OPENTRUST

OpenTrust MFT 3.3.0 System Maintenance Guide

OpenTrust MFT 3.3.0 System Maintenance Guide

Release Date: 2015-04-28
Revision: r149906

OpenTrust
175 rue Jean-Jacques Rousseau
CS 70056
92138 Issy-les-Moulineaux Cedex
France
www.opentrust.com

Contents

This page intentionally left blank.

Preface

The following sections contain preface information:

1. Related Documentation

2. Resources

Please use the information provided to contact the appropriate OpenTrust department or representative.

2.1. Contact Support

Support Web Site, including the Support Download Site	`https://support.opentrust.com/` (Login requires a username and password)
Email	`support@opentrust.com`

2.2. Contact Professional Services

Email	`support@opentrust.com`

2.3. Provide Documentation Feedback

As part of an ongoing process to create documentation that is easy to understand and use, as well as relevant to audience roles as administrator users, we welcome feedback about this guide. Please email any comments or suggestions to: `documentation_feedback@opentrust.com`

3. Document Conventions

OpenTrust documentation uses typographical conventions with specific meanings. These conventions are described in the following table.

Convention	How It Is Used
bold	Indicates the most important part of a step in step-based instructions. Example: Click the **OK** button.
italic	Indicates a reference to another document or guide. Example: See the *Release Notes*. Indicates the name of an access right. Example: The *unlock* right allows an administrator to help an end user unlock a smart card.
`monospaced font`	Indicates a file name, directory name or path, code examples and elements, application output, and user-entered text. Example: Save the file in the `/webserver` directory.
`italicized monospaced font`	Indicates an environment-specific or implementation-specific variable. Example: Save the file in the *`root_directory`*`/webserver` directory.
Important:	Contains important information that must be paid attention to. Failure to do so may have a negative impact on the application.
Note:	Contains valuable supplementary information.
Tip:	Contains helpful information that may be useful, for example, a shortcut or another way of performing a task.

This page intentionally left blank.

1 Architecture

The architecture of the OpenTrust MFT application includes:

1.1. Components

The OpenTrust MFT application is composed of:

- one or several Tomcat Java application servers
- one or several Apache Web servers, relaying user requests to the Tomcat servers
- one database PostgreSQL server, which is referred to as the SQL server
- one NFS file server which is referred to as the NFS server.

  The NFS server is optional and not required if all the Java application servers are located on the same host.
- one optional SMTP Connector, which receives emails with the SMTP protocol and transforms an SMTP MIME message into an OpenTrust MFT message, which can then be retrieved by its recipients through the OpenTrust MFT end user Webapp.

As described in the *OpenTrust MFT Server Installation and Upgrade Guide*, these components can be hosted on the same server or on multiple host servers.

In all possible server architectures, at least two Java applications run on at least one Tomcat server:

- the Administration application
- the End user application

An Apache Web server dedicated to Administration application access is always hosted on the same server as the Administration application. Thus:

- the term "Administration application" refers to both the Java application and the Apache Web server bundled together on the same machine
- the term Web server refers to the Apache Web server(s) relaying user requests to the end user Java application server(s)

Note: In single-host installations, all topics concerning NFS are not relevant and can be safely ignored

1.1.1. Communication Protocols

The following two tables display the protocols the OpenTrust MFT application uses for:

- communication between internal components of OpenTrust MFT

- communication with the external environment

Table 1.1. Communication Protocols Used Internally by the OpenTrust MFT Application

Source	Destination	Port Number	Protocol	Description
Administration application	SQL (PostgreSQL)	5432	PGSQL (TCP)[1]	Database access
Administration application	NFS (nfsd)	2049 (nfsd, standard port), 111 (portmap, standard port), 9049 (rpc.mountd, arbitrarily fixed port), 32803 (nfslockd, arbitrarily fixed port), 662 (statd, arbitrarily fixed port) [2]	NFS/RPC (both TCP and UDP!)	Access to the NFS server when theme files are uploaded
Administration application	End user application (Tomcat)	8980	HTTP (TCP)	Internal management (e.g. reread of configuration upon config change)
End user application	SQL (PostgreSQL)	5432	PGSQL (TCP)[1]	Database access
End user application	NFS (nfsd)	See NFS settings above[2]	NFS/RPC (both TCP and UDP!)	Access to the NFS server when uploading/ downloading files
Web server	End user application (Tomcat)	8900	AJP	Relaying of user requests from the frontal Web server to the Tomcat End user application
Web server [3]	Administration application (Tomcat)	8900	AJP	Optional: relaying of administrators' requests from the end user Web server to the Tomcat Administration application
SMTP Connector [4]	End user application	443	HTTPS (TCP)	Optional: emails transformed into OpenTrust MFT messages by the SMTP Connector are sent to the End user application

