Operation and Maintenance NTF

Operation Directions

Copyright

Disclaimer

The information in this document is subject to change without notice.

Ericsson makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Ericsson shall not be liable for errors contained herein nor for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Trademark List

Solaris	is a trademark of Sun Microsystems, Inc.
Apache Software Foundation	This product includes software developed by the Apache Software Foundation (http://www.apache.org/).
The Hypersonic SQL Group & The HSQL Development Group	This product includes software developed by The Hypersonic SQL Group and The HSQL Development Group. See Copyright Notices in the Appendix.

Contents

1 Introduction

1.1 Scope

This document provides information on operation and maintenance specific for the Notification component (NTF).

1.2 Audience

The document is primarily written for O&M personnel.

Part of the document may also be useful for personnel responsible for software installation.

1.2.1 Prerequisite Knowledge

The readers of this document should have a good understanding of Messaging-over-IP and good knowledge and experience of:

- IP based networks
- Protocols LDAP, SNMP, SMTP, IMAP, HTTP, SMPP or CIMD2, RADIUS.
- UNIX

Note: Operation and maintenance of the system is only to be performed by personnel authorized by having attended the corresponding courses.

1.3 Related Documents

The following documents also contain information related to the component, or are referred to from this document:

- Messaging-over-IP Overview
- Glossary
- Installation Guide NTF
- Solaris Documentation (http://docs.sun.com) *System Administration Guide, Solaris 10*

1.4 Document Conventions

1.4.1 Notational Conventions

This document uses the following notational conventions:

Bold font style is used for emphasis, to indicate keywords and buttons.

Italic font style is used for references, window/page/menu titles and specific terms.

`Bold Monospace font` is used to describe user input.

`Monospace font` is used for code, paths and on-screen computer output.

Square brackets “[]” are used to enclose parameters that are optional.

Curly brackets “{}” are used to enclose parameter values given as examples.

Less than and greater than characters “<>” are used to enclose variable names.

Backslash “\” at the end of a line means that this line is continued onto the next line.

The dollar sign “$” is the UNIX Korn (and Bourne) shell prompt.

The percent sign “%” is the UNIX C shell prompt.

The number sign “#” is the superuser prompt.

The number sign “#” is also used for comments.

A star “*” matches zero or more characters.

A question mark “?” matches one character.

A tilde “~” means the home directory of the current user.

`<BASEDIR>` is the installation directory for NTF, for example `/apps/ntf`.

`<NTF_HOME>` is the instance directory for NTF, for example `/apps/ntf` or `/global/ms2/ntf`. In a non-HA system, `BASEDIR` and `NTF_HOME` are the same.

2 About the Component

NTF is a component within Messaging-over-IP that provides notification services. For an overview of the NTF component, see *Messaging-over-IP Overview*.

2.1 Third Party Products

The NTF component includes software developed by the Apache Software Foundation (http://www.apache.org/).

The NTF component includes software developed by the Hypersonic SQL Group and The HSQL Development Group

2.2 SNMP Environment

The component includes the component specific NTF MIB, which has the following OID:

`enterprises.ericsson.ipms.ipmsProducts.ntf`(1.3.6.1.4.1.193.41.3.9).

3 Operation

Operation includes tasks as described in:

- Section 3.1 Start NTF on page 5
- Section 3.2 Stop NTF on page 5
- Section 3.3 Restart NTF on page 6
- Section 3.4 Check Status of NTF on page 6
- Section 3.5 Autostart Enable/Disable on page 7
- Section 5.5.1 Lock/Unlock Component on page 26
- Section 3.8 Auto Recover on page 9

3.1 Start NTF

The `/etc/init.d/rc.ntf` script with argument `start` can be used to manually start the NTF component in a controlled way.

1. Become superuser (root).

   ```
   % su -
   ```

2. Run the script with **`start`** option.

   ```
   # ./etc/init.d/rc.ntf start
   ```

   The result should be like the following output:

   ```
   NTF : <host name> :  result ok
   NTF : <host name> :  info ntfWatchdog started on PID:
   <pid>
   #
   ```

3.2 Stop NTF

The `/etc/init.d/rc.ntf` script with argument `stop` can be used to stop the NTF component.

1. Become superuser (root).

   ```
   % su -
   ```

2. Run the script with **`stop`** option.

```
# /etc/init.d/rc.ntf stop
```

The result should be like the following output:

```
NTF : <host name> :  result ok
NTF : <host name> :  info ntfWatchdog process stopped.
```

3.3 Restart NTF

The `/etc/init.d/rc.ntf` script with argument `restart` can be used to restart NTF component.

1. Become superuser (root).

   ```
   % su -
   ```

2. Run the script with **`restart`** option.

   ```
   # /etc/init.d/rc.ntf restart
   ```

   The result should be like the following output:

   ```
   NTF : <host name> :  info ntfWatchdog process stopped.
   NTF : <host name> :  result ok
   NTF : <host name> :  info ntfWatchdog started on PID:
   <pid>
   #
   ```

3.4 Check Status of NTF

The `rc.ntf` script with argument `status` can be used to check the status of the NTF component.

1. Become superuser (root).

   ```
   % su -
   ```

2. Run the script with **`status`** option.

   ```
   # /etc/init.d/rc.ntf status
   ```

   The result can be one of the following,

   After a *start*:

   ```
   NTF : <host name> :  status ntf started on PID: <pid>
   (autostart)
   NTF : <host name> :  info ntfagent started on PID:
   <pid> (autostart)
   NTF : <host name> :  info ntfWatchdog started on PID:
   <pid> (autostart)
   #
   ```

where "(autostart)" indicates that autostart is enabled, while "(not autostart)" indicates that autostart is disabled.

