl</code>	The remote application that gave the request, if any.
<code>operation</code>	The type of operation requested.
<code>object</code>	The object of the operation (identical to the object criteria qualifier in the security and administration file).
<code>code</code>	Any code phrase given by the requester with the FASTCopy qualifier code, scrambled by a one-way encryption function.
<code>flags</code>	Internal FASTCopy security check flags.

-incoming (negatable)

This qualifier describes a request by a remote user on any peer node to carry out a FASTCopy operation opposite the local node. Its negation, `noincoming`, covers all locally initiated operations.

-l_user=user_name

Identifies requests which specified this particular local `user_name` using the FASTCopy qualifier `user`.

-l_password

Specifies a request which includes a user name (specified with the FASTCopy qualifier `user`) that has a valid logical password on the local node (specified with the FASTCopy qualifier `password` in the requesting operation's command line). A logical password can be created using the `sl_passwd` utility. If the specified user name and password match a logical user name and password but not a system user name and password, the `actions verb` must be used to assign the requested operation to an existing local user account for any operation to be carried out.

-l_application=FASTCopy|FASTLOGIC

The local application that receives the request. The value is either `FASTCopy` for requests handled by FASTCopy or `fcopyd`, for example, if the request is for local or remote file transfer, or `FASTLOGIC` for requests handled by the batch daemon `flogid`, such as requests for submitting a FASTCopy batch operation or using the `fmonitor` program.

-operation=operation_type

The `operation` qualifier is used to describe a particular type of operation that is requested. Possible values for `operation_type` include:

data-in - The request is for data transfer to the local node. This applies both to remote FASTCopy operations pushing files to the local node and local FASTCopy operations which pull files from a remote source to the local node.

data-out - The request is for data transfer from the local node. This covers both remote FASTCopy operations which attempt to copy local files as well as local attempts to transfer local files to a remote target node with FASTCopy.

execute - The request is for local task activation. This can either be by a peer node requesting the execution of a remote command on the local node, or a local operation using the `local_command` or `exit_command` qualifiers.

submit - The request is for the submission of an operation to the batch daemon. This covers any FASTCopy operation issued on the local node with the batch qualifier.

cmd-in - The request is to perform an `fmonitor` command on the local node.

cmd-out - The request is to perform an `fmonitor` command on a peer node.

monitor - The request is to send a monitor report to the local node. This covers any report sent from a monitored node by a FASTCopy operation that specifies the local node as the operation's monitoring node, using the `mon_node` qualifier. Monitor reports can be passed on from one monitoring node to another, so that the requester node is not necessarily the peer node. To avoid monitoring loops, no node will accept a monitor report from a remote node if it (the local node) is designated as the requester node.

-object=objectname

This qualifier matches a property associated with the requested operation. The exact object depends on what the operation is: For `fmonitor` commands, the object is the command's name (i.e. `list`, `hold`, etc.); for an incoming FASTCopy operation initiated on a peer node, the object is the remote file specification given in the command line, which specifies files on the local node (The node where the security check takes place). There are some contexts in which a requested operation has no object associated with it, for example, when an outgoing FASTCopy operation is initiated locally.

-outgoing (negatable)

This qualifier describes a request by a local user to carry out a FASTCopy operation between this and any peer node. It does not cover local requests to submit or monitor a batch operation, or remote (incoming) requests for FASTCopy operations. Its negation, `nooutgoing`, covers both incoming FASTCopy requests and local requests for batch submission and monitoring.

-password (negatable)

Specifies a request which includes a valid local user name with its correct password. By default, only remote requests of this type are approved. The negation, `nopassword`, identifies requests which did not include a local user name and password using the FASTCopy qualifiers `user` and `password`.

-peer_address=address

Specifies the address of the node from which the request arrives. Combined with a net mask, it can be used to designate an entire network of peer nodes. Like the peer node's name, its address can also be authenticated. The local node's address can be specified using the loopback address string `127.0.0.1` rather than the full node address.

-peer_net_mask=net_mask

Combined with an address string, this is used to designate a set of addresses. This allows you to use a single `criteria` line to specify a group of nodes, or an entire network. Each “zero” value bit in the net mask string matches any value of the corresponding bit of the requester's address, while any “one” bit in the net mask requires the corresponding bit in the requester's address to match the specified `peer_address`.

-peer_node=node_name

Specifies the node opposite which the operation will be carried out. When using FASTCopy, this may be the same as the `requester` node. The `peer_node` qualifier is important because the identity of a peer node can be verified by an authentication method to avoid impersonation. If you want to specify the local node, use the string `localhost` rather than the full node name.

-purge

This qualifier matches operations where the requester wants to purge local batch jobs. This matches the `fmonitor purge` command.

-r_application=FASTCopy|FASTLOGIC

The remote application that issues the request. The value is either `FASTCopy` for requests handled by FASTCopy or `fcopyd`, for example, if the request is for local or remote file transfer, or `FASTLOGIC` for requests handled by the batch daemon `flogiced`, such as requests for submitting a FASTCopy batch operation or using the `fmonitor` program.

-read

This qualifier matches operations where the requester wants to read local files, either to copy them to a remote node or to view them with an application. This matches file transfer operations where files are pulled from the local node (`operation=data-out`) or *fmonitor* commands where the command is `list`.

-target_group=group_name

This qualifier is used to address FASTCopy operations submitted for batch execution under the specified group name using the FASTCopy `group` qualifier. See [Working with Groups](#) (41) below for details.

-target_node=node_name

This qualifier is used to address FASTCopy operations submitted for batch execution on the specified node.

-target_user=user_name

This qualifier has two different meanings, depending on the request. It will match FASTCopy operations submitted for batch execution under the specified *user_name*. In addition, in the context of an incoming FASTCopy request, it matches the user name specified in the operation's command line. This reserves the log-in information given by the requester, even if a security file record changed the local user under which the operation is carried out.

-time=time_frame

This qualifier identifies a request by the time it arrives at the local node. It allows you to alter the node's response to FASTCopy requests over time. See *Setting Time Frame* in the *FASTCopy Reference Guide* for details.

-write

This qualifier matches operations where the requester wants to write local files. This matches file transfer operations where files are transferred to the local node (`operation=data-in`).

Lines beginning with the `criteria` verb can be used to define a request in great detail. A given record can include multiple `criteria` lines or none at all. Each line can contain several qualifiers, provided they are of different types. For a record to apply to a certain request, that request must match all the qualifiers in at least one

of the `requester` lines as well as all of the qualifiers in at least one of the `criteria` lines.

EXAMPLE

```
requester -node=pluto -user=mickey
requester -node=uranus -user=beavis
criteria -operation=data-out -l_user=herschel
criteria -operation=data-in -l_user=planets
actions -approve
```

In the contrived example, requests coming from user “mickey” on node “pluto” OR user “beavis” on node “uranus” for either pulling files using the local user name “herschel” or pushing files using the local user name “planets” will be approved.

Note: You cannot use qualifiers of the same type in one `criteria` line. If you do so, only the last qualifier applies.

For example, if you write:

```
requester -node=pluto
criteria -operation=data-in -operation=data-out
actions -noapprove
```

only the second operation qualifier will apply, so requests from node "pluto" to pull files from the local node will be refused, but the status of requests for data transfer to the local node may be permitted, depending on the defaults specified elsewhere.

The correct way to write that record would be:

```
requester -node=pluto
criteria -operation=data-in
criteria -operation=data-out
actions -noapprove
```

The Actions Verb

The `actions` verb instructs RMFT FASTCopy Receiving Agent on how it should respond to a request from the specified `requester` that matches the given criteria. A simple example of a possible action might be to approve or deny a request. The `actions` verb can also be used to divert the request to a specific user account, to modify it or to monitor it.

The following qualifiers can be used with the `actions` verb:

-approve (negatable)

Specifies if the request should be approved. Its negation, `noapprove`, can be used to deny a given request. The default setting for all requests is generally `noapprove`, unless a valid user name and password are explicitly given in the request.

-l_user=local_user_name

Instructs FASTCopy to carry out the requested operation under the specified local user account instead of the account FASTCopy would have normally used, which would be either the one from which the command was issued or one given by a remote user with the `user` qualifier. On UNIX systems, setting the local user context for the incoming request is done automatically. On Windows, `fcopyd` can set the local user context only if that user's name and password are added to the proxy database with the `sl_passwd` utility (see [The sl_passwd Utility](#) (43) for details).

-class=class

Can be used only in the login security file; assigns the specified `class` name to an incoming operation's `class` parameter (which is blank otherwise) during a login security check. The value assigned can be used as a criteria in later security checks in other security files. Assigning a `class` to specific operations allows later security checks to differentiate between different kinds of requests that matched different, possibly secret and complex criteria during the login, and have otherwise similar parameters.

-remote_command (negatable)

Approves remote task activation by a FASTCopy operation initiated on another node. This is a FASTCopy default action. You can refuse remote execution of local images through FASTCopy using the negated form of the qualifier, `-noremote_command`.

-out_dir=directory_name

Generally, an incoming FASTCopy operation is allowed to write files according to the specified local user's permissions. Specifying a directory with this qualifier means that data can only be transferred to that directory, under the condition that the local user is allowed to write files there.

-inp_dir=directory_name

As above, an incoming FASTCopy operation is only allowed to read files according to the specified local user's permissions. Specifying a directory with this qualifier limits remote `data_out` operations to reading files from that directory only, under the condition that the specified local user is allowed to read files from there.

-root_dir=directory_name

This qualifier denies access for either read or write to any directory that is not a subdirectory of the specified *directory_name*.

-inp_include_sub_dirs (negatable)

Indicates that if an `inp_dir` is specified, the FASTCopy operation can also copy files residing in its subdirectories. This is the default.

-out_include_sub_dirs (negatable)

Indicates that if an `out_dir` is specified, the FASTCopy operation can also transfer files into its subdirectories. This is the default.

-dir_create (negatable)

Specifies if the operation is allowed to create directories. This is the default.

-overwrite (negatable)

Specifies if the operation is allowed to overwrite existing files on the local node. This is the default.

-size_limit=limit

Limits the maximum size of a single file that can be transferred to the local node by a FASTCopy operation.

-line_cipher=encryption_method

Determines what encryption method should be used to encrypt files transferred during an operation. Only the data (files) are encrypted by this encryption method; In particular, the messages sent back and forth between the program and RMFT FASTCopy Receiving Agent on the remote node are not encrypted. By itself, the `-line_cipher` qualifier does not make FASTCopy encrypt the transferred data.

The `-line_encrypt` qualifier must be specified in the command line or the *Security and Administration File* to force that action; A third qualifier, `-line_phrase`, must be included in the *Security and Administration File* to specify a key that will be used for the encryption. If the method specified with the requested operation does not match the value specified in the security file, an error occurs. If no method is specified or this qualifier is omitted, the encryption method is assumed to be `des-cbc`.

Refer to the appendix describing encryption methods in the *FASTCopy Reference Guide*, for an explanation of the various encryption methods available.

-line_encrypt

Specifies that transferred files should be encrypted, using the encryption method specified with `-line_cipher` and the key specified by the `-line_phrase` qualifier. The cipher and phrase specified by the requested operation must agree with those specified in the security file for the encryption to take place or an error will occur. Only the data (files) are encrypted during the operation. In particular, the messages sent back and forth between the program and RMFT FASTCopy Receiving Agent on the remote node are not encrypted. If no method is specified, the encryption method is assumed to be `des-cbc`. Refer to the appendix describing encryption methods in the *FASTCopy Reference Guide*, for an explanation of the various encryption methods available.

While the cipher and phrase must be specified both in the requested operation and in the responding node's security file, specifying `-line_encrypt` in either one of these places will cause files to be encrypted. The exception to this rule is when an incoming request is pushing files to the local node using the `-max_small_file` FASTCopy qualifier, in which case, if `-line_encrypt` is specified only in the receiving node's security file and not in the operation's command line, the files will be sent unencrypted.

-line_phrase=key

Specifies a key string that will be used as a key in encrypting transferred files, with the encryption method specified by the `-line_cipher` qualifier; Only the data (files) are encrypted using this key; In particular, the messages sent back and forth between the program and RMFT FASTCopy Receiving Agent on the remote node are not encrypted. If no method is specified, the encryption method is assumed to

be `des-cbc`. Refer to the appendix describing encryption methods in the *FASTCopy Reference Guide*, for an explanation of the various encryption methods available.

By itself, the `-line_phrase` qualifier does not make FASTCopy encrypt the transferred data; The `-line_encrypt` qualifier is used to force that action. `-line_phrase` only tells FASTCopy what key to use if it is required to encrypt a file. Both the cipher and phrase specified in the requested operation must agree with those specified in the security file for the encryption to take place, or an error will occur. Because of this, this form of encryption can be used only if the cipher and key phrase are known on both the local and the remote node. The *key* chosen can be any string including upper and lower-case characters, numbers, non-alphanumeric keyboard symbols, spaces, etc. If the string includes spaces, it should be enclosed in quotation marks. For good cryptographic strength, it should be at least 10 characters long. When choosing a *key*, exercise the same considerations you would use to choose a good password, i.e. avoid using recognizable names, words or phrases that can be easily guessed.

-mac[=phrase]

Specifies that all messages passed between the local and the remote node should be authenticated using the given key *phrase*. When `-mac` is used, all the data sent over the line, including both the files transferred and any communication between the FASTCopy program and the remote RMFT FASTCopy Receiving Agent, is signed with the specified key. This allows both sides to ensure that anything they receive during the operation does indeed come from the expected source, and that no data was modified or replaced in transit by a third party. The `-mac` qualifier can be specified without a key *phrase*, in which case FASTCopy will use an internal key to sign all its communication packets during the operation. This protection, however, has no cryptographic strength, since the FASTCopy program can be “cracked” by a third party and the internal key used to forge communication packets. If `-mac` is used with a key *phrase*, it must be the same phrase as that specified by the requested operation, or the operation will fail. When an agreed-upon key is used, FASTCopy gives both parties with strong cryptographically-protected authentication of their partner's identity. The *phrase* chosen can be any string including upper and lower-case characters, numbers, non-alphanumeric keyboard symbols, spaces, etc. If the string includes spaces, it should be enclosed in quotes. For good cryptographic strength, it should be at least 10 characters long. When choosing a key *phrase*, exercise the same considerations you would use to choose a good password, i.e. avoid using recognizable names, words or phrases that can be easily guessed.

-monitor

Activates a local monitoring task, allowing the requested operation to be monitored locally with the *fmonitor* command. This is different from FASTCopy central monitoring: the operation does not generate monitor messages but is simply

tracked by the local batch daemon, which records start, end, success or failure information only.

-mon_level=[min|default|detailed]

Specifies the level of detail of messages sent to a central monitoring node by the requested operation. This qualifier is relevant only if a central monitoring task is associated with this operation by a `-mon_node` qualifier specified either in the command line or in the security file. If the FASTCopy qualifier `-mon_level` is used in the operation's command line with a value that conflicts with that specified in the security file, it overrides this qualifier's value unless the qualifier `-mon_must` is also specified in the same `actions` line.