[1] If a content-filtering firewall controls this connection, the firewall must be configured without stateful packet inspection.

[2] For simplicity purposes, it is assumed that the NFS server is configured so that the ports picked by Portmap in a usually random manner are set to arbitrarily fixed values. See the `/etc/sysconfig/nfs` configuration file to fix the ports used by NFS.

[3] This connection is optional; it is only needed if access to the Administration application through the end user Web server is requested.

[4] This connection is optional; it is only needed if the SMTP Connector is installed and configured.

These protocols are also used by the OpenTrust MFT application to communicate with the external environment:

Table 1.2. Communication Protocols Used by the OpenTrust MFT Application to Communicate with the External Environment

Source	Destination	Port Number	Protocol	Description
NFS, Web server, Administration application, End user application, SQL	SMTP server	25	SMTP (TCP)	email sending (no reception needed)
NFS, Web server, Administration application, End user application, SQL	NTP server	123	NTP (UDP)	time synchronization
NFS, Web server, Administration application, End user application, SQL	DNS server	53	DNS (UDP)	Domain Name Service

Source	Destination	Port Number	Protocol	Description
End user application	db.local.clamav.net	80	HTTP (TCP)	ClamAV anti-virus updates. Warning: several public IP addresses are associated with the name “db.local.clamav.net” to ensure a high-availability of the ClamAV updates site. Not applicable if proxy server is used.
Web server, Administration application	PKI server (hosting CRLs)	80	HTTP (TCP)	CRL download, necessary only if X509 client certificate authentication is configured
Administration application, End user application [1]	Corporate LDAP Server	389	LDAP	Optional, required only when authenticating users to an external corporate LDAP directory
End User Desktop	Web server	443	HTTPS (TCP)	End user access to the End user application
End User Desktop	Web server	80	HTTP (TCP)	End user access to the End user application (redirect HTTP requests to HTTPS ones)
Administrator Desktop	Administration application	443	HTTPS (TCP)	Administration application access for administrators
Administrator Desktop	Administration application	80	HTTP (TCP)	Administration application access for administrators (redirect HTTP requests to HTTPS ones)
Corporate SMTP gateway [2]	SMTP Connector	25	SMTP (TCP)	Optional: Corporate SMTP gateway access to the SMTP Connector

[1]This connection is optional; it is only needed if the OpenTrust MFT application is configured so that authentication is delegated to a corporate LDAP directory (such as Microsoft Active Directory).

[2]This connection is optional; it is only needed if the SMTP Connector is installed and configured so that corporate emails are relayed to OpenTrust MFT .

1.1.2. Directory Structure

When the OpenTrust MFT server application is installed, the installer creates a directory structure containing the files for:

- “OpenTrust Core” on page 9
- “OpenTrust MFT” on page 10

1.1.2.1. OpenTrust Core

The OpenTrust Core framework is installed in the `/opt/opentrust/mft/otc` directory and contains common libraries and binaries.

Directory	Description and Sub-directories
`lib`	OpenTrust Core libraries: • `perl`: Perl libraries • `php`: PHP libraries • `shell`: Shell libraries
`share`	Collection of image files and JavaScript library.
`bin`	Executable scripts, internal use only

1.1.2.2. OpenTrust MFT

The OpenTrust MFT application is installed in the `/opt/opentrust/mft` directory, which contains:

- Configuration files
- Application data (exchanged files and database)
- Libraries
- CGI and PHP scripts (Administration application only)

Within this directory, sub-directories are organized as follows:

Directory	Description and Sub-directories
`etc`	Configuration files of OpenTrust MFT application: • `mft.conf`: main configuration file • `cron.d`: configuration for cron tasks • `logrotate.d`: configuration files for log files rotation • `init.d`: init script
`lib`	OpenTrust MFT application libraries: • `cryptonit`: file encryption binaries (End user application only) • `mft-clamscan`: wrapper scripts of the ClamAV anti-virus (End user application only) • `perl`: Perl libraries required by the OpenTrust MFT administrative tasks (init/backup/cron scripts, etc.) • `php`: PHP libraries required to display the audit logs interface (administrator application only) • `modules`: OpenTrust MFT application modules libraries
`sbin`	Collection of scripts: some are called internally by cron; others can be called manually for administration purpose (see below “Support Scripts” on page 11).
`var`	Application and runtime data, such as SQL databases, logs, CRLs: • `mnt`: Directory shared by the NFS server and remotely mounted by the Administration application and the End user application. This shared directory contains the subdirectories: `files` (files uploaded/downloaded by end users), `themes` (custom look and feel resource files), and `connector` (files deposited by an out-of-band mechanism to be processed by the OpenTrust MFT file connector in offline mode • `ca`: CA certificates. Contains data when X09 client certificate authentication is configured (Web server and Administration application) or audit logs signature is enabled (Administration application). • `crl`: CA CRLs (see above) • `db`: the PostgreSQL database files • `log`: (see “Log Files” on page 12) OpenTrust MFT application logs (End user application and Administration application), Tomcat logs, Apache logs, PostgreSQL logs, internal audit log files in XML format (audit log files are only written on disk when the database is not available) • `run`: runtime PID files • `secure`: contains private keys and other private data, namely the Web servers private key (Web server and Administration application) and the private key signing audit logs if logs signature is enabled in configuration (Administration application) • `tmp/clamav`: directory where the ClamAV anti-virus temporarily uncompresses archives whose content is to be scanned • `backup`: directory containing the application backup data (see “Backup and Recovery” on page 19)

Directory	Description and Sub-directories
`www`	Web interface (OpenTrust MFT and PHP scripts), the sub-directory `i18n/` contains translation files

1.1.3. Support Scripts

This section describes the scripts that can be found in the `/opt/opentrust/mft/sbin` directory.

Availability of these scripts depend on how the OpenTrust MFT server application has been installed.

The detailed usage of a script is displayed when the script is launched with the `--help` parameter.

Support Script	Description
`CRL-get.pl`	Downloads the CRL from a CA (Web server and Administration application).
`logs-manager`	Manage the audit logs sending queue and log module database. Used by cron to send audit logs to the log module and may be used to archive and purge audit logs.
`nfs-management`	Manage the NFS configuration for server and client (mount point `/opt/opentrust/mft/var/mnt`).
`mft-config`	Perform low-level configuration of the OpenTrust MFT setup.
`mft-diag`	Prints version information about OpenTrust MFT, connects to the OpenTrust MFT database and to the audit logs database, dumps the local OpenTrust MFT configuration, restore OpenTrust MFT resources such as the administration application, the mail templates, and the default domain.
`mft-system`	Issue commands related to the internal workings of OpenTrust MFT , such as reloading modules and checking server status.
`mft-tag`	Manage labels through the front-end connector and the admin connector.
`mft-user`	Manage user specifications through the front-end connector and the admin connector.
`opentrust-mft-backup.pl`	Backs up the OpenTrust MFT server application (see “Backup and Recovery” on page 19).
`trust-management`	Adds a CA certificate or a CRL into the OpenTrust MFT trusted certificate store. Useful when configuring X509 client certificate authentication or when signing audit logs.
`opentrust-update-i18n`	Build internationalization files containing labels displayed within the application.