3.5 Autostart Enable/Disable

The autostart function makes NTF start automatically after boot.

The `rc.ntf` script with argument `enableautostart` or `disableautostart` can be used to enable or disable autostart of the NTF component. Autostart is enabled during installation

3.5.1 Enable Autostart

1. Become superuser (root).

   ```
   # su -
   ```

2. Run the **`rc.ntf`** script with **`enableautostart`** as argument.

   ```
   # /etc/init.d/rc.ntf enableautostart
   ```

   The result can be one of the following:

   ```
   NTF:<host name>:result ok
   NTF:<host name>:info enableautostart set
   ```

   or (if already enabled):

   ```
   NTF:<host name>:result error enableautostart already
   set
   NTF:<host name>:info S99ntf already exists
   #
   ```

3.5.2 Disable Autostart

1. Become superuser (root).

   ```
   # su -
   ```

2. Run the `rc.ntf` script with **`disableautostart`** as argument.

   ```
   # /etc/init.d/rc.ntf disableautostart
   ```

   The result can be one of the following:

   ```
   NTF:<host name>:result ok
   NTF:<host name>:info disableautostart set
   ```

   or (if already disabled):

   ```
   NTF:<host name>:result error disableautostart already
   set
   ```

```
NTF:<host name>:info s99ntf already exists
#
```

3.6 MCR Registration/Unregistration

During installation, NTF registers itself in the MCR.

If the MCR registration must be changed, for example if the version number should be changed, change the `<NTF_HOME>/cfg/ntf_cmp.cfg` file, and run the `<BASEDIR>/bin/ntf_mcr_reg.sh` script with <NTF_HOME> as input. MCR username and password is read from <NTF_HOME>/cfg/notification.cfg.

3.6.1 MCR Registration

```
# <BASEDIR>/bin/ntf_mcr_reg.sh -d <NTF_HOME>

NTF registered succesful
#
```

3.6.2 MCR Unregistration

```
# <BASEDIR>/bin/ntf_mcr_unreg.sh -d <NTF_HOME>

NTF unregistered succesful
#
```

3.7 User Registrations

3.7.1 Mailboxes registration/unregistration

During installation, NTF registers a number of gnotification mailboxes in MUR. The Message Store then uses these mailboxes to store the notifications in. NTF then fetches the notifications from the mailboxes in Message Store.

If the number of mailboxes must be changed, or if the registrations failed during installation, the gnotification mailboxes can be registered/unregistered manually.

The number of mailboxes are defined by the **IMAPThreads** parameter in `<NTF_HOME>/cfg/notification.cfg`. If this parameter isn't defined, the default value will be used.

Username and password for MUR is read from `<NTF_HOME>/cfg/notification.cfg`, or default it the parameter is not changed.

NTF will also register a notification.off user in the MUR during installation. This user is used by other components to send notification.off messages to NTF.

If the NTF for example failed to register this user during installation, it can be done manually.

The script will only update MUR if there will be any changes. Both gnotifications and notification.off is handled in the same call.

So, if the gnotification mailboxes should be 3, just edit/add the **IMAPThreads** parameter and give it the value **3**. Then run the mur registration script.

3.7.1.1 User Registration

```
# ./<BASEDIR>/bin/ntf_mur_reg.sh -d <NTF_HOME>

Gnotification registration
**************************
gnotification3_0 registered.
gnotification3_1 registered.
gnotification3_2 registered.
GNotification registered successful

Notification.off registration
*****************************
<notification.off user> registered.
Notification.off registration successful


#
```

3.7.1.2 User Unregistration

```
# ./<BASEDIR>/bin/ntf_mur_unreg.sh -d <NTF_HOME>

Gnotification unregistration
**************************
gnotification3_0 removed.
gnotification3_1 removed.
gnotification3_2 removed.
GNotification unregistered successful

Notification.off unregistration
*****************************
<notification.off user> removed.
Notification.off unregistration successful
#
```

3.8 Auto Recover

NtfWatchdog is used to start up the NTF in case the NTF processes stop, due to an unexpected process termination or crash.

4 Preventive Maintenance

Preventive maintenance means regularly performed measures to prevent faults from occurring and to reduce the time needed to correct faults. Measures are described in:

- Section 4.1 Backup on page 11
- Section 4.2 View Static Data in MIB on page 13
- Section 4.3 View Dynamic Data in MIB on page 14
- Section 4.4 Manage Disk Space on page 17

4.1 Backup

It is important to back up the NTF component to be able to restore the component in the event of a severe failure such as power failure, hard disk problems, or if the component stops functioning.

The consequences of not backing up the component can for example be that it may not be possible to rollback to a previous version, and in the event of a severe failure with a following reinstallation of the software, the configurations may not match.

A backup should be taken:

- Prior to any installation, upgrade or rollback.
- Any time the configuration has been changed.

The following files are recommended to back up:

Table 1 Files to back up.

File/Directory	Contents	Time recommended	Consequence of loss
`<NTF_HOME>/cfg/notification.cfg`	See: Section 5.6.1 notification.cfg on page 27	Installation. Upgrade. Rollback.	Since each installation has a different configuration, loss of this file may result in incorrect operation, such as connecting to the wrong message store or reading from the wrong notification mailbox.