-mon_must

Specifies that any values specified in the same `actions` line with the qualifiers `-mon_node`, `-mon_type` and `-mon_level` should override the values of any matching qualifiers specified in the requesting operation's command line.

-mon_node=node_name[,node_name...]

Activates a central monitoring task, instructing the requested operation to report its progress to a central monitoring daemon on the node `node_name`. If more than one monitoring node is specified, the list must be enclosed in quotation marks. If the FASTCopy qualifier `-mon_node` is used in the operation's command line with a value that conflicts with that specified in the security file, it overrides this qualifier's value unless the qualifier `-mon_must` is also specified in the same `actions` line.

-mon_type=label

Specifies an identifying label for messages sent to a central monitoring node by the requested operation. The label is used to direct these messages to a particular monitor handler application. This qualifier is relevant only if a central monitoring task is associated with this operation by a `-mon_node` qualifier specified either in the command line or in the security file. If the FASTCopy qualifier `-mon_type` is used in the operation's command line with a value that conflicts with that specified in the security file, it overrides this qualifier's value unless the qualifier `-mon_must` is also specified in the same `actions` line. The default label assigned to a monitor message is `fcopy`.

-activity_time=time_frame

Limits the time during which the operation can take place on the local node. FASTCopy checks the time as long as the operation is active, terminating it if it exceeds the set *time_frame*. See *Setting Time and Time Frame Expressions* in the FASTCopy *Reference Guide* for further details.

-recurs (negatable)

Specifies that any file and directory specific security files that are below the file containing this rule should be taken into account in any security check that pertains to the files and directories associated with them. This is the default action. If you do not want records in lower-level security files to override records in a given security file, that file must contain a matching record explicitly specifying `-norecurs` (the negated form). This will cause FASTCopy to ignore all the records in more specific (lower-level) security files, UNLESS they explicitly tell it to refuse an operation it would otherwise approve (i.e. a lower-level security file contains a record that explicitly specifies `-noapprove` and matches an operation that would otherwise be approved by the higher-level file.)

-sl_security (negatable)

Specifies that file and directory specific security files can be transferred from this node (on outgoing operations) or to this node, to directories below the level of the file in which this qualifier is specified (if it is used in a directory's security file; if it is used in the general *Security and Administration File*, it allows this in the whole file system). To prevent security leaks, you cannot transfer security files to or from a node or directory unless you specify this explicitly using this qualifier (the negated form is the default).

-reason=string

Specifies text that should be displayed to the user if the operation is not approved. If there is another record that overrides this one and approves the operation, the text will not be displayed.

Defaults for FASTCopy actions include:

- `-remote_command` (true)
- `-inp_inc_subdirs` (true)
- `-out_inc_subdirs` (true)
- `-overwrite` (true)
- `-dir_create` (true)
- `-recurs` (true)

If the request explicitly includes a valid local user name and password, the default is `approve`, otherwise it is `noapprove`.

EXAMPLE

```
requester -node=hercules
criteria -operation=data-in
criteria -operation-data-out
actions -approve -noremote_command -monitor -l_user=guest
```

The above record specifies that users logging in from node “hercules” are allowed to pull and push files from the local node (`criteria -operation=data-in` and `-operation=data-out`). However, they are assigned the local user account of “guest” regardless of what local user they specify in their FASTCopy command. Furthermore, their capability to initiate task activation through FASTCopy's remote command feature is disabled (`-noremote_command`) and a local monitoring task is also activated which records all the details of their operation on the local node.

The `ifnot` Verb

The `ifnot` verb is the logical opposite of the `actions` verb in that it instructs RMFT FASTCopy Receiving Agent on how it should respond to a request that does not match the rule specified by the record: i.e. the request does not match both a specified `requester` and meet at least one `criteria`. Otherwise, the `ifnot` verb behaves just like the `actions` verb. It also can instruct the daemon to approve or deny a request, and it accepts all the qualifiers detailed under `actions`. As a general rule, if a given request matches a record, then any actions specified in that record with the `actions` verb are added to the list of actions for that request. If the request does not match the record, any actions specified in the record with the `ifnot` verb are added to the list. A single record cannot contain both an `actions` and an `ifnot` line.

There are some situations that require you to use the `ifnot` verb, such as when you want to limit access in a context where it would not normally be limited. For example, by default any one with a local user name and password can copy files; if you want only users who specify a given code to be able to copy a specific file, you can create a file-level security file for that file, including the following record:

```
requester
criteria -code=TrustMe -read
ifnot -noapprove
```

Any request to read the file that would be approved by default or because of previous records in the file or previous security files, for example a request that included a legitimate local user name and password, will fail unless the `-code` qualifier was specified in the command line with the value "TrustMe". This record will apply even if a higher-level security file approved the request and specified the `-norecurs` action, because `-noapprove` is used here explicitly.

Despite the example, cases where approval is implicit and `ifnot` must be used to explicitly deny access should be very rare. Security is best served if all operations are disallowed by default (i.e. by higher-level or more general rules) and for any operation, approval is specifically given in a security file record.

The Search Verb

As explained earlier (see *Parsing the Security File*), during a security check the security file is searched for records matching the request's details, and any actions described in matching records are retained and apply to the request, with later records taking precedence over conflicting earlier records. However, this standard parsing of the security file can be modified, by adding a line beginning with the `search` verb to a record.

The `search` verb instructs the security module how to proceed in its examination of the security file. It can modify the default search pattern using one or both of the following two qualifiers:

- `-ignore_defaults` - Instructs the security module to ignore all actions prescribed by any records matching the request which preceded the current one.
- `-stop_search` - Instructs the security module to stop searching the file for additional records, keeping only the actions found this far.

EXAMPLES

```
requester -node=alf -user=joe
actions -approve
search -stop_search
==
requester -node=alf
actions -noapprove
```

By adding the **search** line, we ensure that if the user “joe” from the “alf” node issued the request, the security module will record the `approve` action and finish its examination of the security file; If it is anyone else, it will continue to the next record.

Here's another example:

```
requester node=alf
actions -approve -monitor
==
requester -node=alf -user=joe
actions -approve
search -ignore_defaults
```

In the above example, the actions in the second record do not necessarily override the monitoring instruction given in the first record. However, by adding the `search`

line with the `-ignore_defaults` qualifier, we have made sure that if the user “joe” from the “alf” node issues a FASTCopy request, it will not be monitored.

```
requester node=alf
actions -approve -monitor
==
requester -node=alf -user=joe
criteria -operation=data-out
actions -approve
search -ignore_defaults -stop_search
==
requester -node=alf -user=joe
actions -noapprove
```

In the above example, the first record specifies that all FASTCopy requests from node “alf” should be approved and monitored; the third record specifies that FASTCopy requests from the user “joe” on node “alf” should be refused; the second record is an exception to both rules; it specifies that if the user “joe” on node “alf” tries to copy files from the local node, this action should be approved (this is not overridden by the third record because of the `-stop_search` qualifier), and not monitored (because of the `-ignore_defaults` qualifier).

The On Verb

Maintaining a Global Security File

In some cases it is simplest to maintain a single *Security And Administration File*, even if administering multiple nodes on the system, rather than maintaining a different file on each node. While each node must contain the file **softlink.security** or a symbolic link by that name to a common file, the file can be the same on all nodes. This allows you to update the file only once, and then replicate the updated file on all the other nodes in your network.

However, you may still sometimes want different rules to apply on different nodes or types of nodes. You can do this with a single *Security And Administration File* by adding record beginning with the verb `on`.

The `on` verb identifies a node or network on which the rule the record describes is valid. This verb is used when an identical security file is kept on several nodes, containing rules which are different for each node. This verb's qualifiers are used to specify either a single node or a sub-network on which the particular record is a valid rule. If the verb is absent from a record, the rule it describes will be valid on every node where the security file is found.

`on` qualifiers include:

-node=node_name

This qualifier specifies that the record is valid on node *node_name*. You must specify the real name of the local node rather than using the string `localhost`, since using `localhost` will make this record apply on all nodes. The *node_name* can be the node's full name or an alias, depending on how your node recognizes itself.

-address=address

This qualifier specifies that the record is valid on a node with the specified *address*. Combined with a net mask, it can be used to designate an entire network of nodes.

-net_mask=net_mask

This qualifier combined with an address string, is used to designate a set of addresses. This allows you to use a single criteria line to specify a group of nodes, or an entire network. Each "zero" value bit in the net mask string matches any value of the corresponding bit of the requester's address, while any "one" bit in the net mask requires the corresponding bit in the requester's address to match the specified address.

A global *Security And Administration File* with some modifications using the `on` verb is most useful when the nodes are very similar; if you find yourself adding too many node-specific records, it may be better to use separate security files.

EXAMPLE

Here's an example of how the `on` verb can be used:

```
requester
criteria -operation=execute -password
actions -approve
==
on -node=vimto
requester
criteria -operation=execute
actions -noapprove
==
on -node=alf
requester -node=localhost
criteria -operation=execute
actions -approve
```

On all the nodes in the network, both local and remote users are able to execute images using FASTCopy's remote command feature if they give a correct local password. There are two exceptions: node "vimto", where all requests for image execution through FASTCopy are refused, and node "alf", where local users are able to execute images using FASTCopy without giving a password.

Note: In the third record's `requester` line, the string `localhost` is used in the node name, while in the `on` line, the node's real name must be used for the record to be specific to the "alf" node.

Using Variables in Records

When writing a security file, you might want to describe general cases which apply to any user, node or group. For example, you might want to stipulate that the requester's user name matches the local user name he can use, or that the requester's group must match the target group.

To do this, you can use the following **variables**:

- `%n` - substitutes for the requester's node name anywhere in the record.
- `%u` - substitutes for the requester's user name anywhere in the record.
- `%g` - substitutes for the requester's group name anywhere in the record.

- %N - substitutes for the full name of the local node anywhere in the record. Like the node specified with the `on` verb, this is the node's real name or alias, not the string `localhost`.
- %p - substitutes for the peer node name anywhere in the record.
- %U - substitutes for the local user name used in a request (which is the same as the `criteria` qualifier `l_user`) anywhere in the record.

In addition, the string `localhost` can be used anywhere as a reference to the local node (where the security file is checked). When characterizing a requested operation, any requests originating from the local node have the string `localhost` assigned as their node name, so that it matches that value in the security file.

Here's an example of variable use in a security file, as part of a "personal mailboxes" system: The node "mule" is a system's file server. Users on all the other nodes in the system use FASTCopy to transfer files to or from the file server. For management purposes, each remote user can only upload or download files from his own personal directory on the file server. The users' personal directories are arranged hierarchically according to their nodes of origin and user names.

The *Security And Administration File* includes the following record:

```
on -node=mule
requester -node=localhost
ifnot -approve -root_dir=/data/users/%n/%u
```

This record stipulates that if the requester's node is not the local node (`localhost`), file transfer will be allowed only to and from a specific directory that matches the user's name and node. This single record stipulates that user "bill" on node "quagga" can transfer files only to or from the directory `/data/users/quagga/bill` (and its sub directories) while user "helen" on node "zebra" can transfer files only to or from the directory `/data/users/zebra/helen` (and its subdirectories). An equivalent record could be written that uses the `criteria` qualifier `-incoming` to select all the operations that did not originate on the local node.

The record would then look like this:

```
on -node=mule
requester
criteria -incoming
actions -approve -root_dir=/data/users/%n/%u
```

but would function the same.

Implementing the Proxy Security Mechanism

The Proxy Security Mechanism provides flexible tools for administering FASTCopy operations between a system and other nodes in a network. It also allows the administrator to restrict and control remote requests based on what is known about the requested operation, and use a number of effective security measures. Of course, each of these functions requires different considerations and the use of different features. This section attempts to organize the various features of the Proxy Security Mechanism according to the context in which they are most useful.

Different Security Checks

The Proxy Security Mechanism operates differently depending on whether a requested operation is initiated locally (by a local user using FASTCopy) or remotely (by a remote request received by *RMFT FASTCopy Receiving Agent*); local requests are checked only against the main *Security and Administration File*, while incoming requests are subjected to two different kinds security checks: The first check examines rules in both the main *Security and Administration File* and in the "login" security file. This check is made to establish the local user account that the remote request will be carried under.

Once the local user context of a remote operation has been established, further security checks are carried for each file or directory that is effected by the remotely-initiated operation. These security checks examine the rules in the main *Security and Administration File*, in the security file of the specific data file that will be pulled or the specific directory into which files will be pushed, and in all the security files of any directories in the path.

The Login Security Check

The login stage of an operation provides an important check point. At this stage, you should check login parameters such as passwords and codes, and set up security parameters that will affect the entire operation, such as message authentication and encryption parameters (the encryption method and phrase should be set at this stage; actual encryption can be decided on per file).

Placing any rules that relate to these parameters in the login security file provides increased security, because that file can only be read by privileged users. RMFT FASTCopy Receiving Agent operates as a privileged user only during the login check, before setting the local user context that an incoming operation will actually run under.

To successfully pass the login security check, a remote request must match at least a single record that explicitly specifies an action of `approve`, and must end with a legitimate system user name as the value of the operation's `l_user` parameter. This parameter reflects either the user name given in the operation's command line

with the FASTCopy qualifier `user`, or the user assigned to the operation by the `actions` qualifier `l_user`. If no local user context can be determined, the operation will fail.

Once an operation has successfully passed the login security check, it must still be explicitly approved by at least one matching record, either in the main *Security and Administration File* or in a file or directory level security file. Because sometimes the exact operation allowed will depend on what login check the operation passed, the parameter `class` is provided, to differentiate between different types of requests in later security files.

EXAMPLE

The login security file includes these two records:

```
requester
criteria -l_user=guest -password
actions -approve -line_phrase=fgrg#t4@8$%x -noremote_command -class=furriners
===
requester
criteria -peer_addr=192.147.160.64 -peer_netma=255.255.255.0
actions -approve -l_user=guest -class=homeboys
search -ignore_defaults
```

The first record specifies that any incoming operation specified with the local user name “guest” and an appropriate password should be approved (at the login level), but must encrypt files using the specified phrase (and the default encryption method, `des-cbc`), and it will not be allowed to execute local images. The second record permits anyone from a remote node with an address beginning with the digits `192.147.160` (the local sub network) to carry out operations under the local user account “guest”, without having to specify a password. This record also overrides the restrictions in the first record, if they apply (using the `search` directive).

The two rules distinguish between two different classes of incoming operation: one triggered from remote nodes that are “friendly”, part of the organization's network, and one triggered from remote nodes that are “outside” the familiar network, but are initiated by (hopefully) friendly users, who know the correct password for the “guest” account. The `class` parameter allows you to distinguish between these two classes in later files; for example, you want developers from other nodes in your organization to be able to put files in a certain directory called `/pub/version2.x/patches/`.