1.2. Services

This section lists the services to monitor per machine:

Machine(s)	Service	Port	Description
Web server	/usr/sbin/httpd.worker	80	Apache HTTP in worker mode (multi-thread mode)
Web server	/usr/sbin/httpd.worker	443	Apache HTTPS in worker mode (multi-thread mode)
Administration application	/usr/sbin/httpd	80	Apache HTTP (traditional multi-process mode)
Administration application	/usr/sbin/httpd	443	Apache HTTPS (traditional multi-process mode)
End user application	/usr/sbin/clamd	3310	ClamAV Daemon
End user application, Administration application	/usr/java/latest/bin/java	8980	Java Tomcat HTTP listener
End user application, Administration application	/usr/java/latest/bin/java	8900	Java Tomcat AJP listener

Machine(s)	Service	Port	Description
End user application, Administration application	portmap	111	Portmap (NFS client)
SQL	postgres	5432	PostgreSQL DB Server
NFS	nfsd	2049	NFS main server daemon
NFS	rpc.mountd	9049 (arbitrarily fixed value)	NFS mount server daemon
NFS	rpc.statd	662 (arbitrarily fixed value)	NFS lock recovery server daemon

The SSH service is available by default on every server (port 22).

The following table lists the local infrastructure services used by all machines of the OpenTrust MFT application:

Service	Port	Description
crond	N/A	cron tasks to be run by cron
mft-restarter	N/A	Service for restarting other services through a Web interface (not used yet)
/usr/libexec/postfix/master	25	Postfix mail server

1.3. Log Files

Important: Log files contain technical information. Log files should not be confused with audit logs, which record events related to users and administrators activity. Technical log files are written in simple files, whereas audit logs are recorded in the SQL logs database and are accessed through the Administration Console GUI.

While running, OpenTrust MFT generates technical log records for both *application* events (Administration application, End user application) and *service* events (Apache server, PostgreSQL server). Administrators can consult these files locally or via the SSH protocol. Log files are located in the `/opt/opentrust/mft/var/log` directory, except for the optional SMTP Connector component, whose log entries are directly written in Postfix's log file `/var/log/maillog`.

Service logs are rotated when their size exceeds 10 MB; 5 rotated log files are retained, moreover rotated files are backed up in the `archives` directory (see below).

Application logs are rotated when their size exceeds 50 MB; 20 rotated log files are retained.

By default, the application logs level is “INFO”; to run the application in “DEBUG” mode, edit the configuration file `/opt/opentrust/mft/etc/mft.conf` and in the `[tomcat]` section set `log_level=DEBUG`.

To run the optional SMTP Connector component in “DEBUG” mode, edit the configuration file `/opt/opentrust/mft/etc/mft.conf` and in the `[SMTPConnector]` section set `debug=1` (it is set to 0 by default).

Directory	Content
`archives`	Archived system log records.
`xml`	Temporary directory for system audit log records, which are later inserted in the audit log database. Data is only written here when the audit log database is not available.
`admin`	Administration application logs (namely produced by the Administration application graphical interface or by the Admin SOAP Connector).
`admin/xml`	Temporary directory for application administration audit log records, which are later inserted in the audit log database. Data is only written here when the audit log database is not available.
`user`	End user application logs.
`user/xml`	Temporary directory for End user application audit log records, which are later inserted in the audit log database. Data is only written only when the audit log database is not available.

File	Component	Content
`admin/appli.log`	Administration application	Logs related to the Administration application, including error messages.

File	Component	Content
`admin/errors.log`	Administration application	Java stacktraces relative to errors occurring during the application administration.
`admin/tech.log`	Administration application	Same content than `appli.log` plus additional technical log messages.
`user/appli.log`	End user application	Logs relative to the End user application, including error messages.
`user/errors.log`	End user application	Java stacktraces relative to errors occurring during the end user activity.
`user/tech.log`	End user application	Same content than `appli.log` plus additional technical log messages.
`http_access_log`	Web server	Record of each access request for HTTPS pages. Each line contains the source IP address, date, HTTP method, location, protocol, version, status code, size of response, referer, and user-agent (in respective order), as illustrated in the sample line below (the *Referer* and *User Agent* have been replaced by a "-"): `172.18.16.66 - - [21/Apr/2008:16:03:37 +0200] "GET /zephyr/connect HTTP/1.1" 200 6140 "-" "-"` For more information, go to `http://httpd.apache.org/docs/2.0/logs.html` and navigate to Log Files | Access Log | Combined Log Format.
`http_error_log`	Web server	Apache error logs for HTTPS requests. Usually contains application connection errors.
`http_plain_access.log`	Web server	Record of each access request for HTTP pages.
`http_plain_error.log`	Web server	Apache error logs for HTTP requests. Usually contains application connection errors.
`pg-mft.log`	SQL	Trace records of the OpenTrust MFT PostgreSQL database.
`php.log`	Administration application	PHP trace records.
`/var/log/maillog`	SMTP Connector	Trace activity of the optional SMTP Connector component.