Table 1 Files to back up.

File/Directory	Contents	Time recommended	Consequence of loss
`<NTF_HOME>/cfg/systemnotification.cfg`	See: Section 5.9 Configuring System Notifications on page 51	Installation. Upgrade. Rollback.	System notifications will not work without this file.
`<NTF_HOME>/templates/*`	See: Section 5.7 Configuring Notification Templates for Different Languages on page 30	Installation. Upgrade. Rollback.	All NTF messages for to the user will be simple messages in English.
`<NTF_HOME>/logs/NotificationTrace.log`	See: Section 5.4 Log Files on page 23	After an error has occurred.	Information for tracing an error in NTF may be lost.
`<NTF_HOME>/logs/ntfagent.log`	See: Section 5.4 Log Files on page 23	After an error has occurred.	Important information for tracing an error in NTF may be lost.
All files in NTFs data directory <NTF_HOME>/data	These files contain journal files and databases with information about notifications that have not yet been delivered. They also contain persistent management information from the NTF MIB. When NTF is started, the state of NTFs pending notifications is reconstructed from this file.	Upgrade. Rollback	Old, not completed outdial notifications and reminders will be lost.[1] Slamdown information for slamdownlist will be lost.

[1] If this is acceptable, you do not need to restore these files.

4.1.1 Backup Script

Run the script `bckntfconfig.sh` to backup the NTF files in the table above. The files are added to a tar file named `NTF.<instancename>.backup.tar`, where `instancename` is the value of the parameter `MCR_INSTANCE_NAME` in `notification.cfg` (e.g. **`ntf1@host.domain`**). The script writes out the location of the created tar file to `stdout`.

In a **standard system**, the script is located in `/apps/moipbackup/bckntfconfig.sh`

In a **HA-system**, the script is located in `<NTF_HOME>/moipbackup/bckntfconfig.sh`. To back up all NTF instances on a cluster node, you must run the script once for each instance.

4.2 View Static Data in MIB

Static parameter values are not updated by the component during operation. The static parameters defined by the NTF MIB can be divided into groups as described in:

- Section 4.2.1 NTF on page 13
- Section 4.2.2 Consumed Service on page 13

For instructions on how to view static MIB parameters, see *MIB Management User's Guide*, or see the documentation supplied with your SNMP management tool.

4.2.1 NTF

The NTF MIB defines the following static parameters for the NTF:

Table 2 NTF MIB Static Parameters

Parameter	Type	Description
ntfName	DisplayString	Name of the NTF(as in MCR).
ntfVersion	DisplayString	Version of the NTF.
ntfInstallDate	TimeStamp	Date and time when the current version of the NTF was last installed or upgraded.

4.2.2 Consumed Service

For NTF, a consumed service is an output interface. Each type of output is one consumed service, SMS is one consumed service, MMS another etc. If all SMSCs are working, that consumed service is up. If some of the SMSCs are not working, the cosumed service is impaired, and if all SMSCs are down, the consumbed service is down.

The NTF MIB defines the following static parameters for the consumed service within the NTF:

Table 3 NTF MIB Consumed Service Static Paramers

Parameter	**Type**	**Description**
ntfConsumedServiceName	DisplayString	The name of the ntfConsumedService entry
ntfConsumedServiceIndex	INTEGER	The index for the service

4.2.3 Consumed Service Instance

NTF reports server instances that are not working (status is down). The instance is added if the status is changed from up. The instance is removed if the status is turned up again.

The NTF MIB defines the following static parameters for the consumed service instance within the NTF:

Table 4 NTF MIB Consumed Service Instance Static Parameters

Parameter	**Type**	**Description**
ntfConsumedServiceInstanceName	DisplayString	The name of the ntfConsumedService entry
ntfConsumedServiceInstanceIndex	INTEGER	The index for the instance
ntfConsumedServiceIndex	INTEGER	The index for the service
ntfConsumedServiceInstanceHostname	DisplayString	The host for this instance
ntfConsumedServiceInstancePort	INTEGER	The port for this instance
ntfConsumedServiceInstanceZone	DisplayString	The logical zone for this instance

4.3 View Dynamic Data in MIB

Dynamic parameter values may be updated by the component during operation. The dynamic parameters defined by the NTF MIB can be divided into groups as described in:

- Section 4.3.1 NTF on page 15
- Section 4.3.2 Consumed Service on page 16

For instructions on how to view dynamic MIB parameters, see *MIB Management User's Guide*, or see the documentation supplied with your SNMP management tool.

4.3.1 NTF

The NTF MIB defines the following dynamic parameters for the NTF:

Table 5 NTF MIB Dynamic Parameters

Parameter	Type	Description
ntfOperationalState	INTEGER 1=enabled, 2=disabled	Indicates if the NTF is in operation. The NTF may be disabled if mailboxPollerStatus is inactive or all notification consumed-services are down.
ntfCurrentUpTime	TimeStamp	Time (in hundreds of a second) the NTF has been running since it was last restarted or put into enabled state.
ntfAccumulatedUpTime	TimeStamp	Total time (in hundreds of a second) the NTF has been in enabled state since the installation date.
ntfNotifInQueue	INTEGER	The total number of external notifications that are queued to be handled.
ntfNotifForRetry	INTEGER	The total number of external notifications in queue that are stored for retry.
ntfInternalQueues	INTEGER	The total number of notifications that NTF has received and started processing, but not yet delivered to the subscriber.