You want customers from outside your organization to only be able to download files from that directory. To do this, you can create a security file for that directory (`/pub/version2.x/patches/.sl_sec`) and include the following records in it:

```
requester
criteria -class=homeboys -operation=data-in
actions -approve
====
requester
criteria -class=furriners -operation=data-out
actions -approve
```

And if you want to restrict members of the “furriners” class from doing anything anywhere else in your file system, you can include this rule in your main *Security and Administration File*:

```
requester
criteria -class=furriners
actions -noapprove
```

Operations within a Closed Network

It is important to distinguish between the way the Proxy Security Mechanism can be used in a trusted networking environment and how it can be used in one which contains potentially hostile or untrusted remote users.

A trusted environment is a network in which any peer nodes which can contact your system are considered friendly. It is usually a closed network, or one at least enclosed inside a firewall, and the peer nodes are either administered by the same person who administers the local node, or by trusted individuals.

A trusted environment can be a closed network using an identical, replicated *Security And Administration File* to manage all FASTCopy activities, with specific local exceptions defined using the `on` verb. Within the trusted environment you can use identifiers such as logical groups, that are useful for managing file transfer activities but require that you trust the remote system administrator to allow a requester to issue a request in the name of a given group (through his own node's security file). Within a trusted environment some security restrictions can be lifted, for example, if you are sure of the remote user's identity, you can disable the need for a proper user name and password, and allow the remote user to automatically carry out the operation in the context of a certain local user, using the actions qualifier `-l_user`.

For example, if you really trust the identity of a remote user “cramer”, you can put the following in the main *Security and Administration File* or in the login security file:

```
requester -user=cramer
criteria -data-out -nopassword
actions -l_user=jerry -approve
```

However, this is recommended only inside a secure and trusted environment, where you are certain that no unknown party will be able to connect to your machine. In most environments, the minimal security measures of requiring a user name and password should be enforced and perhaps augmented by other security features, as described below.

Data Transfer Firewall

If you are not operating within a closed and trusted networking environment, FASTCopy's Proxy Security Mechanism can protect your system from unwanted FASTCopy operations by unknown or hostile remote users. The security mechanism can create a Data Transfer Firewall, blocking any remote requests by unidentified or unauthenticated users.

The default security settings ensure that all remote FASTCopy operations will be refused, unless the remote user provides a valid local user name and a matching password. This will prevent unauthorized nodes you may be networked with from using FASTCopy to infiltrate your system. However, in all cases when the requesting node is one from outside your trusted environment or firewall, it is best to restrict the request as much as possible, for example completely denying remote activation of images or access to privileged accounts. In addition, the Proxy Security Mechanism provides further methods to protect your system from security breaches and to authenticate the remote user's identity.

The security file rule base can be used to screen any incoming operation and restrict or modify it according to its parameters (e.g. remote node name, address, remote user name, operation required, etc.). For example, you can use the security file to approve or refuse requests based on the `peer_node` and `peer_address/peer_net_mask` identifiers. Identification based on parameters such as requester user name or group is less useful here, since these can be easily faked. Use such identifiers only if the peer node is a trusted one.

Proxy Login with Logical Passwords

If you are concerned that sensitive information must be transferred over the network, or that hostile remote users may have obtained your passwords illegitimately, or if you don't want to provide remote users with real system passwords, FASTCopy supports a proxy login process, allowing you to define logical usernames and passwords using the `sl_passwd` utility. During the login security check, the user name and password specified in the remote operation's command line are first compared to any local user and system password on the local node; if no match is found, `fcopyd` then checks for a match with any user names and passwords defined using `sl_passwd`; if a match occurs, the user name given in the incoming operation's command line is assigned to the `target_user` identifier, and the `l_password` flag is set to "true". If you want to approve the operation, you must then set the local user context using the `actions` qualifier - `l_user`. This allows you to permit access to your system without providing remote users with information about real system user accounts and passwords.

The use of the `sl_passwd` utility is described in [The `sl_passwd` Utility](#) (43).

When the Network is Not Trusted

Two security features are especially useful when the network is not trusted (e.g. connections over the Internet). The first is message authentication, which can be required on incoming requests. This provides authentication of the other party's identity for both sides, and ensures that the transferred packets are not tampered with or replaced by unknown attackers elsewhere on the network.

The second feature is encryption of data transmitted over the network for privacy, using a previously agreed upon `-line_cipher` and `-line_phrase`. You can force encryption on incoming requests using the `-line_encrypt` qualifier.

Since the other side must know the message authentication and line encryption phrases your node demands, both these features also serve as password-like identifiers.

Securing Specific Resources

The Proxy Security Mechanism allows you to attach a security file to any specific file or directory, so that you can refine the requirements to access that resource. These file and directory-level security files have been described earlier, but here are some considerations about how they are used.

If you generally block your file-system to remote FASTCopy operations (a prudent measure), you can expose one or more directories specifically to remote users, using directory-level security files. In addition, you can use require that users give a resource-specific code (using the `-code` qualifier) to access the particular file or directory.

When *RMFT FASTCopy Receiving Agent* carries out a security check to determine if a given file can be transferred, it evaluates any matching rules in the main *Security and Administration File*, the security files of any directories along the path, and the security file associated with the relevant file (or directory) itself. The actions specified in later files override actions specified in previous (more general) files, unless a higher-level file included the `actions` qualifier `-norecurs`. In this case, rules in further files are ignored, unless they tell *fcopyd* to refuse an operation it would otherwise approve.

When the transferred files are links, the security files evaluated by *fcopyd* depend on the behavior specified with the `-links` qualifier. If the operation specifies that only the link should be copied, the security check includes only the link's security file (and any security files in its path); if the operation specifies that both the link and the file it points to should be copied, *fcopyd* first checks all the security files that apply to the link, and if the operation is approved at this stage, it also checks all security files that apply to the actual file the link points to. If the link is part of a chain of links, i.e. it is a link that points to a link that points to a file, only the security files that apply to the first link and the actual data file are evaluated.

Control Of Outgoing Operations

The `criteria` qualifier `-outgoing` is used to tag operations initiated by a node's local users, as opposed to incoming operations issued from peer nodes. This criteria applies to FASTCopy tasks such as pulling or pushing files to or from peer nodes (`data-in` and `data-out`), as well as to remote task activation (`execute`). It does not apply to FASTCopy operations submitted for batch execution or to local monitoring requests.

If Use of this qualifier allows you to control and restrict the activities of local users, in much the same manner as you can limit incoming requests from remote users. Remember that if you approve outgoing operations, they might still be refused by a peer node, if its own security file does not approve them.

The most important use of this qualifier is to automatically set global parameters of outgoing FASTCopy operations. For example, you can use the `-mon_node`, `-mon_type`, `-mon_level` and `-mon_must` qualifiers to automatically force any operation to report to a given central monitoring node. You can set the `-line_phrase` and `-line_cipher` that will be used when transferred data is encrypted, even if you leave the decision about what files should actually pass encrypted to the user (although you can force encryption using `-line_encrypt`).

In the same way, you can force message authentication using a specific phrase, with the `actions` qualifier `-mac`. These modifications can be made to any subset of outgoing operations, for example to any outgoing transfer requests made to a certain node or sub-network.

Working with Groups

Groups are sets of users which are internally defined in FASTCopy through the security file. Any job submitted for batch operation can be assigned a group, and that group is considered the owner of the job for monitoring and control purposes. Groups should be used to control (local) batch operations by local users, or to manage operations by users on different nodes inside a closed, trusted network.

This is because the remote node's administrator must be trusted to allow a user to submit a request in the name of a given group. To submit an operation for batch execution under a specific group name, a user specifies the group name in the command line using the FASTCopy qualifier `-group`. For the submission to be accepted, the local security file (on the node on which the job is submitted) must permit the user to submit jobs to the requested `target_group`. Ongoing jobs can then be monitored and controlled based on their group name, again, if the security file allows the user to do so. If no group qualifier is used, the job is assigned to the default group `users`.

A good use for groups is when users on different nodes are all working on a single project over the network. The group name can be used to designate all FASTCopy batch jobs related to that project. The security file can be used to provide

authorizations for all FASTCopy batch operations submitted under that group name to operate under a certain local user name, even if the user did not give a local user name or password. This way there is no need to provide each remote user with authorization.

You can also designate a project manager, and use the security file to provide only him with complete monitoring capabilities over the whole group's jobs.

EXAMPLE

```

requester -group=ateam
criteria -operation=submit
criteria -operation=cmd-in
actions -noapprove
===
on -node=gabriel
requester -user=hannibal -group=ateam
requester -user=bill -group=ateam
requester -user=sam -group=ateam
requester -user=max -group=ateam
requester -user=claire -group=ateam
criteria -operation=submit
actions -approve
==
on -node=raphael
requester -user=hannibal -group=ateam
requester -user=eric -group=ateam
requester -user=joy -group=ateam
requester -user=debbie -group=ateam
criteria -operation=submit
actions -approve
==
on -node=server
requester -group=ateam
actions -approve -l_user=ouser
==
requester -group=ateam
criteria -operation=cmd-in -target_user=%u -target_group=%g
actions -approve
==
requester -group=ateam -user=hannibal
criteria -operation=cmd-in
actions -approve

```

Above is a sample security file for a network using FASTCopy in a network-wide project. The project members on every node are allowed to submit batch operations under the group name "ateam"; On the server node, batch operations submitted using the "ateam" group name are approved without a need for local user name and password, and are carried out using the local user name "ouser".

The first record is a default, forbidding anyone from submitting or monitoring jobs using the "ateam" group name, because this name is reserved for people working on this specific project. The second and third records specify which users on which

nodes are allowed to submit jobs using that group name. Since this record comes after the default, it takes precedence.

The fifth record permits each user in the group to monitor jobs he has submitted under the group name, but not those submitted by anyone else. The final record specifies that the group manager, user "hannibal" (who has accounts on both nodes), is allowed to monitor all the group's submitted jobs.

The `sl_passwd` Utility

The `sl_passwd` utility is used for creating and managing the logical passwords that are employed by FASTCopy's Proxy Security Mechanism. The `sl_passwd` image is stored in the same directory as other FASTCopy-related executables. It stores its data (user names and passwords) in the file `sl_passwd` in the base directory.

On Windows NT, if you want RMFT FASTCopy Receiving Agent to set an incoming request's user context to a specific user account, you must also add the user name and password of that account to the proxy database using `sl_passwd`, as detailed below.

To see a list of the commands and options `sl_passwd` offers, type:

```
sl_passwd help
```

The output looks like this:

```
Usage: sl_passwd <command> [-qualifier] [-qualifier] ...
Commands:
list [username_wildcard]
add <username> [-password=xxx] [-expiration=ddmmyy]
[-comment=xxx] [-suspend] [-system_password]
modify <username> [-password=xxx] [-[no]expiration=ddmmyy]
[-[no]comment=xxx][-[no]suspend] [-[no]system_password]
remove <username_wildcard>
```

To view a list of all the user names that have logical passwords, type:

```
sl_passwd list
```

(You can add a user name or a partial user name containing wildcards as an additional argument to view only the relevant entries).

The output looks like this:

```
alf:/usr/flogic>_sl_passwd list
Username Comment Expiration
-----
*elad Wed Oct 10 00:00:00 1990
dotan this is a test
alf:/usr/flogic>_
```

An asterisk (*) near a user name indicates that his password is temporary and will expire eventually. Comments and expiration dates can be added when the user's password is first entered or with the `modify` command.

To add a new user, type the following:

```
>sl_passwd add username -password=password -comment=text
-expiration=ddmmyy
```

All the parameters are optional. If a username or a password are not specified, `sl_passwd` will prompt you for them. The user name given can be an existing local user name or a new one. If an incoming FASTCopy request specifies this user name and password, the `l_password` criteria will be matched.

When adding a password, you can optionally include a comment with the `-comment` qualifier and specify an expiration date in the `ddmmyy` format. These values will appear in the appropriate fields when you use the `list` command.

Whenever you use a string separated by spaces, remember to enclose it in quotation marks. The quotation marks and any wildcards should always be escaped by backslashes if used from the UNIX shell.

EXAMPLE

```
alf:/usr/flogic>_sl_passwd add joe -password=mxzyptlk
-comment=\"proxy login for joe at ACME\" -expiration=310495
alf:/usr/flogic>_sl_passwd list
Username      Comment                               Expiration
-----
*elad Wed                               Oct 10 00:00:00 1990
*joe      proxy login for joe at ACME Mon May 1 00:00:00 1995
alf:/usr/flogic>_
```

On Windows NT, you will need to add information for real user accounts if you want to login incoming requests under those accounts. This lets you change the incoming request's user context with the `-l_user` actions qualifier.

To add a system user to the `sl_passwd` proxy database, use the `add` command with the real (system) user name and the `-system_password` qualifier. When prompted, enter the system password (you will be prompted twice, for verification purposes):

```
c:\>sl_passwd add username -system_password
```

EXAMPLE

```
C:\Program Files\fcopy>sl_passwd add jane -system
System password:
Verification:
```

This step can provide a security hole (remote users don't need to know the system password), but if coupled with a logical password, this actually enhances security, because remote users can be required to give a password, but giving them this password tells them nothing about the real system password.

Finally, to remove a user, type:

```
sl_passwd remove username
```

SSL 3.0 Authentication

As an administrator, you may wish to prevent users who do not provide a valid certificate from accessing the system. This can be accomplished by adding FASTCopy SSL 3.0 qualifiers to the main *Security and Administration Files* (**softlink.login** and **softlink.security**).

To prevent unauthenticated users from performing FASTCopy operations on the server, the following FASTCopy SSL 3.0 qualifiers may be included in the **softlink.login** security file

- `-authenticate`
- `-common_name=xxxx`

-authenticate

In the context of the **softlink.login** file, the `-authenticate` qualifier verifies that the client's certificate is valid and that the `Common Name (CN)` field of the client's certificate matches the hostname of the client or the value of the `-peer_common_name` qualifier (described below). If the client's certificate is not valid or if there is a `Common Name` mismatch, the request will be refused.

-peer_common_name

In the context of the **softlink.login** file, the `-common_name` qualifier indicates the expected contents of the client certificate's `Common Name` field, when it is not the client's hostname.

Note: Remote users who are required to provide the value of the *server* certificate's `Common Name` field, must be provided with this information by the administrator. If the user is not required to provide this information the value of the server's `Common Name` field must be its hostname.

Example 1:

If the **softlink.login** file contains the following records:

```
req
action -noapprove
====
req
crit -auth
action -approve -class=ssl_auth
```

and the **softlink.security** file contains the following record:

```
requester
crit -class=ssl_auth
action -approve
```

FASTCopy sessions that have not been encrypted with SSL 3.0 protocol will be rejected by the server.

FASTCopy users who try and copy files without utilizing FASTCopy's SSL 3.0 feature, will receive the following error:

```
FCOPY-E-NETLOGIN, remote login failed on node localhost
-SEC-E-NOAPPLN, permission rejected at security file C:\Program
Files\SoftLink\
security\softlink.login, line #2
```

Example 2:

In this example, the system administrator has added user Bob to the proxy database using the [The sl_passwd Utility](#) described on page 43.