1.4. Databases

The OpenTrust MFT server application has an internal PostgreSQL database server in which application configuration (domains, policies), application data (users, projects, file information), and audit logs are stored.

Important: Do not modify these databases manually.

The following table describes the contents of each database:

Database	Description
mft	Application database: stores configuration information, applications data, and access rights.
logs	OpenTrust MFT audit logs

1.5. NFS mounts

This section is not relevant in single-host installations.

The Administration application and End user application(s) need to mount a directory shared by the NFS server using the NFS protocol: `/opt/opentrust/mft/var/mnt/`. The mount point on the Administration application and End user application(s) is also `/opt/opentrust/mft/var/mnt/`.

Note:

The disk partition hosting the NFS server shared directory may be located on a NAS/SAN server accessed remotely (via iSCSI, for example). However, clients such as the Administration application and End user application(s) cannot access directly the NAS/SAN server via iSCSI; they must go through the NFS server.

1.6. Crontabs

In addition to the system crontab containing administrative tasks, the OpenTrust MFT application has its own crontab.

The crontab is defined in the same way the system crontabs are and is installed as `/opt/opentrust/mft/etc/cron.d/mft`. When the OpenTrust MFT services are started, a symlink in `/etc/cron.d` is created and points to the OpenTrust MFT server application crontab.

The OpenTrust MFT server application crontab is automatically recreated when the OpenTrust MFT service is started and automatically deleted when the OpenTrust MFT service is stopped.

Note: The OpenTrust MFT server application crontab is automatically generated. Do not edit the OpenTrust MFT server application crontab's configuration file. Changing the crontab's configuration file risks losing custom added actions. The system crontab may be edited without risk to the OpenTrust MFT server application.

2 System Administration

System administration topics for the OpenTrust MFT server application include:

2.1. Start the MFT Services

The OpenTrust MFT services can be accessed through the provided initialization script installed in `/opt/opentrust/mft/etc/init.d/mft` and linked to `/etc/init.d/mft` for convenience. Or an administrator can use the `service` command if it is provided by the operating system.

Use the syntax below to invoke the script as `root`:

```
# /etc/init.d/mft task [SERVICE]
```

or

```
# service mft task [SERVICE]
```

Where *`task`* stands for `start`, `stop`, `restart`, or `status`, as described in “Get the Services State” on page 15 and *`SERVICE`* (optional) is the service name. If no *`SERVICE`* is provided, then all services are assumed.

The available services are:

- `httpd`
- `tomcat`
- `mft-db`
- `etc`
- `mft-restarter`
- `clamd`
- `mft-smtp-connector` (if the optional SMTP Connector component has been installed)

In multi-machines installation mode only the relevant services are available on a machine.

To display more information when a service task is executed, add the `verbose` keyword at the end of the command:

```
# service mft task [SERVICE] verbose
```

2.1.1. Get the Services State

A service state may be queried using the command with the `status` operation:

```
# /etc/init.d/mft status
```

or

```
# service mft status
```

If at least one of the OpenTrust MFT components is not running due to an error or is not in “started” status, an error will be reported.

To display the state of all installed services, add `verbose` to the command. If only one service has to be checked, then add the service name after `status`:

```
# service mft status tomcat
```

2.1.2. Start, Stop, Restart Services

The OpenTrust MFT services will start and stop automatically at system boot or shutdown, but they can also be manually restarted, stopped, and started. The following tasks can be used with the service script:

- *start*: start the services
- *stop*: stop the services
- *restart*: issue a stop then a start unconditionally

To globally restart the OpenTrust MFT application with all of its services:

```
# service mft restart
```

or

```
# /etc/init.d/mft restart
```

`verbose` may be added at the end of the command line to get more detailed information for the current service.