Parameter	Type	Description
ntfMailboxPollerStatus	INTEGER 1=active, 2=inactive	This parameter indicates if NTF can read new messages.
ntfLogLevel	INTEGER 0=none, 1=errors, 2=verbose, 3=debug	This parameter controls the amount of logging in NTF.

4.3.2 Consumed Service

The NTF MIB defines the following dynamic parameters for the management and monitoring subsystem within the NTF :

Table 6 NTF MIB Consumed Service Dynamic Parameters

Parameter	Type	Description
ntfConsumedServiceStatus	INTEGER 1=up, 2=down, 3=impaired	Denotes whether the service is up, down or impaired. up indicates that all consumed services are accessible. down indicates that all consumed services are not accessible. impaired indicates that one or more, but not all consumed services are accessible.
ntfConsumedServiceTime	INTEGER	The number of seconds since the last change of service status.
ntfConsumedServiceNumSuccess	INTEGER	The total number of successfully notifications sent to service
ntfConsumedServiceNumFailures	INTEGER	The total number of notifications with failures sent to service

4.4 Manage Disk Space

There is no need to clean old files created by NTF.

The data in the data directory `<NTF_HOME>/data` does not grow over time. The log files automatically roll over to a .sav file when reaching the maximum log file size, which is 1 MB by default.

The amount of data stored is not large (a few hundred megabytes), but if there is little disk available, you can review the size of data files when a new system has been running fo a while, to see how much data is need for this systems particular traffic pattern.

4.5 Self Diagnostics

The Self Diagnostics simplifies some of the regular preventive health checks needed to keep the Messaging-over-IP system operational.

Self Diagnostics is described in `Operation and Maintenance Messaging-over-IP`

4.5.1 Self Diagnostics test cases for the NTF component

Below is a decription of the NTF specific test cases in Self Diagnostics:

(M) for mandatory or (C) for conditional.

ntf.env.ver.gen Registered version (C)

ntf.env.conf.cons.gen Checks that config parameter is correct (M)

ntf.env.data.datadir Checks that the datadirectory is writable (M)

ntf.env.data.free Checks that the datadirectory has enough free space (M)

ntf.env.data.slamdown_journal_size Checks the size of the slamdown_journal file (M)

ntf.env.data.database_files Checks for needed database files (M)

ntf.env.data.dbsize Checks the size of the database files (C)

ntf.info.extra Extra info from log files (C)

ntf.env.conf.dirdns.gen Checks DNS alias for directory access (C)

ntf.env.conf.static.sieve_filter Checks that the sieve filter is the same in MS and NTF (M)

ntf.fun.perf.op.gen Operational state (M)

ntf.fun.perf.adm.gen Administrative state (C)

ntf.fun.perf.loglev.gen Checks the log level (M)

ntf.fun.perf.logerr.NotificationTrace_log.gen Checks for error in logfile (C)

ntf.fun.perf.pr.all.gen Checks processinformation (C)

ntf.fun.perf.core.gen Checks for core files (C)

ntf.fun.perf.gnotsize Check the size of notification mailboxes (C)

ntf.fun.perf.gnotdiff Check that all notification mailboxes are polled, and that messages are deleted (M)

ntf.fun.perf.threads Collect thread status and check for java level deadlocks in the notification process (M)

ext.ntp.gen NTP synchronization (M)

ntf.ext.cs.status.multimediamessage.gen Status for consumed service MultimediaMessage (C)

ntf.ext.cs.status.wapgateway.gen Status for consumed service WapGateway (C)

ntf.ext.cs.status.pagernotification.gen Status for consumed service PagerNotification (C)

ntf.ext.cs.status.callmwinotification.gen Status for consumed service CallMwiNotification (C)

ntf.ext.cs.status.outdialnotification.gen Status for consumed service OutdialNotification (C)

ntf.ext.cs.status.shortmessage.gen Status for consumed service ShortMessage (C)

ntf.ext.cs.status.externalsubscriberinformation.gen Status for consumed service ExternalSubscriberInformation (C)

ntf.ext.cs.failure.multimediamessage.gen Unsuccessful requests for consumed service MultimediaMessage (C)

ntf.ext.cs.failure.wapgateway.gen Unsuccessful requests for consumed service WapGateway(C)

ntf.ext.cs.failure.pagernotification.gen Unsuccessful requests for consumed service PagerNotification (C)

ntf.ext.cs.failure.callmwinotification.gen Unsuccessful requests for consumed service CallMwiNotification (C)

ntf.ext.cs.failure.outdialnotification.gen Unsuccessful requests for consumed service OutdialNotification (C)

ntf.ext.cs.failure.shortmessage.gen Unsuccessful requests for consumed service ShortMessage (C)

ntf.ext.cs.failure.externalsubscriberinformation.gen Unsuccessful requests for consumed service ExternalSubscriberInformation (C)

4.5.2 Self Diagnostics data collection for the NTF component

This section describes the data collected by Self Diagnostics when started with the -c switch. The assembled data is stored in a compressed archive file, see `Operation and Maintenance Messaging-over-IP` for more details. The data-file is stored in /apps/selfdiag/ntf and named selfdiag_<TIMESTAMP>.tar.gz. For example

/apps/selfdiag/ntf/selfdiag_2006_11_01_164230.tar.gz.