Thus, if the **softlink.login** file contains the following record, Bob will be able to access the server without having to specify a username or password, on condition that the value of his certificate's `Common Name` field matches the value of the `-common_name` qualifier specified in the **softlink.login** file.

```
req
action -noapprove
====
req
crit -auth -common_name=Client-Demo
```

```
action -approve -l_user=Bob
```

Example 3:

If the **softlink.login** file contains the following record, Bob will be able to access the server without having to specify a user name, password, or target directory. Access will be granted on condition that the value of his certificate's Common Name field matches the value of the `-common_name` qualifier specified in the **softlink.login** file.

```
req
action -noapprove -reason="user was not fully authenticated"
====
req
crit -auth -common_name=Client-Demo -nopassword
action -class=ssl_auth -approve -default_dir=e:\target -l_user=Bob
```

Note: The `-reason` and `-nopassword` qualifiers have also been included in the **softlink.login** file. Thus, to successfully perform a FASTCopy operation, Bob must also specify `-nopassword` in his FASTCopy command. If access is denied, Bob's error message will contain the text specified with the `-reason` qualifier.

IMPORTANT: If the `-class` qualifier is used in the **softlink.login** file, it must also be added to the **softlink.security** file.

See also [2. RMFT FASTCopy Receiving Agent Manager](#) below.

2. RMFT FASTCopy Receiving Agent Manager

RMFT FASTCopy Receiving Agent (Windows service name: RMFT FASTCopy Server) responds to transfer requests from FASTCopy Clients. Therefore, it must be installed on any machine to/from which files are transferred using RMFT FASTCopy protocol. RMFT FASTCopy Receiving Agent Manager enables you to:

- Determine whether FASTCopy Clients need to establish a secure connection (SSL) to access RMFT FASTCopy Receiving Agent and whether they also need to provide a client certificate.
- Stop and start the RMFT FASTCopy Receiving Agent service if necessary
- Resolve esoteric problems by setting traces.

RMFT FASTCopy Receiving Agent Manager can be opened on the RMFT Server machine or on a machine on which only the RMFT Receiving Agents are installed. If you open RMFT FASTCopy Receiving Agent Manager on the RMFT Server machine, you will be able to configure RMFT FASTCopy Receiving Agent either locally or on remote machines. If you open RMFT FASTCopy Receiving Agent Manager on a machine on which only RMFT FASTCopy Agent is installed, you will only be able to configure RMFT FASTCopy Receiving Agent on that machine (i.e. you will not be able to remotely connect to other RMFT FASTCopy Receiving Agents).

Opening RMFT FASTCopy Receiving Agent Manager

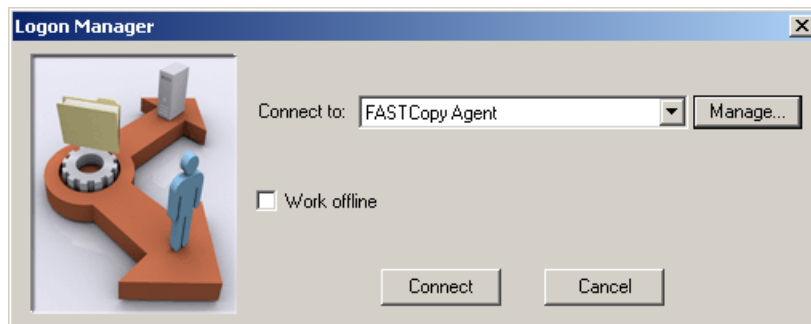
This section describes how to open RMFT FASTCopy Receiving Agent Manager, how to configure connection settings and how to connect to different nodes running RMFT FASTCopy Receiving Agent .

To open RMFT FASTCopy Receiving Agent Manager:

1. Select **Start > All Program > RepliWeb > Managed File Transfer > Daemon configuration tools > RMFT FASTCopy Receiving Agent Manager**.

If RMFT Server is not installed on the machine (i.e. only the RMFT Receiving Agents are installed), **RMFT FASTCopy Receiving Agent Manager** opens. In this case, continue from [Configuring RMFT FASTCopy Receiving Agent](#).

If RMFT Server *is* installed on the machine, the **Logon Manager** dialog box opens.



Note: If RMFT FASTCopy Receiving Agent Manager is opened on the RMFT Server machine, the IP address/hostname of the RMFT Server machine will be displayed in the **Connect to** field. If you want to open RMFT FASTCopy Receiving Agent Manager on another machine (from the RMFT Server machine), you will need to add that machine's login details to the **Logon Manager** as described in [Managing RMFT FASTCopy Receiving Agent Nodes](#).

2. Click **Connect** to connect to the machine displayed in the **Connect to** field of the **Logon Manager** dialog box.

-OR-

Select another machine (node) from the drop-down list and then click **Connect**.

For information on adding RMFT FASTCopy Receiving Agent nodes, see [Managing RMFT FASTCopy Receiving Agent Nodes](#).

3. If prompted, enter your password for logging in to FASTCopy Receiving Agent Manager and the domain name (if necessary).

Note: You will only be prompted for a password if the **Save password** check box in the **Update Logon Details** dialog box has not been selected.

4. Click **Connect**.
RMFT FASTCopy Receiving Agent Manager opens.

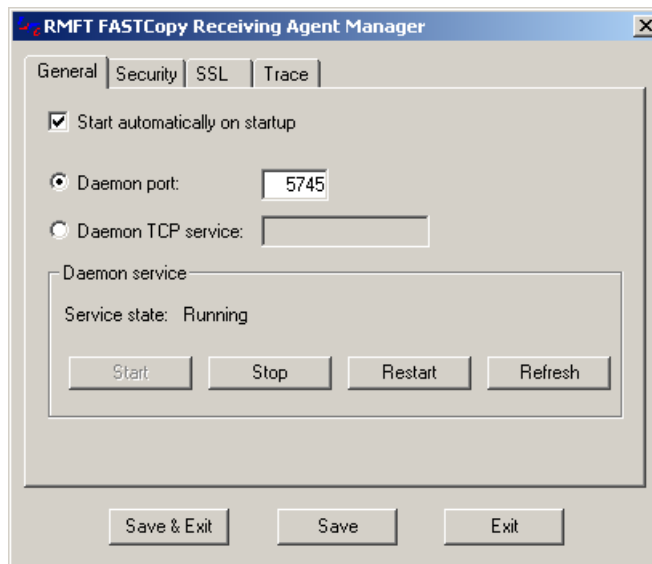
Configuring RMFT FASTCopy Receiving Agent

RMFT FASTCopy Receiving Agent Manager consists of the following tabs.

- [General Tab](#)
- [Security Tab](#)
- [SSL Tab](#)
- [Trace Tab](#)

General Tab

In the **General** tab, you can configure RMFT FASTCopy Receiving Agent startup options and manage the RMFT FASTCopy Receiving Agent service.



To start RMFT FASTCopy Receiving Agent automatically during Windows startup:

- ◆ Select the **Start automatically on startup** radio button.

To change the default RMFT FASTCopy Receiving Agent port number:

- ◆ Select the **Daemon port** radio button and specify the new port number in the adjacent field.

By default, all FASTCopy file transfer operations use port 5745.

To change the RMFT FASTCopy Receiving Agent service name:

1. Open `~\system32\drivers\etc\services`
2. Change the FASTCopy service name (fcopy\$server) and port number. **Save** the file.
3. Select the **Daemon TCP service** option and specify the service name.

To stop the RMFT FASTCopy Receiving Agent service:

- ◆ Click the **Stop** button

If RMFT FASTCopy Receiving Agent is stopped, the **Service state** field will indicate that the service is **Stopped**.

To restart the RMFT FASTCopy Receiving Agent service:

- ◆ Click the **Restart** button.

If RMFT FASTCopy Receiving Agent is restarted successfully, the **Service state** field will indicate that the service is **Running**.

To refresh the RMFT FASTCopy Receiving Agent service:

- ◆ Click the **Refresh** button.

To start the RMFT FASTCopy Receiving Agent service:

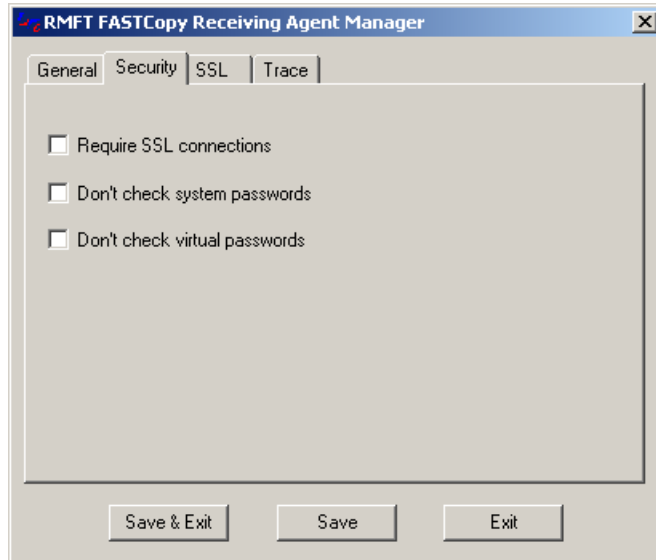
- ◆ Click the **Start** button.

If RMFT FASTCopy Receiving Agent is started successfully, the **Service state** field will indicate that the service is **Running**.

Security Tab

In the **Security** tab of RMFT FASTCopy Receiving Agent Manager, you can determine how RMFT FASTCopy Receiving Agent should respond to connection requests from FASTCopy Clients.

1. Click the **Security** tab.



2. To require all FASTCopy Clients to connect using a secure connection, select the **Require SSL connections** check box.

If you select this option, only FASTCopy Clients with a valid client certificate will be able to establish a session with the RMFT FASTCopy Receiving Agent machine. This option also requires the CA certificate for the issuing client certificate authority to be imported to the CA store on the RMFT FASTCopy Receiving Agent machine. The CA certificate for the issuing server certificate authority must also be imported to the CA certificate store on the FASTCopy Client machine.

3. To prevent RMFT FASTCopy Receiving Agent from trying to log in to a system account that has the same user name as the virtual user, select the **Don't check system passwords** check box.

This option should be selected if a system account with the same user name as the virtual user name exists on the RMFT FASTCopy Receiving Agent machine. Under normal circumstances, FASTCopy tries to log in to the system account using the password provided. If it is unable to log in to the system account, FASTCopy will check its proxy database for a virtual account that matches the user name and password. In certain situations, this may result in the real system user being locked out of his or her account due to repeated attempts to log in to his/her account using the

virtual password. In addition to the possibility of locking out a system user, system event monitors may falsely report that a malicious user is trying to gain access to the system.

4. To prevent unauthorized users from accessing the system using a virtual user name and password, select the **Don't check virtual passwords** check box.

This option should be selected if FASTCopy users have been allocated real system accounts. Under normal circumstances, FASTCopy tries to log in to the system account using the password provided. If it is unable to log in to the system account, FASTCopy checks its proxy database for a virtual account that matches the user name and password. This may result in unauthorized system access if the system user account is no longer valid, but a virtual user account with the same user name and password exists.

5. To require FASTCopy Clients to log in using only a certificate (as opposed to a certificate and a password), select all of the available check boxes.

If you select this option, only users with a valid certificate will be able to transfer files to/from the RMFT FASTCopy Receiving Agent machine.

6. Select the **General** tab.
7. In the **Daemon service** region, click **Restart**.
8. Click **Save & Exit**.

For more information on creating virtual user accounts and adding them to FASTCopy's proxy database, see [The sl_passwd Utility](#).

SSL Tab

You can configure RMFT FASTCopy Receiving Agent to authenticate itself using either MSCAPI or OpenSSL certificates (the default). Both of these procedures are described in the following sections:

- [Using MSCAPI Certificates](#)
- [Using OpenSSL Certificates](#)

Using MSCAPI Certificates

You can test functionality using the MSCAPI compliant demo certificates provided with the installation.

Note: The demo MSCAPI certificates provide no real security and are intended for testing purposes only.

In a standard installation, the demo certificates reside in the following directory:

~\RepliWeb\RMFT\ssl

The following server-side certificates are available for testing:

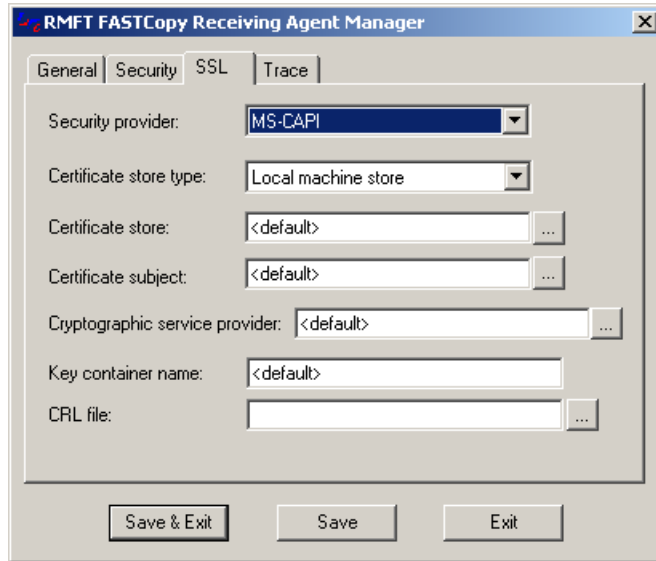
- **fc_server.pfx** - the server certificate
- **softlink_ca.cer** - the Certificate Authority certificate
- **softlink.crl** - Contains details of the **softlink_revoked.pfx** certificate (see below).
- **softlink_revoked.pfx** – Demo certificate that has been revoked by the issuing CA

The password for the server private key file is: **demosever**

Unless configured differently, RMFT FASTCopy Receiving Agent will authenticate itself using the demo MSCAPI compliant server certificate provided with the installation. RMFT FASTCopy Receiving Agent assumes that this certificate has been imported to the Local Machine's **Personal** store.

To override the default MSCAPI settings:

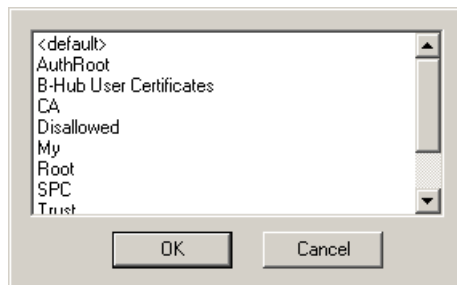
1. Open RMFT FASTCopy Receiving Agent Manager as described in [Opening RMFT FASTCopy Receiving Agent Manager](#) above.
2. Select **MSCAPI** from the **Service provider** drop-down list.



3. From the **Certificate store type** drop down list, select **Local machine store**.
4. In the **Certificate store** field, specify the certificate store name.