If the command should only be targeted to one of the OpenTrust MFT services, then add the service name after the task. For example, to only restart the internal database, enter:

```
# service mft restart mft-db
```

or

```
# /etc/init.d/mft restart mft-db
```

2.2. Perform Common Maintenance Tasks

Common maintenance tasks for the OpenTrust MFT server application include:

- "Configure Postfix" on page 16
- "Check File Integrity" on page 16

2.2.1. Configure Postfix

2.2.1.1. Email Aliases

To modify default email redirections, edit the `/etc/aliases` file. This file contains a list of local accounts and a list of email addresses to which all email for the named account should be redirected.

For example:

```
root: admin@example.com, admin2@example.com
mft:  admin-mft@example.com, admin2-mft@example.com
```

To apply the modifications, run the **`/usr/bin/newaliases`** command after editing the `/etc/aliases` file.

At installation time, two aliases have been created, one for the `root` user and one for the `mft` user.

2.2.2. Check File Integrity

The integrity of OpenTrust MFT server application files can be verified using the RPM Package Manager. The **`rpm`** utility allows an administrator to compare information about the installed files in a package with information taken from the package metadata stored in the rpm database. The information compared is by default the size, MD5 sum, permissions, type, owner and group of each file.

Execute the following command to verify all OpenTrust packages and dependencies:

```
$ rpm -qa 'opentrust*' | xargs rpm --verify
```

Refer to the **rpm(1)** man page for a description of the command options and output.

2.2.3. Tune the Memory Size of the Application Server

The application server must have a minimum of 4 GB of RAM to run OpenTrust MFT in a production environment. To configure the memory size of the application server you must edit the `/opt/opentrust/mft/etc/mft.conf` file and set the *`tomcat_java_perf_opts`* property as follows:

```
[network]
...
tomcat_java_perf_opts=-server -Djava.awt.headless=true -Xms4096M -Xmx4096M -XX:PermSize=512M -
XX:MaxPermSize=512M -XX:+DisableExplicitGC
...
```

If the memory size of the application server is underestimated, any connection attempt to OpenTrust MFT is subject to be timed out or may result with an "Error 500" message.

2.3. Setting Log Levels

The log level determines the type of information that is logged. OpenTrust MFT provides the following log levels:

1. *`DEBUG`*: information that is helpful to diagnose a product issue.
2. *`INFO`*: generally useful information.
3. *`WARN`*: anything that can potentially cause application oddities.
4. *`ERROR`*: runtime errors.

If you want to set log levels, you must configure the *`log_level`* property in the `/opt/opentrust/mft/etc/mft.conf` file. For example, if you want to troubleshoot your application, you must set the *`log_level`* property as follows:

```
[tomcat] # Startup log level of application logs. Correct values are DEBUG, INFO, WARN, ERROR
log_level=DEBUG
```

This page intentionally left blank.

3 Backup and Recovery

OpenTrust recommends excluding servers hosting OpenTrust products from standard corporate backup policies and procedures and instead recommends the backup procedures provided in this documentation. A backup strategy must be defined and rigorously implemented. The backup archive consistency and usability should be verified by running recovery procedures in a test environment on a regular basis.

3.1. Back Up data

3.1.1. The Back Up Program

A dedicated program is provided for backing up the OpenTrust MFT data. It will ease the backup process and only back up necessary data. Please note that uploaded files are not in the program scope. Moreover, system-dependent files, outside of the `/opt/opentrust/` directory, **are not backed up**.

All backup commands must be executed as the `root` user.

The backup program path is `/opt/opentrust/mft/sbin/opentrust-mft-backup.pl`.

The default output backup directory is `/opt/opentrust/mft/var/backup/`. To use an alternate directory, use the `--output` command line parameter to specify the alternate directory.

Note: The back up program execution is not automatically scheduled in a crontab command upon OpenTrust MFT application installation (as it is the case for CRLs fetching for example); it is the customer duty to schedule the back up program execution.

Tip: All command line options and parameters may be displayed with the `--help` option:

```
# /opt/opentrust/mft/sbin/opentrust-mft-backup.pl --help
```

In multi-machines mode, the available backup options depend on the machine type. For instance, the `--tar data` option is only available on the NFS server, and the `--db` option is only available on the SQL server.

3.1.2. Back Up the Server Data

Backing up the server data and the internal database should be done on a daily basis. For example, the following command:

```
# /opt/opentrust/mft/sbin/opentrust-mft-backup.pl --db --tar data --timestamp --compress
```

will output two separate files in the default output backup directory, one with a full dump of the internal database, such as `db-20091124140359.sql.gz`, and another file with the server data, such as `data-20091124140359.tar.gz`.