5 Corrective Maintenance

Corrective maintenance means fault tracing and correction. Measures are described in:

- Section 5.1 Monitor Traps and Correct Faults on page 21
- Section 5.2 Restore Data on page 21
- Section 5.4 Log Files on page 23
- Section 5.5.2 Change Configuration Parameters in MIB on page 26
- Section 5.6 Change Configuration Parameters in File on page 27

5.1 Monitor Traps and Correct Faults

The NTF MIB defines SNMP traps that can be sent from the NTF component, which are described in:

- Section 5.1.1 ntfStarted on page 21
- Section 5.1.2 ntfStopped on page 21

5.1.1 ntfStarted

The `ntfStarted` trap (OID: 1.3.6.1.4.1.193.41.3.9.2.1) is issued when the NTF component is started. It reports:

a The operational state of the NTF (“ntfOperationalState”).

b The administrative state of the NTF (“ntfAdministrativeState”).

5.1.2 ntfStopped

The `ntfStopped` trap (OID: 1.3.6.1.4.1.193.41.3.9.2.2) is issued when the NTF component is stopped. It reports:

a The operational state of the NTF (“ntfOperationalState”).

b The administrative state of the NTF (“ntfAdministrativeState”).

5.2 Restore Data

For information on how to rollback the NTF installation to a previous version, see *Installation Guide NTF*, section *Rollback*.

5.2.1 Restore Instance Data

To restore NTF instance data from a backup:

1. Stop the NTF you want to restore configuration data for.

   See Section 3.2 Stop NTF on page 5.

2. Move to the <NTF_HOME> directory.

   ```
   # cd <NTF_HOME>
   ```

3. Locate the tape archive (*tar-file*) that was created when NTF was backed up. It may have been compressed by the backup tools, and must then be uncompressed:

   If the file is called *.Z, uncompress it with the uncompress command.

   If the file is called *.gz, uncompress it with the gunzip command.

4. Copy the tape archive (*tar-file*) to <NTF_HOME>

   ```
   # cp <tar-file> <NTF_HOME>
   ```

5. Either make a full restore from the tape archive (*tar-file*) or make a partial restore by defining which files that should be restored.

   ```
   # tar xvf <tar-file>
   ```

   or

   ```
   # tar xvf <tar-file> <files_to_extract>
   ```

6. If the backup were taken on a different host, than it is restored on, the following parameters in `<NTF_HOME>/cfg/notification.cfg` must be manually configured.

Table 7 Parameters that must be manually configured in notification.cfg

Parameter	Description
MCR_INSTANCE_NAME	This value must be changed to apply to the current hostname. **Example: {ntf1@hostname.domainname}**
IMAPHost	Defines which Message Store this NTF should run towards. **Example: {ms.domainname}**

7. If you restore data files from a NTF where the file names includes an instance number, you must then manually rename the restored data files:

```
# mv outdial.<inst_nr>.journal outdial.journal
```

```
# mv outdial.<inst_nr>.journal.second outdial.journa
l.second
```

8. Restart the NTF to make configuration take effect.

   See Section 3.1 Start NTF on page 5.

5.3 Monitoring Communication with Snoop

The command snoop can be used to monitor communication between the NTF host and other hosts. It is probably most useful for monitoring the communication with the SMSC, with ESI, MVAS/MAS and with ESS. Since the MS is usually on the same host as NTF, the communication between NTF and MS can not be monitored this way.

A full description of the snoop command is found in the man page for snoop. Examples of usage that are useful with NTF are:

```
# snoop tcp port portno
```

Snoop displays, in a brief text format, all TCP communication to and from the port. If the port number of an SMSC is used for example, you will see information about TCP packets to and from the SMSC. If the port number is used for XMP communication, you will see traffic to and from ESIs and MVAS/MASes. If the port is used for event sending you will see traffic to and from the ESS.

```
snoop -o filename tcp port portno
```

Snoop logs TCP communication to and from the designated port in a binary file. This file can then be analyzed with some protocol analyzer. More information is available in the binary format and protocol analyzers usually understand the SMPP protocol used to communicate with SMSCs and HTTP which is used for XMP, so it is easier to analyze the communication that way. Snoop itself can also be used to view, in text format, the content of the binary file, with the command

```
# snoop -i filename
```

5.4 Log Files

The NTF log files reside in the directory `<NTF_HOME>/logs`. The log files can also be accessed from the MoIP log directory `/apps/logs/ntf`, through the symbolic link: `ntf.logdir`.

The NTF component maintains the following log files:

Table 8 NTF Log Files

File	Description
NotificationTrace.log	Contains error and debug messages for the notification process. This file can also be placed somewhere else, by defining the parameter LogFile in notification.cfg
NotificationProcess.log	Contains initialization and error messages for the notification process.
ntfagent.log	Contains log information for management and monitoring.
error.log	Contains information about some serious errors concerning the NTF database for outdial notifications and reminders.

5.4.1 Description of NotificationTrace.log

The format of each entry in the `NotificationTrace.log` file is:

```
<Date-Time> [<Log Level>] [<Thread Name>] <Message>
```

where:

`<Date-Time>` indicates when the system has encountered the event.

`<Log Level>` indicates the type of event, e.g. Error.

`<Thread Name>` indicates the name of the current Thread, e.g. Thread-10.

`<Message>` further describes the event.

Each entry is on a separate line. Here follows an example of an entry in the log file:

```
<date> [ Error ] [EventSender-Thread] Problem sending \
to MER Info: Preferred RadiusServer is: <host name> at \
port: 1813
```

The only log level that should be used during operation is Error. Error indicates that an error has occurred in the system.