Below is an example of the dialog that opens if you select the store using the browse button.

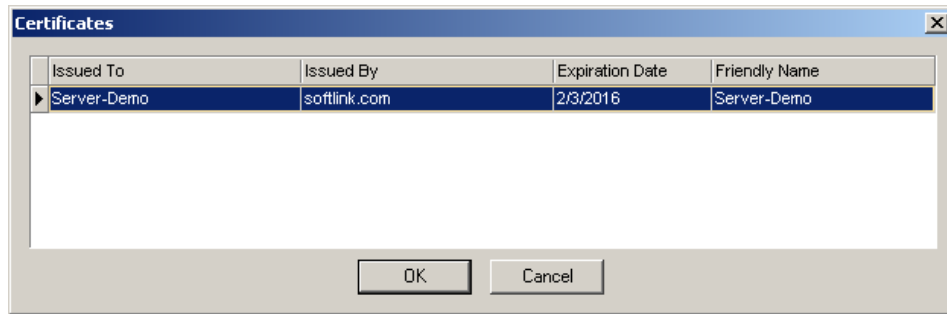
Note: If you imported your certificate to the **Personal** store, you can leave the <default> value.



5. In the **Certificate subject** field, specify the certificate subject.

Note: If you are using the demo certificate, you can leave the <default> value.

If you use the browse button, the **Certificates** dialog box opens.



Select the certificate and that you want to use, then click **OK**. The certificate's subject is displayed in the **Certificate subject** field.

6. In the **Cryptographic service provider** field, leave the <default> value or specify your cryptographic service provider, either manually or using the browse button.
7. In the **Key container name** field, leave the <default> value or specify the name of your key container.
8. [Optional] In the **CRL file** field, specify the pathname of the file containing certificates issued by, but that have since been revoked by the CA.
9. Click **Save**.
10. Select the **General** tab.
The **General** Tab is displayed.
11. In the **Daemon service** region, click the **Restart** button.
12. Click **Save & Exit**.

Using OpenSSL Certificates

You can test functionality using the OpenSSL compliant demo certificates provided with the installation.

In a standard installation, the demo certificates reside in `~\RepliWeb\RMFT\ssl`.

The demo server certificates are as follows:

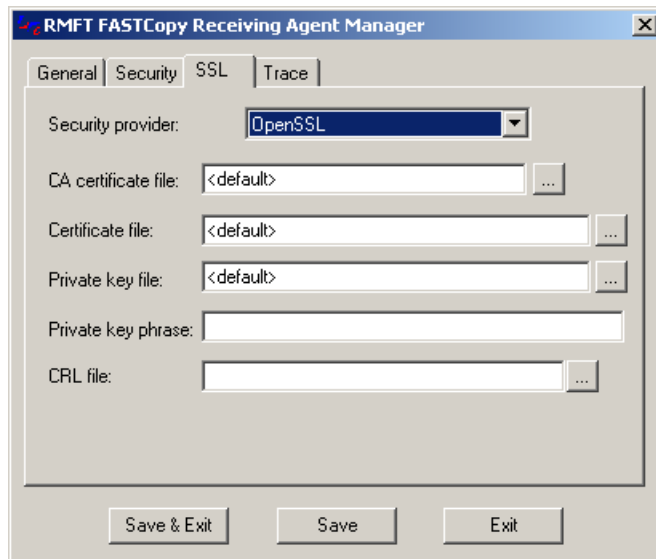
- **fc_server_cert.pem** (server certificate)
- **fc_server_key.pem** (server private key)
- **trusted_ca_cert.pem** (CA certificate)

The password for the server private key file is: **demosever**

Unless configured differently, RMFT FASTCopy Receiving Agent will authenticate itself using the demo OpenSSL certificates (provided with the installation).

Note: The default OpenSSL certificates provide no real security and are intended for testing purposes only.

1. Open RMFT FASTCopy Receiving Agent Manager as described in [Opening RMFT FASTCopy Receiving Agent Manager](#) above.
2. From the **Service provider** drop-down list, select **OpenSSL**.



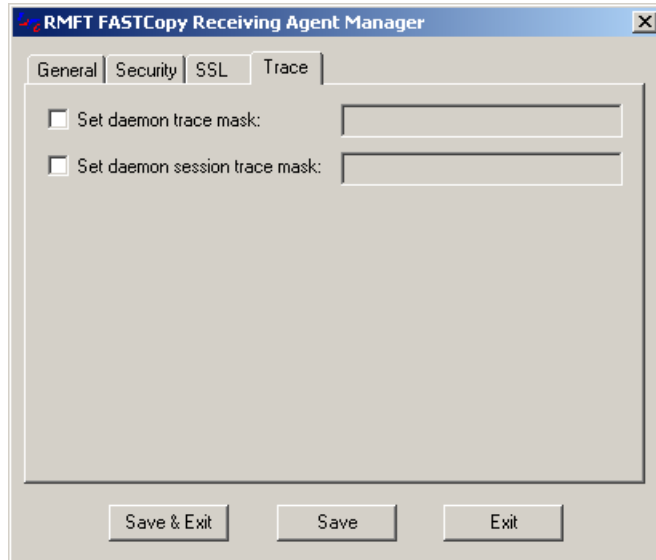
3. In the **CA certificate file** field, specify the location of your CA file.
4. In the **Certificate file** field, specify the location of your server certificate.

5. In the **Private key file** field, specify the location of your private key file.
6. In the **Private key phrase** field, specify the private key file's password.
7. [Optional] In the **CRL file** field, specify the pathname of the file containing certificates that have been revoked by their issuing CAs.
8. Click **Save**.
9. Select the **General** tab.
10. In the **Daemon service** region, click the **Restart** button.
11. Click **Save & Exit**.

Trace Tab

Traces are intended to determine the cause of seldom occurring RMFT FASTCopy Receiving Agent issues and should not be used without first consulting [RepliWeb Support](#).

1. Select the **Trace** tab.



2. To specify a daemon trace mask:
 - a. Create the directory **Logs** under `~\RepliWeb\RMFT\fastcopy`
 - b. Select the **Set daemon trace mask** check box and enter the value provided by RepliWeb Support in the adjacent field.

The next time a FASTCopy operation is performed, the trace file **fcopyd.log** will be created in `~\RepliWeb\RMFT\fastcopy\Logs`
3. To specify a daemon session trace mask, select the **Set daemon session trace mask** check box and enter the value provided by RepliWeb Support in the adjacent field.

The next time a FASTCopy operation is performed, the session trace file **nifcopy2.log** is created in `~\RepliWeb\RMFT\fastcopy`
4. Click the **Save & Exit** button to save your settings and close the RMFT FASTCopy Receiving Agent Manager.

Managing RMFT FASTCopy Receiving Agent Nodes

In the **Manage Nodes** dialog box, you can:

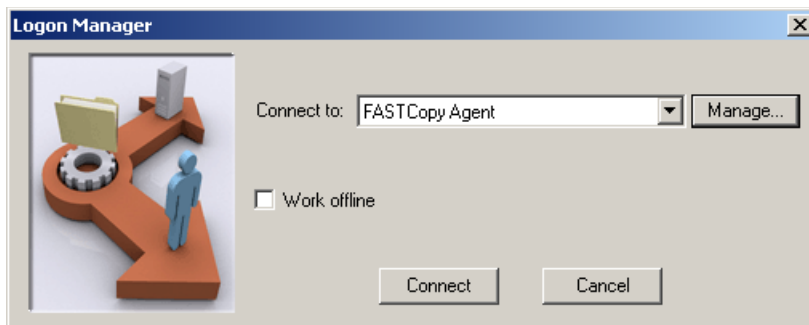
- Add RMFT FASTCopy Receiving Agent nodes
- Update the login settings of existing RMFT FASTCopy Receiving Agent nodes
- Remove RMFT FASTCopy Receiving Agent nodes

Adding RMFT FASTCopy Receiving Agent Nodes

To add an RMFT FASTCopy Receiving Agent node:

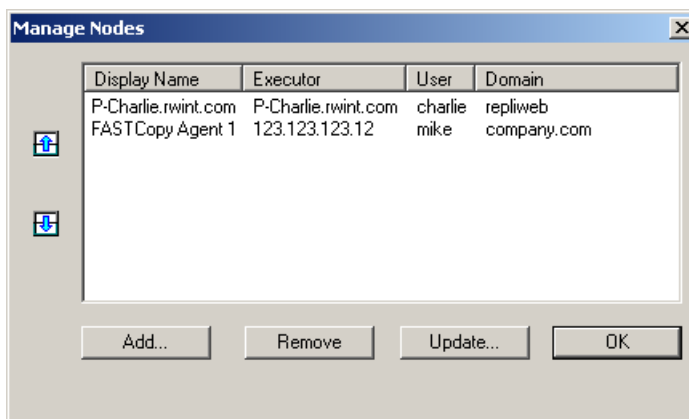
1. Open RMFT FASTCopy Receiving Agent Manager.

The **Logon Manager** dialog box opens.



2. Click the **Manage** button to the right of the **Connect to** field.

The **Manage Nodes** dialog box opens.



3. To add a new FASTCopy Server node, click **Add**.

The **New Node** dialog box opens.

The screenshot shows the 'New Node' dialog box with the following fields and options:

- Display name: [Text Input]
- Hostname/IP Address: [Text Input]
- Authentication Methods:
 - System credentials
 - User: [Text Input]
 - Password: [Text Input] Save password
 - Domain: [Text Input]
 - Integrated Windows Authentication
 - Certificate only
- Administrator Type:
 - Role: [Dropdown Menu: Administrator]
 - Group name: [Text Input]
- Secure Logon:
 - Never
 - Always log on securely
 - Only log on securely if required
 - Certificate subject: [Text Input] [Browse Button]
- Buttons: Save, Cancel

4. In the **Display name** field, enter a display name for the RMFT FASTCopy Receiving Agent machine.

Note: Multiple users can connect to the same server machine, providing that each user's logon settings contain a different display name and logon credentials.

5. In the **Hostname/IP Address** field, Enter the hostname or IP address of the RMFT FASTCopy Receiving Agent machine.
6. Choose one of the following login methods as appropriate:
 - **System credentials** to log in using a system user name and password. If you choose this method, provide your logon details in the designated fields.
 - Select the **Save password** check box to be able to open RMFT FASTCopy Receiving Agent Manager without being prompted for a password each time.
 - **Integrated Windows Authentication** to log in without providing a user name or password. If you select this method, continue from [Step 8](#).

Note: Certificate-based log in requests will only succeed if a valid server-side certificate exists. For detailed instructions on loading server certificates to facilitate SSL login, please refer to the *RMFT GQS Agent Configuration Guide*.

- **Certificate only** to log in using only a certificate. If you select this method, continue from [Step 8](#).

Note: Requests to log in using **Integrated Windows Authentication** will only succeed if the RMFT FASTCopy Receiving Agent machine has been set up to permit IWA access to the required folders.

Note: The RMFT GQS Agent can be configured to require administrators to log in to RMFT FASTCopy Receiving Agent Manager using a particular login method or methods. The default settings permit administrators to use any of the available login methods.

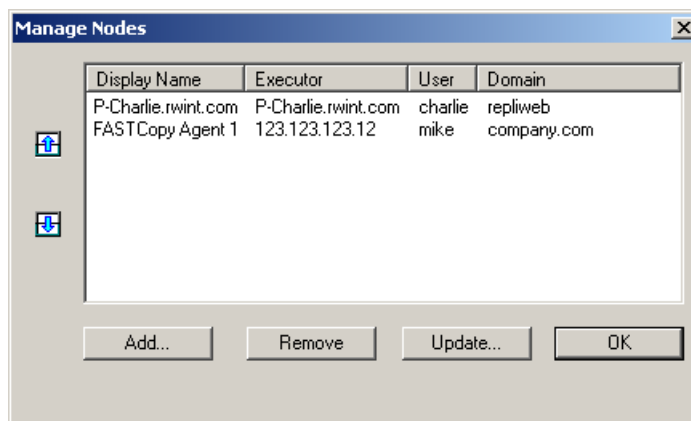
For detailed instructions on requiring administrators to use a particular login method or methods, please refer to the *RMFT GQS Agent Configuration Guide*.

7. Optionally, select one of the options in the **Secure Login** area.

Note If you selected the **Certificate only** login method described in [Step 6](#), these options will not be available. In this case, you must specify a client certificate in the **Certificate subject** field. If, however, you selected the **System credentials** or **Integrated Windows Authentication** login methods, you can optionally specify a certificate to authenticate the server's identity and encrypt all communications between the Logon Manager and the remote RMFT FASTCopy Receiving Agent Manager.

For more information on logging in securely (SSL), see [Secure Login Options](#).

8. Click **Save** to save your settings and add the RMFT FASTCopy Receiving Agent machine to the list displayed in the **Manage Nodes** dialog box.



9. Click **OK** to return to the **Logon Manager** dialog box.

The **Connect to** field shows the name of the *first* machine in the **Manage RMFT Nodes** dialog box.

Secure Login Options

This section describes the options available in the **Secure Logon** area of the **New Node** or **Update Logon Details** dialog boxes. Using the RMFT GQS Agent Manager, the RMFT administrator can determine whether a certificate is required to open the RMFT FASTCopy Receiving Agent Manager or whether other means of authentication (IWA/login credentials) are sufficient.

IMPORTANT: The settings selected in the **Secure Login** area of the **New Node** dialog box will only be applied if the RMFT GQS Agent is configured correctly. For an explanation of how to configure the GQS to enable or require SSL connections, please refer to the *RMFT GQS Agent Configuration Guide*.

If you are not required to log in to RMFT FASTCopy Receiving Agent Manager with a certificate or do not want to authenticate the identity of the RMFT FASTCopy Receiving Agent machine:

- ◆ Select the **Never** option.

To always log in to RMFT FASTCopy Receiving Agent Manager with a certificate, even if not required:

1. Select the **Always log on securely** option.
2. In the **Certificate subject** field, use the certificate browser to enter the full path of the server certificate file (OpenSSL) or specify a unique part of the certificate's subject (MSCAPI).

For an explanation of the certificate selection procedure, see either [Selecting OpenSSL Certificates](#) or [Selecting MSCAPI Certificates](#) as appropriate.

To log in securely to RMFT FASTCopy Receiving Agent Manager, only if required:

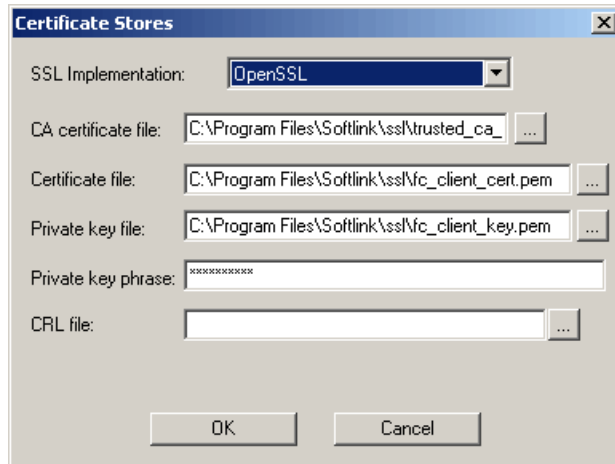
1. Select the **Only log on securely if required** option.
2. In the **Certificate subject** field, use the certificate browser to enter the full path of the server certificate file (OpenSSL) or specify a unique part of the certificate's subject (MSCAPI).