Table 3.1. Recommended Daily Backup Commands for Each Server Hosting OpenTrust MFT Components

Server	Command
Single-host	# opentrust-mft-backup.pl --db --tar data,config,logs --timestamp --compress
Web server, Administration application, End user application hosts	# opentrust-mft-backup.pl --tar config,logs --timestamp --compress
SQL host	# opentrust-mft-backup.pl --db --tar config,logs --timestamp --compress
NFS host	# opentrust-mft-backup.pl --tar data --timestamp --compress

Tip: It is possible to back up the server data and the internal database without stopping the OpenTrust MFT service and its components. The **`mft-db` component must be running and the `httpd` service should be stopped**.

Application logs archives may be generated separately using the `--tar logs` command line parameter, to facilitate the use of a long-term archiving infrastructure:

```
# /opt/opentrust/mft/sbin/opentrust-mft-backup.pl --tar logs --timestamp --compress
```

Tip: Add those command lines to a cron job to trigger the backups automatically, according to the defined backup strategy and frequency.

Note: There is no support for remote archive copying. Use the method most appropriate for the infrastructure. Checking the archive file integrity when copying to a remote/removable media is a good best practice.

Note: Do not let backup files fill the servers local file system; periodically remove and archive older backup files.

3.1.3. Back Up Secure Data

Secure data are stored only on the Web servers and Administration application servers.

Using the `--tar secure` command line option will **add secure data to the backup** such as:

- Web server private key (Web servers)
- Logs signing certificate private key (Administration application server)

Important: Secure data values do not change often, so there is no need to back them up daily, but secure data must be backed up as soon as the OpenTrust MFT application is fully configured.

Do not let backup tarballs containing sensitive data linger on a machine's local file system; always put the backup file in a safe and delete it from the local file system after checking its integrity.

Note: Due to the multi-host architecture constraints, the OpenTrust MFT keystore and truststore, shared by all Tomcat servers (and thus located on the NFS server), are not backed up with other secure data; they are backed up when using the `--tar data` command.

Use the following command line to create a backup file of the secure data:

```
# /opt/opentrust/mft/sbin/opentrust-mft-backup.pl --tar secure --timestamp --compress
```

An archive will be created, with a filename such as: `data-20091124140050.tar.gz`

3.2. Restore Backed Up Data

To restore an OpenTrust MFT server application, the following elements must be available and the integrity must have been verified:

- Operating system backup or installation media
- Application RPM files as released by OpenTrust for **the same version** of the OpenTrust MFT from which the backup has been generated
- OpenTrust MFT application backup files (data, secure, database)

The following restore procedure assumes that the operating system is operational. Follow the installation procedure as described in the *OpenTrust MFT Server Installation and Upgrade Guide*, including network and services setup until the OpenTrust MFT startup dialog. It is not necessary to perform the configuration steps.

1. Log in as the `root` user to proceed to the restore.
2. Copy the backup files to the server then change directory to this location.
3. Stop the OpenTrust MFT service:

   ```
   # service mft stop
   ```

   or

```
# /etc/init.d/mft stop
```

4. Restore the server data backed up in “Back Up the Server Data” on page 19:

```
# tar xvfz data-[timestamp].tar.gz -C /
```

5. Uncompress the OpenTrust MFT internal database backed up in “Back Up the Server Data” on page 19:

```
# gunzip db-[timestamp].sql.gz
```

6. Start the OpenTrust MFT database:

```
# service mft start mft-db
```

or

```
# /etc/init.d/mft start mft-db
```

7. Restore the database content:

```
# psql -h /opt/opentrust/mft/var/run/db -p 5432 -U mftdb -f db-[timestamp].sql postgres
```

8. Start the OpenTrust MFT service:

```
# service mft start
```

or

```
# /etc/init.d/mft start
```

9. Download the last available CRL:

```
# /opt/opentrust/mft/sbin/CRL-get.pl
```

If the service or one of its components does not restart after repeated attempts, contact an assigned OpenTrust technical representative. For more information on the service script, see “Start the MFT Services” on page 15.

This page intentionally left blank.