5.4.2 Description of ntfagent.log

The format of each entry in the `ntfagent.log` file is:

`<Date-Time>|<pid>|[<Log Level>]|<source>:  <Message>`

where:

`<Date-Time>` indicates when the system has encountered the event.

<pid> indicates the process ID of the source

`<Log Level>` indicates the type of event, i.e. Error or Verbose.

<source> indicates which sub-part of NTF that generated the event.

`<Message>` further describes the event.

Each entry is on a separate line.

```
2000-03-14 11:51:31.194 + 0100|531|Information|AgentThread:
Sending start trap.
```

Example 1   ntfagent.log

5.4.3 Description of NotificationProcess.log

The `NotificationProcess.log` lists error messages that are directed to `standard error`. This file is normally empty, since errors are logged in `NotificationTrace.log`.

Errors are logged in `NotificationProcess.log` if they occur during the startup phase before proper logging has been established, and some low-level errors java may also occur there.

5.4.4 Description of error.log

The format of each entry in the `error.log` file is:

`<Date-Time> <Message>`

where:

`<Date-Time>` indicates when the system has encountered the event.

`<Message>` further describes the event.

Each entry is on a separate line.

5.5 Control NTF Through the MIB

5.5.1 Lock/Unlock Component

This following parameter in the NTF-MIB controls the behaviour of the started component.

Table 9 NTF Administrative States

Parameter	Description	value
ntfAdministrativeState	In the unlocked state, NTF processes notifications to the end users	1=unlocked
	In the locked state, NTF does not process notifications to the end users	2=locked
	In the shutdown state, NTF starts to finish all ongoing operations and is smoothly going to the locked state	3=shutdown

The ntfAdministrativeState MIB attribute value can be changed by using a SNMP management tool.

For instructions on how to change the value of a parameter in the MIB, see *MIB Management User's Guide*, or see the documentation supplied with your SNMP management tool.

5.5.2 Change Configuration Parameters in MIB

The following parameters in the NTF-MIB control the behaviour of the started component.

Table 10 NTF MIB Changeable Values

Parameter	**Description**	**value**
ntfLoadConfig	1=active 2=inactive	If this variable is set to active, NTF configuration file is reloaded. After the reload is done, the value is set to inactive
LogLevel	0=No logging 1=Error 2=Verbose 3=Debug	This variable has the same effect as changing the LogLevel parameter, but is effective immediately, without restarting NTF.

The MIB attribute value can be changed by using a SNMP management tool.

For instructions on how to change the value of a parameter in the MIB, see *MIB Management User's Guide*, or see the documentation supplied with your SNMP management tool.

5.6 Change Configuration Parameters in File

The NTF component uses the configuration files as described in:

- Section 5.6.1 notification.cfg on page 27
- Section 5.6.2 MCRAddon.xml on page 28
- Section 5.7 Configuring Notification Templates for Different Languages on page 30
- Section 5.9 Configuring System Notifications on page 51
- Section 5.6.3 XmpErrorCodes.cfg on page 29

5.6.1 notification.cfg

The `notification.cfg` configuration file can be used to configure most of the NTF functions.

The file will probably only be set once, at installation time. The file is read at start up. Changes to this file will not take effect until NTF is restarted or NTF is instructed to reload the file through the MIB management interface. Back up the file prior to an upgrade or new installation of the NTF, otherwise the configuration parameter values will be lost.

The `notification.cfg` file is by default located in the directory:

```
<NTF_HOME>/cfg
```

The format is: “ <Parameter>=<Value>”

The name of all parameters is case insensitive

The tables below show which configuration parameters in the `notification.cfg` file that can be changed. Some of these parameters are, by default, not present in the file. To change the value for such a parameter (from the default value), you just have to add a line in the file that defines a new value for the parameter.

There is a tool that can analyze and clean the NTF configuration file. It is described in Section 7.2.1 checkconfig on page 127.

5.6.2 MCRAddon.xml

The MCRAddon.xml file can be used to configure extra parameters for MCR components.

When NTF reads a component from MCR it also checks this file for extra parameters for that component. If the component doesn't exists in this file no additional parameters are read.

MCRAddon.xml must be located in <NTF_HOME>/cfg to be read. If it don't exists there, no components addons will be used, but there will be no errors.

The file is an xml document in the following format.

```
<register>
 <service name="ShortMessage">
  <component name="SMSCA">
   <username>USERNAME</username>
   <password>PASSWORD</password>
   <systemtype>SYSTEMTYPE</systemtype>
  </component>

  <component name="SMSCB">
   ...
  </component>
 </service>

 <service name="SERVICETYPE">
  ...
 </service>
</register>
```

5.6.3 XmpErrorCodes.cfg

NTF handles XMP results in two stages. The result first goes to a handler for the general cases which will pass the result on to a more intelligent handler if necessary. If the result shall be regarded as a success or failure for the consumed service is always determined by the general handler.

The operation of the general handler is controlled by the configuration in XmpErrorCodes.cfg, which tells NTF how to react on different XMP error codes. There are general settings for all XMP services, which can be overridden by individual settings for each XMP service.

The file can have one common section and one section for each service. All sections are optional. Each section starts with a heading consisting of the service name within square brackets, followed by one or more XMP code-action pairs. A line beginning with a hash mark is a comment.

The XMP codes can be any number allowed in the XMP protocol.

The actions can be

- remove. Retry this XMP service request on the next available XMP server. Stop using the current XMP server. NTF checks for unavailable servers at a regular interval and making the servers availible again. The interval time is defined in the parameter mcrExpiryTime. This count a failure for this consumed service.