For an explanation of the certificate selection procedure, see either [Selecting OpenSSL Certificates](#) or [Selecting MSCAPI Certificates](#) as appropriate.

Selecting OpenSSL Certificates

To use OpenSSL:

1. Select **OpenSSL** from the **SSL Implementation** drop-down list.



2. In the **CA certificate file** field, specify the full path of the trusted Certification Authority file.
3. In the **Certificate file** field, enter the full path of your certificate file, either manually or using the browse button.
4. In the **Private key file** field, enter the full path of your private key file, either manually or using the browse button.
5. In the **Private key phrase** field, specify the pass phrase of the private key file.
6. (Optional) In the **CRL file** field, specify the full path of the file containing certificates issued by, but that have since been revoked by the selected CA.
7. Click **OK**.

The full path to your certificate is displayed.

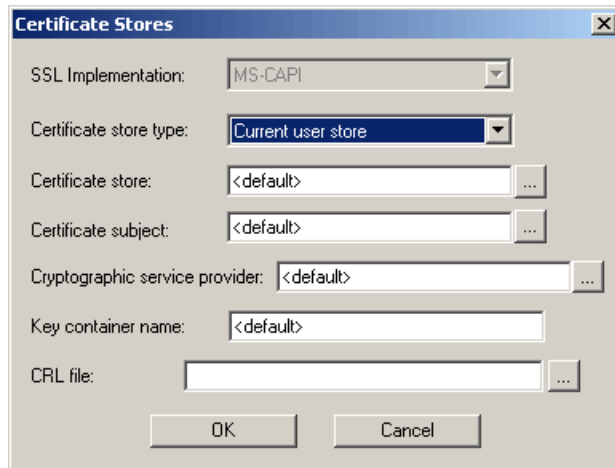
Selecting MSCAPI Certificates

If you want to use MSCAPI as your service provider, you must first perform the following steps:

- a. Import the MSCAPI client certificate to a local certificate store.
- b. Import the MSCAPI CA certificate to a local certificate store.

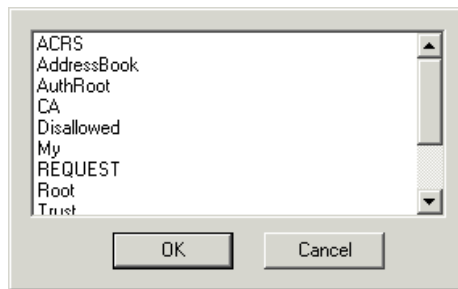
To select a certificate from a store:

1. Select **MS-CAPI** from the **SSL Implementation** drop-down list.



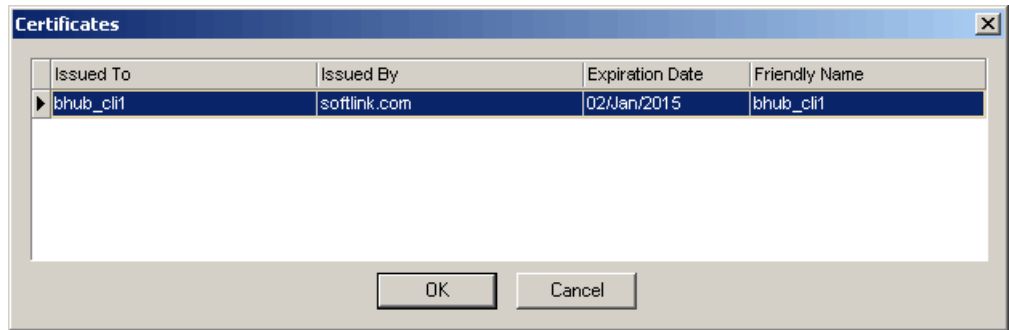
2. From the **Certificate store type** drop down list, select one of the following as appropriate:
 - Current user store <default>
 - Local computer store
 - Local enterprise store
3. In the **Certificate store** field, specify the certificate store, either manually or using the browse button.

If you use the browse button, a dialog similar to this one opens.



4. Select the store containing your certificate and then click **OK**.
5. In the **Certificate subject** field, specify the certificate subject, either manually or using the browse button.

If you use the browse button, the **Certificates** dialog box opens.



Select a certificate and click **OK**.

The certificate's subject is displayed in the **Certificate subject** field.

6. (Optional) In the **Cryptographic service provider** field, specify your cryptographic service provider, either manually or using the browse button.
7. (Optional) In the **Key container name** field, specify the name of your key container.
8. (Optional) In the **CRL file** field, specify the pathname of the file containing certificates issued by, but that have since been revoked by the selected CA.
9. Click **OK**.

The certificate's subject is displayed.

3. Starting RMFT FASTCopy Receiving Agent on UNIX

In a standard installation RMFT FASTCopy Receiving Agent is installed and run by default, enabling your node to respond to incoming FASTCopy requests from remote nodes. However, you may want to run additional standalone instances of RMFT FASTCopy Receiving Agent for testing purposes. This section explains how to install and run RMFT FASTCopy Receiving Agent from the command line, as well as how to change the default service and port used by RMFT FASTCopy Receiving Agent.

On UNIX machines, an additional daemon must be active so that the user can initiate and monitor FASTCopy operations in batch mode - the FASTLogic daemon, *flogicd*.

fcopyd comes into operation whenever a connection request for FASTCopy arrives from a remote node. *fcopyd* accepts the connection request and establishes a FASTCopy session with the remote node.

Unless *fcopyd* is either running or enabled, other nodes will be unable to initiate file transfer operations with your node.

Starting fcopyd on UNIX

There are two ways you can start *fcopyd* on UNIX:

1. Enabling *fcopyd* through *inetd*.
2. Starting *fcopyd* directly as a standalone daemon.

Enabling fcopyd through inetd

Add the following line to the *inetd* definition file - `/etc/inetd.conf`:

```
fcopyd$server stream tcp nowait root /usr/bin/fcopyd
in.fcopyd
```

If you install the symbolic links to FASTCopy executables in a directory other than `/usr/bin`, specify that directory instead of the default directory shown above.

To activate *fcopyd*, make *inetd* reread the configuration file by sending the *inetd* process the “hang-up” signal:

```
>kill -1 inetd_process_number
```

Starting fcopyd Directly as a Standalone Daemon

Issue the following command from a “root” account:

```
>fcopyd
```

The command can be modified with any of the qualifiers described in [fcopyd Qualifiers](#) (70), below.

If you choose to start *fcopyd* as a standalone daemon regularly, it is recommended to add the activation command to the startup files of your node.

xinetd Support for RedHat Linux 7.x

The FASTCopy installation script detects the xinetd Internet Superserver daemon on RedHat Linux 7.x. This enables RedHat 7.x users to have a working installation configured automatically (on acceptance of the installation defaults).

On machines running the *inetd* daemon, the *inetd* daemon is detected and configured automatically as before, on acceptance of installation defaults.

Example of RedHat 7.x FASTCopy Installation Prompts

```
Would you like RMFT FASTCopy Receiving Agent to be started by 'inetd'
[no] ?

Would you like RMFT FASTCopy Receiving Agent to be started by 'xinetd'
[yes] ?
```

The installation script creates the following file: `/etc/xinetd.d/fcopyd`

The contents of the **fcopyd** file are as follows:

```
service fcopydserver
{
```

```
    flags      = REUSE_NAMEINARGS
    protocol   = tcp
    socket_type = stream
    wait       = no
    server     = /usr/flogic/bin/fcopyd
    user       = root
    server_args = in.fcopyd
}
```

To start RMFT FASTCopy Receiving Agent service from the command line issue the command:

```
> fcopyd -start
```

To stop RMFT FASTCopy Receiving Agent service from the command line issue the command:

```
> fcopyd -stop
```

fcopyd Qualifiers

You can specify the following qualifiers with the `fcopyd` command:

- | | |
|--|--|
| <code>-log_file=</code> <i>file_name</i> | A log file for <i>fcopyd</i> 's messages (if not specified, the daemon's messages are not reported). |
| <code>-port=</code> <i>port_number</i> | If you want FASTCopy to accept connections on a different port than the default set by the installation. |
| <code>-service=</code> <i>service_name</i> | If you want FASTCopy to use a different network service than the default set by the installation. |

A. Loading Certificates on UNIX/Linux Platforms

On UNIX platforms, the demo certificates reside in: `/usr/flogic/ssl`

You can load RMFT FASTCopy Receiving Agent with OpenSSL certificates, either manually using a command line or automatically on startup. The procedures differ according to the platform on which RMFT FASTCopy Receiving Agent is installed. On all UNIX/Linux platforms except **Solaris on SPARC** and **HP-UX on PA-RISC**, perform the two-stage procedure described below.

For instructions relevant to Solaris on SPARC and HP-UX on PA-RISC continue from [Solaris on SPARC and HP-UX on PA-RISC](#).

Stage 1: Preventing RMFT FASTCopy Receiving Agent from being Loaded

Stage one prevents `xinetd` (Linux) / `inetd` (UNIX) from loading RMFT FASTCopy Receiving Agent and should only be performed if you opted to start RMFT FASTCopy Receiving Agent automatically during installation. If `xinetd/inetd` is not configured to load RMFT FASTCopy Receiving Agent, proceed to stage two.

On Linux Machines:

To prevent `xinetd` from loading RMFT FASTCopy Receiving Agent:

1. Edit the file:
`/etc/xinetd.d/fcopyd`
2. Change the disable value to:
`disable = yes`
3. Save the file.
4. Restart `xinetd` by issuing the following command:
`service xinetd restart`

After disabling `xinetd`, the only way to load RMFT FASTCopy Receiving Agent is by issuing the appropriate command as described in stage two below.

On UNIX Machines (except [Solaris10](#)):**To prevent inetd from loading RMFT FASTCopy Receiving Agent:**

1. Edit the file:
`/etc/inetd.conf`
2. Disable RMFT FASTCopy Receiving Agent by commenting out the following row:

```
# fcopy$server stream tcp nowait root  
/usr/flogic/bin/fcopyd in.fcopyd
```
3. Save the file.
4. Restart `inetd`.

After disabling `inetd`, the only way to load RMFT FASTCopy Receiving Agent is by issuing the appropriate command as described in stage two below.

On Solaris10:**To prevent the RMFT FASTCopy Receiving Agent from being loaded:**

- ◆ Issue the following command:

```
svcadm disable svc:/network/fcopy_server/tcp:default
```

After issuing the above command, the only way to load RMFT FASTCopy Receiving Agent is by issuing the appropriate command as described in stage two below.

Stage 2: Issuing a Command to Load RMFT FASTCopy Receiving Agent with the Required Certificates

To load the RMFT FASTCopy Receiving Agent on machine startup, embed the daemon loading command in the machine's startup scripts.

Command Syntax:

```
fcopyd -certificate=<PATH_TO_CERTIFICATE_FILE> -key=<PATH_TO_KEY_FILE>  
-key_phrase=<KEY_PHRASE> -ca_file=<PATH_TO_CA_FILE>
```

Example:

```
fcopyd -certificate=/usr/flogic/ssl/fc_server_cert.pem
-key=/usr/flogic/ssl/fc_server_key.pem -key_phrase=demoserver
-ca_file=/usr/flogic/ssl/trusted_ca_cert.pem
```

IMPORTANT The example above uses RepliWeb's demo certificates and keys, which offer no real security in so far as the private keys are widely known. Therefore, they should only be used for demonstration/testing purposes.

Solaris on SPARC and HP-UX on PA-RISC

On **Solaris on SPARC** and **HP-UX on PA-RISC** platforms, RMFT FASTCopy Receiving Agent's SSL settings can be saved in an .ini file. Using this method, RMFT FASTCopy Receiving Agent will apply the SSL settings defined in the .ini file each time it initializes.

This provides the following advantages:

- It eliminates the need to specify FASTCopy SSL qualifiers every time RMFT FASTCopy Receiving Agent is loaded, which also means that you can still use inetd/xinetd on Linux/UNIX or services on Solaris 10 to start RMFT FASTCopy Receiving Agent.
- The password for the private key is saved encrypted in the .ini file and not specified in clear text in the command line.

To create and configure the .ini file, issue the following command:

Syntax:

```
fcopyd -configure -certificate=<PATH_TO_CERTIFICATE_FILE>
-key=<PATH_TO_PRIVATE_KEY_FILE> -key_phrase=<PRIVATE_KEY_PHRASE>
-ca_file=<PATH_TO_CA_FILE> -ca_dir=<DIRECTORY_CONTAINING_CA_FILES>
```

Example:

```
fcopyd -configure -certificate=/usr/flogic/ssl/intm-mftc_cert.pem
-key=/usr/flogic/ssl/intm-mftc_key.pem -key_phrase=XXXX
-ca_file=/usr/flogic/ssl/trusted_ca_cert_dfs.pem -ca_dir=/usr/flogic/ssl
```

The .ini file will always be created under the installation root directory (e.g. /usr/flogic) and named **fcopyd.ini**. After the file is created, you can edit it with a text editor or run the `-configure` command again to modify the SSL settings.

Note: The command only has to be issued once. After the command has been run, RMFT FASTCopy Receiving Agent will apply the SSL settings defined in the .ini file each time it initializes.

B. Importing MSCAPI Server Certificates

On the RMFT FASTCopy Receiving Agent machine, the server and CA certificates must be imported using the Microsoft Management Console. After importing the certificates you must configure RMFT FASTCopy Receiving Agent to load the *server* certificate (so that it can authenticate itself to the client).

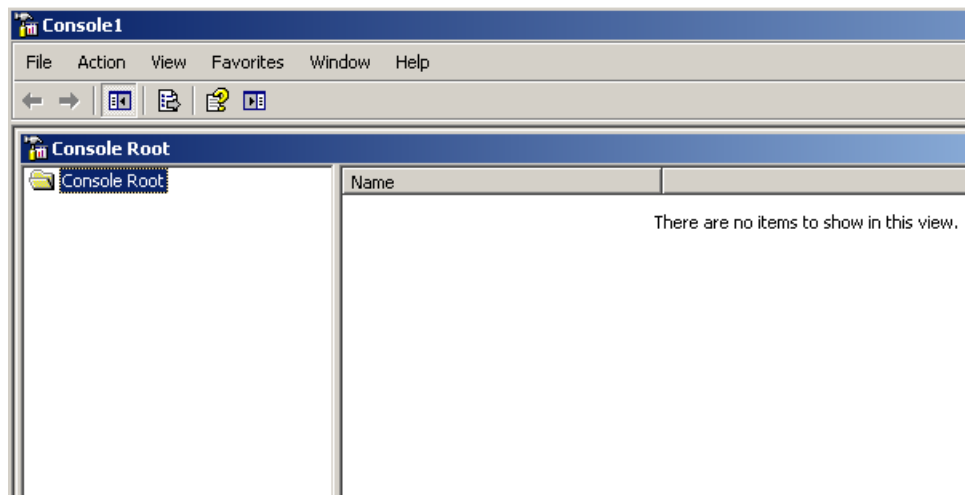
See also [Using MSCAPI Certificates](#).

Importing the Server Certificate

To import the server certificate:

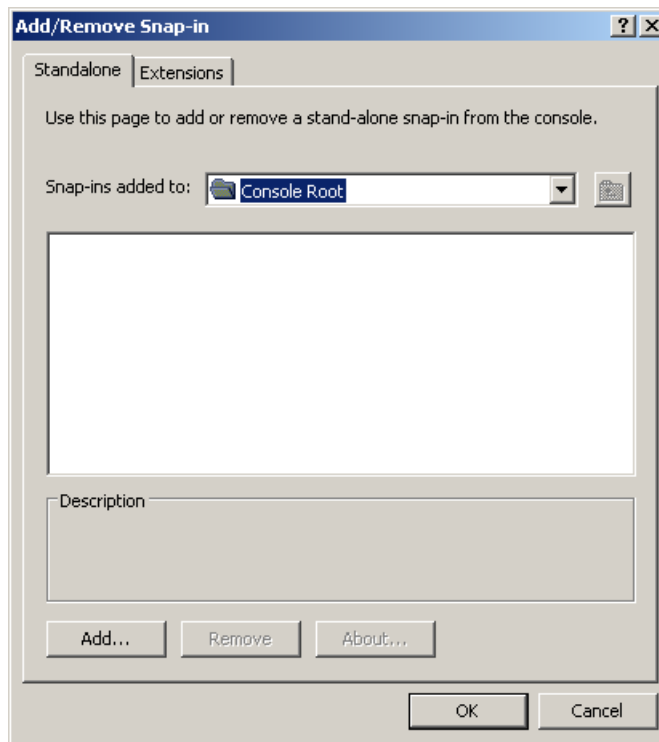
1. Select **Run** from the Windows **Start** menu.
The **Run** dialog box opens.
2. Type MMC (Microsoft Management Console) in the **Open** field and click **OK**.

The **Console** window opens.



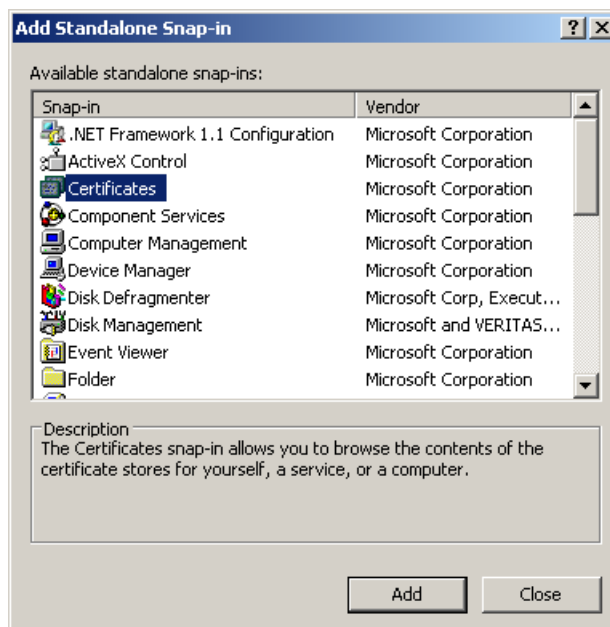
3. From the **File** menu, select **Add/Remove Snap-in**.

The **Add/Remove Snap-in** dialog box opens.



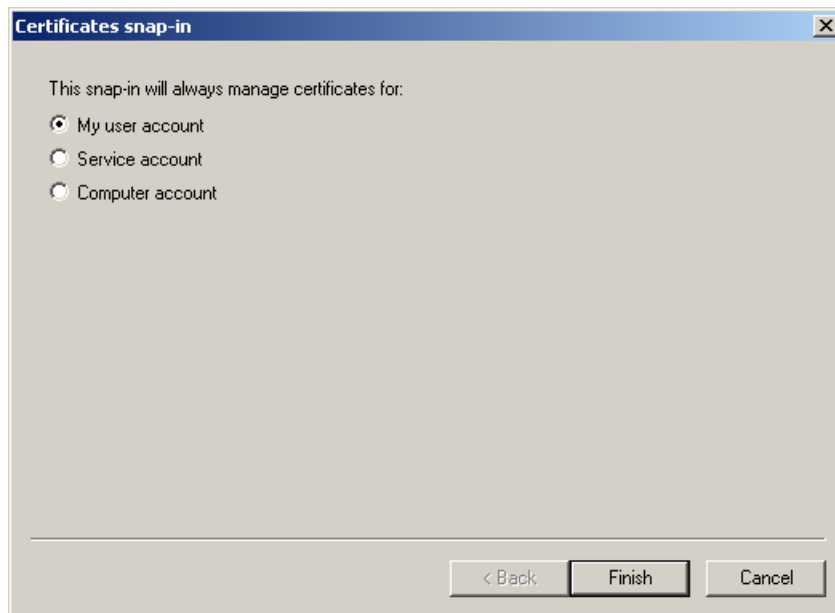
4. Click the **Add** button.

The **Add Standalone Snap-in** dialog box opens.



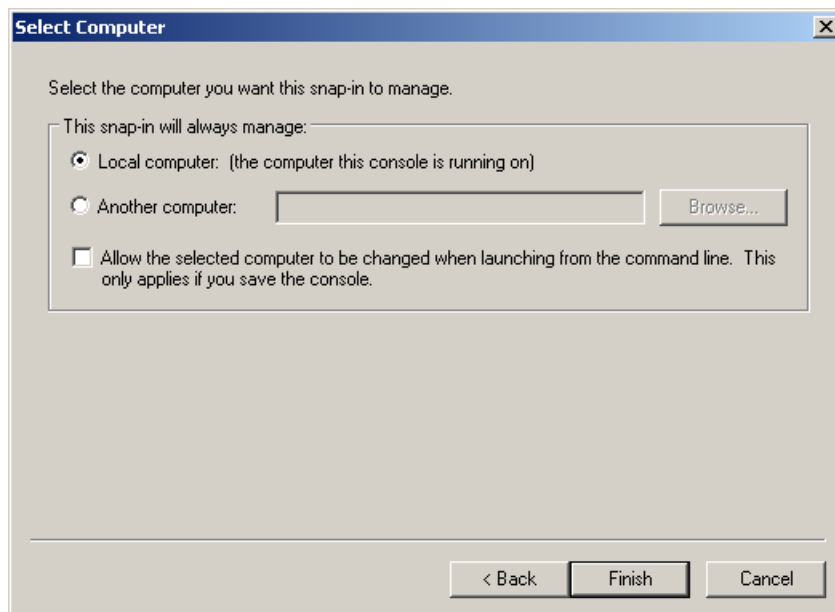
5. Select **Certificates** from the **Snap-in column** and then click **Add**.

The **Certificates Snap-in** dialog box opens.



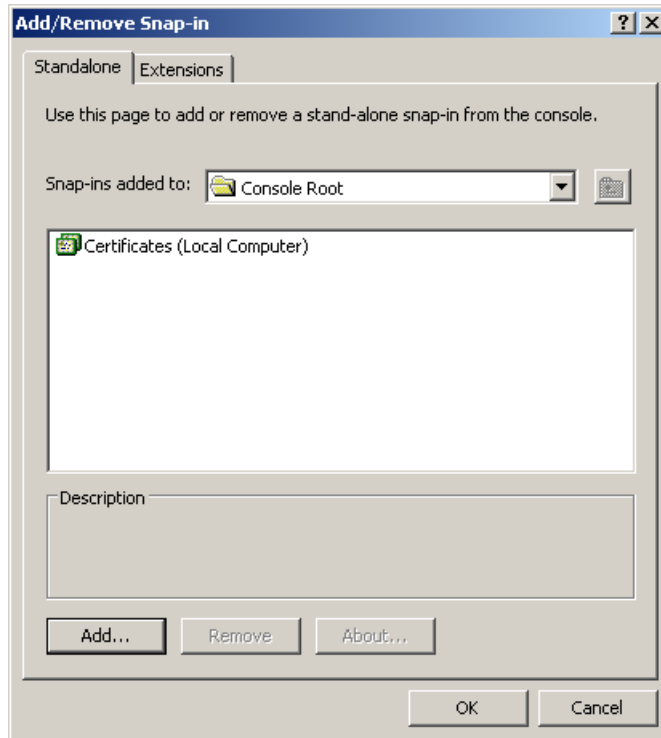
6. Select the **Computer account** radio button and then click **Next**.

The **Select Computer** dialog box opens.



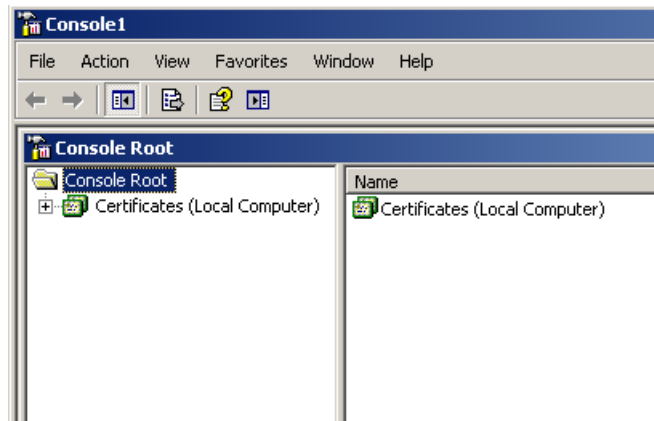
7. Select the **Local computer**, click **Finish** and then close the **Add Standalone Snap-in** dialog box .

The **Add/Remove Snap-in** dialog box displays the newly added **Certificates** snap-in.



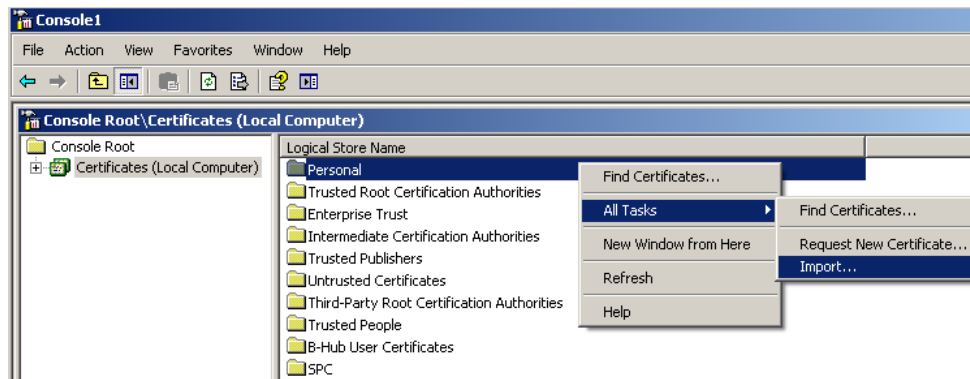
8. Click **OK**.

The **Console Root** window displays the **Certificates** snap-in.



9. Double-click **Certificates** in the right pane of the **Console Root** window.

The certificate stores are displayed in the **Logical Store Name** column.

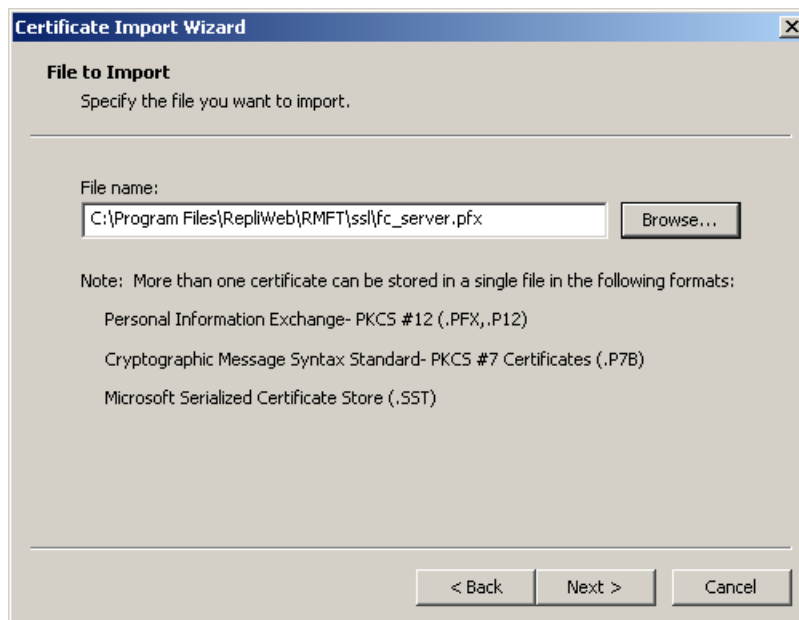


10. Right-click the **Personal** store and select **All Tasks > Import...** as shown above.

The **Certificate Import Wizard** opens.

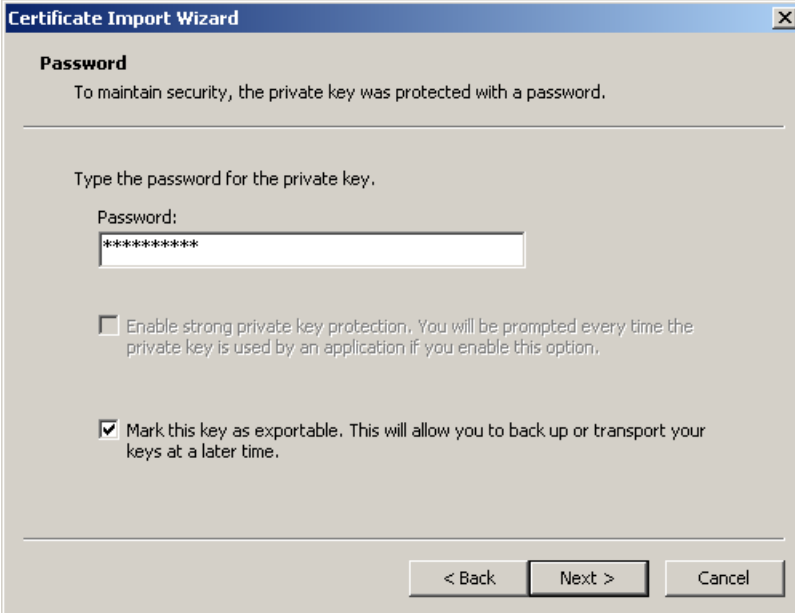
11. Click **Next**.

The **File to Import** dialog box is displayed.