- next. Retry this XMP service request on the next available XMP server. Continue to consider the current XMP server as available. This count a failure for this consumed service.

- report. Forward this result to the specialized result handler for the service. This count a failure for this consumed service.

- pass. Forward the result to the specialized result handler for the service. This count a success for this consumed service.

```
#XMP error handling for NTF
[ common ]
408=next
421=remove
500=report
501=report
502=next
521=remove

[ outdialnotification ]
513=remove

[ pagernotification ]
503=remove
513=remove

[ callmwinotification ]
513=remove
```

Example 2 Example of XmpErrorCodes.cfg

5.6.4 SMPPErrorCodes.cfg

Configuration for error actions for the SMPP protocol. Each error code from the SMPP PDU can have an error action such as "retry" or "failed". The SMPP error codes can be configured for actions in the file <NTF_HOME>/cfg/SMPPErrorCodes.cfg. Error codes not specified in the file will use the action configured in the default field.

The actions are:

- retry. The errorcode will be handled as a temporary error and the notification will be resent later. The mail in the gnotification mailbox will be marked for retry and is fetched the next time NTF reads mails marked for retry.
- failed. The errorcode will be handles as a permanent error and the notification will not be retried.

```
## SMPP error code actions
#  Add an SMPP error code with action if you want
#  another behaviour than the default action.
#
#  Format:
#  <error code>=<action>
#  where action can be equal to "retry" or "failed"

default=failed

# Throttling error
0x00000058=retry

# Message Queue Full
0x00000014=retry
```

Example 3   Example of SMPPErrorCodes.cfg

Note: The configuration parameters SmscErrorAction and SmppErrorCodesIgnored will override the configuration in file <NTF_HOME>/cfg/SMPPErrorCodes.cfg.

5.7 Configuring Notification Templates for Different Languages

When a notification is sent as an SMS, WAP Push or E-mail, the message text is generated from a template. The template defines a text, which can include information from these sources:

- Fixed texts.
- Mail content and headers (sender, date, subject etc.).
- The receivers inbox (number of new messages, whether the inbox is full etc).

The templates for notification texts are configured in one *phrase file* for each supported language, so the user gets notification texts in his preferred language. It is possible to specify templates per CoS as described in Section 5.7.1 CoS specific templates on page 31.

A phrase file contains the following information:

- Translations

  (See Section 5.7.3 Translations on page 32)

- Template strings

  (See Section 5.7.4 Template Strings on page 33)

- Subtemplate strings

  (See Section 5.7.7 Subtemplate Strings on page 43)

- Conversion Specification

  (See Section 5.7.8 Converting the Telephone Number on page 44)

All language files must be stored in the directory <NTF_HOME>/templates, and have a file name consisting of the language name with the file extension ".cphr", for example "en.cphr" for the English phrase file. NTF must be restarted for changes to take place, this includes if a file has been changed or added.

Note: The phrase files must be located in the specified directory and have the required file names. Otherwise, the NTF component will not find the files.

5.7.1 CoS specific templates

It is possible to specify templates per CoS that override the language templates. CoS templates are defined per language and CoS. If no CoS template is found, NTF will use the language template. CoS templates are defined in files named <language>-x-<cosname>.cphr where <language> is the language and <cosname> is COSname in MUR, cosname is case insensitive, for example "en-x-business.cphr".

The format of the CoS file is the same as the regular phrase file, but only the templates that are different from the language template need to be defined. Templates that are not CoS dependent are fetched from the language file.

5.7.2 Format of Phrase Files

The information in a phrase file is stored as a number of lines, where each line defines a name and a value. The format is:

```
<Name>={
<Value>
}
```

For example:

```
general={
```

```
"You have a new message"
}
```

Consider the following format rules when you create a phrase file:

- The name must **not** contain any spaces.
- The value is a combination of one or more constants or tags. Constants are surrounded with quotes. Tags **must not** be surrounded with quotes.
- Long values can be split over multiple lines using newline characters.
- Comment lines must start with a hash mark (#).
- Missing templates, subtemplates, and translations, are given the value of the “general” template.
- Missing languages are given the templates for the default language.
- Any Unicode character can be entered with its hexadecimal code prefixed by "\u", e.g. the newline character can be entered as \u000a. This is most useful for unprintable characters and characters not in ISO 8859-1. Note that the \u000a and other special characters are treated as tags and can not be embedded in a constant text. I.e. they need to be placed outside quotes.

5.7.3 Translations

The simplest lines in the phrase file just specify a translation for an english word into another language. Then the name in the line is an english word and the value is a corresponding word or sequence of words in another language. A translation in a phrase file for spanish could contain the following line.

```
voice=voz
```

Example 4 A translation for the spanish language.

The table below shows all english words that can be translated in the phrase file.

Table 11 Translations

Translations	Description
urgent	String used to indicate that the message status is `urgent`.
normal	String used to indicate that the message status is `normal`.
email	String used to indicate that the message is an e-mail.
voice	String used to indicate that the message is a voice message.

Translations	Description
fax	String used to indicate that the message is a fax message.
video	String used to indicate that the message is a video message.
unknownsender	If a callers telephone number is restricted or not found, this text will be shown instead of the number.