12. In the **File name** field, specify the location of your server certificate.
13. Click **Next**.

14. The **Password** dialog box opens.



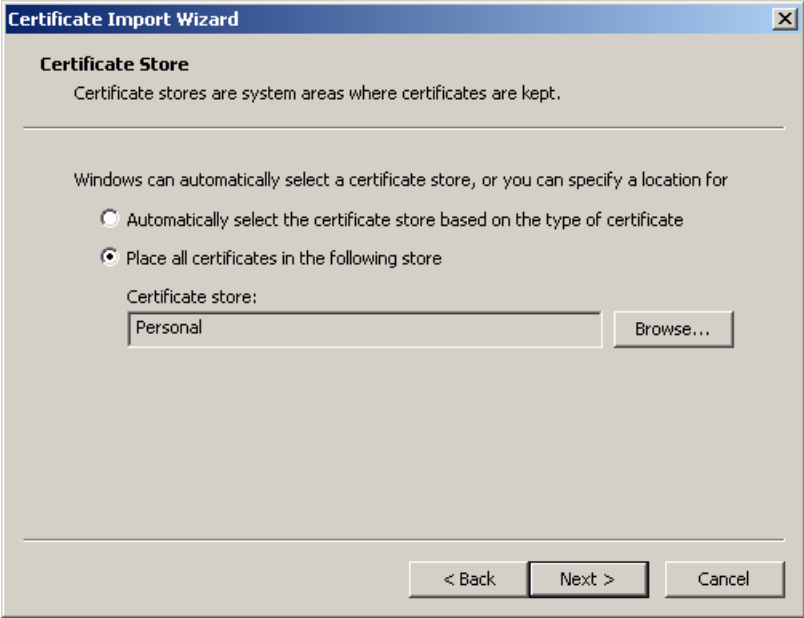
The image shows a dialog box titled "Certificate Import Wizard" with a close button (X) in the top right corner. The main heading is "Password". Below the heading, it says "To maintain security, the private key was protected with a password." There is a horizontal line. Below that, it says "Type the password for the private key." There is a "Password:" label followed by a text input field containing "*****". Below the input field, there are two checkboxes. The first checkbox is unchecked and has the text "Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option." The second checkbox is checked and has the text "Mark this key as exportable. This will allow you to back up or transport your keys at a later time." At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

15. In the **Password** field, type the password for the private key.

16. Select the **Mark this key as exportable** check box and then click **Next**.

The **Certificate Store** dialog box opens.

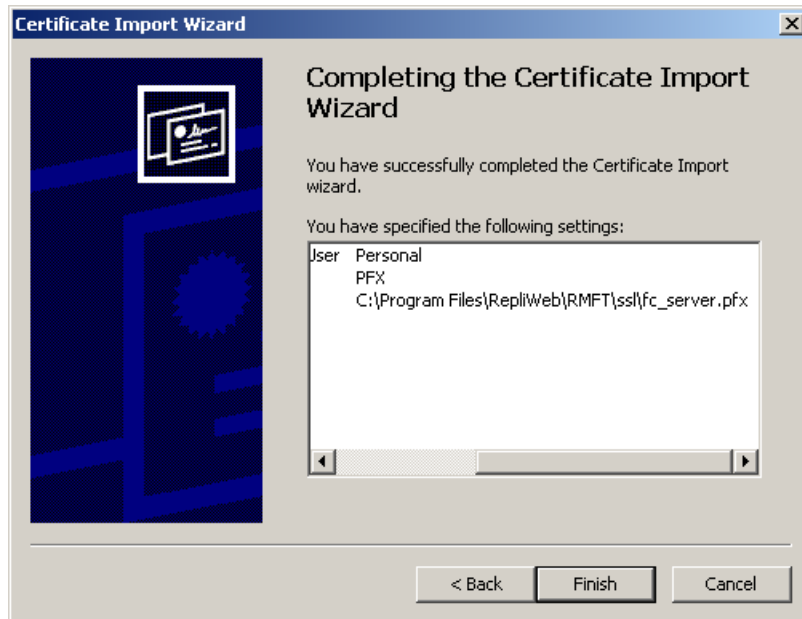
Personal is displayed in the **Certificate store** field.



The image shows a dialog box titled "Certificate Import Wizard" with a close button (X) in the top right corner. The main heading is "Certificate Store". Below the heading, it says "Certificate stores are system areas where certificates are kept." There is a horizontal line. Below that, it says "Windows can automatically select a certificate store, or you can specify a location for". There are two radio buttons. The first radio button is unselected and has the text "Automatically select the certificate store based on the type of certificate". The second radio button is selected and has the text "Place all certificates in the following store". Below the radio buttons, there is a "Certificate store:" label followed by a text input field containing "Personal" and a "Browse..." button. At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

17. Click **Next**.

The **Completing the Certificate Import Wizard** dialog box appears



18. Click **Finish**.

The **Certificate Import Wizard** closes and a message box confirms that the certificate import was successful.

Importing the CA Certificate

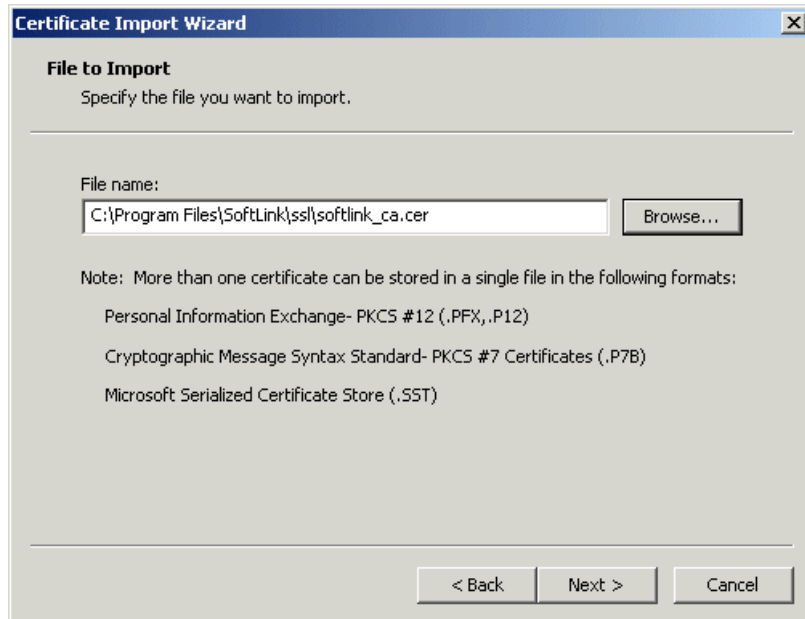
To import the CA certificate:

1. In the **Logical Store Name** column, right-click the **Trusted Root Certification Authorities** store and select **All Tasks > Import**.

The **Certificate Import Wizard** opens.

2. Click **Next**.

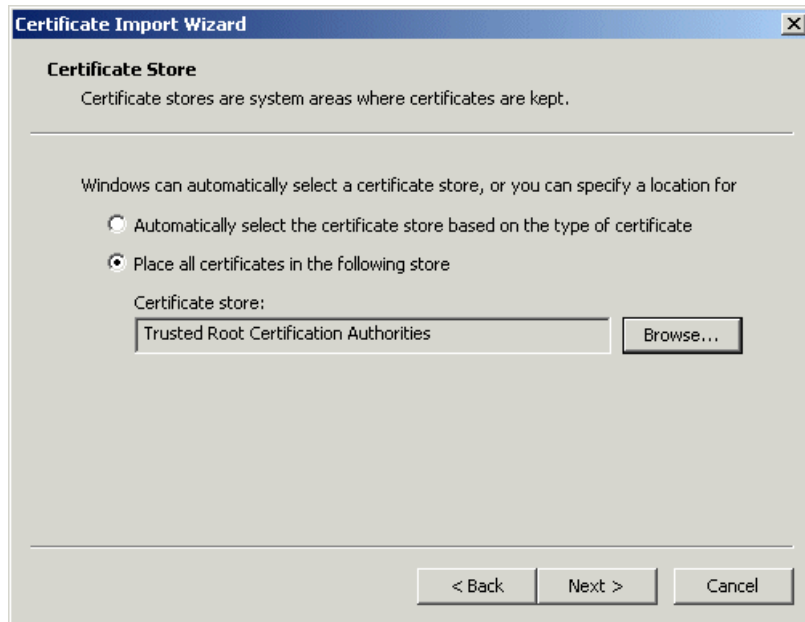
The **File to Import** dialog box is displayed:



3. In the **File name** field, specify the pathname of your CA certificate and then click **Next**.

The **Certificate Store** dialog box opens.

Trusted Root Certification Authorities is displayed in the **Certificate store** field.



4. Click **Next**.

The **Completing the Certificate Import Wizard** dialog box appears.



5. Click **Finish**. The **Certificate Import Wizard** closes and a message should confirm that the certificate import was successful.

INDEX

- Actions Verb, 23
 - activity_time qualifier, 28
 - approve qualifier, 23
 - class qualifier, 23
 - dir create qualifier, 24
 - inp_dir qualifier, 24
 - inp_include_sub_dirs qualifier, 24
 - l_user qualifier, 23
 - line_cipher qualifier, 25
 - line_encrypt qualifier, 25
 - line_phrase qualifier, 25
 - mac qualifier, 26
 - mon_level qualifier, 27
 - mon_must qualifier, 27
 - mon_node qualifier, 27
 - mon_type qualifier, 27
 - monitor qualifier, 26
 - out_dir qualifier, 24
 - out_include_sub_dirs qualifier, 24
 - overwrite qualifier, 24
 - reason qualifier, 28
 - recurs qualifier, 28
 - remote_command qualifier, 23
 - root_dir qualifier, 24
 - size_limit qualifier, 24
 - sl_security qualifier, 28
- Advanced Definer
 - SSL
 - session cipher, 57, 59
- Certificate
 - importing an OpenSSL CA certificate, 65
 - importing an OpenSSL certificate, 65
 - importing an OpenSSL private certificate, 65
 - MSCAPI certificate, CRL file, 67
 - MSCAPI certificate, cryptographic service provider, 67
 - MSCAPI certificate, key container name, 67
 - MSCAPI certificate, store location, 66
 - MSCAPI certificate, store type, 66
 - MSCAPI certificate, subject, 66
 - OpenSSL CRL file, 65
 - password for OpenSSL private certificate, 65
- Criteria Verb, 16
 - class qualifier, 16
 - code qualifier, 16
 - control qualifier, 16
 - hook qualifier, 17
 - incoming qualifier, 18
 - l_application qualifier, 18
 - l_password qualifier, 18
 - l_user qualifier, 18
 - object qualifier, 19
 - operation qualifier, 18
 - outgoing qualifier, 19
 - password qualifier, 19
 - peer_address qualifier, 20
 - peer_net_mask qualifier, 20
 - peer_node qualifier, 20
 - purge qualifier, 20
 - r_application qualifier, 20
 - read qualifier, 21
 - target_group qualifier, 21
 - target_node qualifier, 21
 - target_user qualifier, 21
 - time_frame qualifier, 21
 - write qualifier, 21
- Domain
 - RMFT FASTCopy Receiving Agent Node, connecting to, 51
- FASTCopy Daemon
 - Daemon port number, changing the, 52
 - Daemon service, refreshing the, 52
 - Daemon service, restarting the, 52
 - Daemon service, starting the, 52
 - Daemon service, stopping the, 52
 - qualifiers
 - for changing default port, 70
 - for changing default service, 70
 - for creating daemon log file, 70
- RedHat 7.x, 69
- Security
 - daemon session trace mask, setting a, 60
 - daemon trace mask, setting a, 60
 - starting as a standalone
 - on a regular basis, 69
 - Starting automatically, 51, 52
 - starting on UNIX, 68
 - as a standalone, 69
 - through inetd, 68
 - starting on Windows, from command line, 70
 - stopping on Windows, from command line, 70
- Groups
 - using to designate batch jobs, 41
- Ifnot Verb, 30
 - compulsory use of, 30
 - restricting users with, 30
- Logical Passwords
 - adding a user, 45
 - creating and managing, 43
 - removing a user, 45
- Logical Usernames and Passwords
 - defining, 39
- Logon Method
 - selecting a, 62

- using Integrated Windows Authentication, 62
 - using only a certificate, 63
 - using system credentials, 62
- Non Trusted Networks
 - authenticating requests from, 40
 - encrypting data, 40
- On Verb, 33
 - address qualifier, 33
 - net_mask qualifier, 33
 - node qualifier, 33
 - example of usage, 34
 - using on multiple nodes, 33
- Open network
 - protecting against malicious users, 39
- Outgoing Operations
 - controlling with the outgoing qualifier, 41
- Password
 - RMFT FASTCopy Receiving Agent Node,
 - connecting to, 51
 - Saving, 51
- Proxy Security Mechanism
 - checking login parameters, 36
 - implementation of, 36
 - restricting remote requests with, 36
 - security file examples, 37
 - types of request
 - differentiating between, 37
 - types of security check, 36
- Proxy Security Mechanism
 - active security
 - controlling data transfer, 7
 - protecting sensitive information, 7
 - restricting node access, 7
 - active security principle, 7
 - file level security files
 - names on UNIX, 9
 - general rule based file
 - location on UNIX, 8
 - roles of, 6
 - rule base file
 - login, 9
 - rule base files, 8
 - security check
 - events during, 7
 - modifying operation, 7
 - security file hierarchy, 10
 - security files
 - general vs specific, 10
- Requester Verb
 - group qualifier, 15
 - node qualifier, 15
 - user qualifier, 15
- RMFT FASTCopy Receiving Agent Node
 - Connecting to a, 50
 - Selecting a, 50
- RMFT Manager
 - using GQS to force certificate-based login, 64
- RMFT Server
 - adding a, 61
 - connection settings, defining, 62
 - managing server nodes, 61
- RMFT Server Login
 - always log in securely, 64
 - always log in with certificate, 64
 - log in securely if required, 64
 - never log in securely, 64
 - secure login options, 63
- RMFT Server login password, saving the, 62
- Search Verb, 31
 - ignore_defaults qualifier, 31
 - modifying default search with, 31
 - stop_search qualifier, 31
- Security and Administration file
 - record
 - structure of, 12
 - structure of, 12
- Security and Administration Files
 - mandatory verbs, 13
 - verbs
 - actions, 13
 - criteria, 13
 - ifnot, 13
 - on, 13
 - requester, 13
 - search, 13
- Security File
 - parsing action, 13
 - parsing multiple records
 - order of precedence, 13
- Security File Rules
 - using to screen an incoming request, 39
- sl_passwd Utility, 39
- Specific Files
 - protecting access to, 40
- SSL Qualifiers
 - authenticate, 46
 - in login security file, 46
 - peer_common_name, 46
 - usage examples, 47
 - using to authenticate users, 46
- Trusted network
 - lifting security restrictions within, 38
- Variable
 - for group name, 34
 - for local user name, 35
 - for node name, 35
 - for peer node name, 35
 - for user name, 34
- Variables
 - usage example, 35
 - using in records, 34