5.7.4 Template Strings

A template string is a string that defines the layout and contents of a complete notification message. The following template strings are available:

h SMS “header” notification text.

s SMS “subject” notification text.

c SMS “count” notification text.

f Flash SMS notification text.

e0 First of the ten available templates for Notification by Email. The templates e0 to e9 can be used. (see Section 5.8 Configuring Notification by E-mail on page 49)

slamdown Slamdown information text. See Section 5.13 Configuring Slamdown Information on page 63

ivrtosms Missed call information notification text.

faxprintfail Text in notification of faxprint failure.

WapPushText WAP push notification text.

general default template which is used when use of the other templates fails, e.g. when a template string includes a message count but the message count cannot be determined.

Note: The general template should not contain any tags.

mailquotaexceeded The template for warnings that the users mailbox is full (the name of this template is configurable, see Section 7.1.12.19 QuotaTemplate on page 116, but mailquotaexceeded is the default name).

mailquotahighlevelexceeded The template for warnings that the users mailbox is *almost* full.

smstype0text Normally empty, but if there are many users with old telephones that do not handle SMS type 0 correctly, an explanatory message can be set here.

mwiontext Normally empty, but if there are many users with old telephones that do not handle MWI on correctly, an explanatory message can be set here.

mwiofftext Normally empty, but if there are many users with old telephones that do not handle MWI off correctly, an explanatory message can be set here.

unreadmessagereminder The template used for reminder SMS. The mail based tags can be used if the users inbox contains exactly one message, otherwise only inbox based tags can be used just like in Example 5 on page 35 for updateafterretrieval.

updateafterretrieval The template to use if a update SMS is sent after a user has made retrieval. The mail based tags can be used if the users inbox contains exactly one message, otherwise only inbox based tags can be used. See Example 5 on page 35

updateafterterminalchange The text to use for a update SMS after terminal change. The mail based tags can be used if the users inbox contains exactly one message, otherwise only inbox based tags can be used just like in Example 5 on page 35 for updateafterretrieval.

smsinsteadofmwi A template to use if the user have mwi but the terminal does not support mwi.

<system notification name> Template for system notification defined in `systemnotification.cfg`. Any number of system notifications can be defined (see Section 5.9 Configuring System Notifications on page 51).

The template names (h, s and c) imply notifications with *header*, *subject* and *message count* information, but this is just a guideline, and any content can be configured for each of the notification types. Which SMS template that is used for a particular notification is controlled by notification filter settings. Notification filters can be defined through the *Provisioning GUI* or through the Web interface used for managing mailboxes.

```
updateafterretrieval = {
(*) "You still have "
(2-) TCOUNT
(2-) " new messages"
(1) "one new message from "
(1) FROM
(*) " in your mailbox"
}
```

Example 5 Updateafterretrieval

The format on how to write count templates can be found in Section 5.7.6 on page 39.

5.7.5 Tags

A template string contains constant text in the target language surrounded by quotes. It is also possible to insert *tags* anywhere outside the quotes. A tag is a special word that is replaced by a piece of information from the e-mail or the receivers inbox. Some examples of possible `c` (count) template strings are shown below.

```
c= {
"You have new messages."
}
c={
"You have " TCOUNT " new messages."
}
c={
"You have a new message from " FROM ". "
  "(" VCOUNT " voice messages, " FCOUNT " faxes, "
  ECOUNT " emails, " MCOUNT " new video messages)."
}
```

Example 6

In the example above, the tags `TCOUNT`, `VCOUNT` , `FCOUNT`, `ECOUNT` and `MCOUNT` will be replaced by the total number of new messages, the number of new voice messages, the number of new fax messages, the number of new e-mail messages and the number of video messages in the subscribers inbox. The tag `FROM` will be replaced by the From address in the message.

Any words outside quoted constant text are **tags** that are replaced with dynamic information. The table below shows which tags that can be used in template strings for notification texts.

Table 12 Tags for notification texts

Tag	Description	Based-on
TCOUNT	The total number of new messages in the subscriber's inbox.	Inbox
FCOUNT	The number of new fax messages in the subscriber's inbox.	Inbox

Tag	Description	Based-on
ECOUNT	The number of new e-mail messages in the subscriber's inbox.	Inbox
VCOUNT	The number of new voice messages in the subscriber's inbox.	Inbox
MCOUNT	The number of new video messages in the subscriber's inbox	Inbox
STATUS	This field displays the priority of the message. The data for this field is from the `X-Priority` mail field. If the priority of the message is equal to **1** or **2** then the message is classified as `urgent`. Otherwise the message is classified as `normal`. The string used to notify the status is selected from the translations named `urgent` and `normal` in the phrase file (see Section 5.7.3 Translations on page 32).	Mail
TYPE	This field denotes the type of message. The message can be a voice mail message, a video message, a fax message, or an e-mail message. The string inserted is selected from the translations named `voice, video, fax` and `email` in the phrase file (see Section 5.7.3 Translations on page 32).	Mail
TIME	This field contains the time when the message was deposited. The format is based on the subscriber's preference.	Mail
DATE	This field contains the date when the message was deposited. The date is by default presented in the subscribers preferred date format. The date tag can optionally include a format specification, which will then be used instead of the subscribers preferred format. The optional date format is described in Section 5.7.5.2 DATE= format tag on page 38.	Mail
SUBJECT	The subject of the e-mail message. This is in the `subject` header field of the e-mail message. If no subject is defined then `<no subject>` is substituted for this string.	Mail

Tag	Description	Based-on
FROM	The sender of the e-mail message. This is in the `from` header field of the e-mail message. If the sender is a non-subscriber, only the `<cli-number>@mvas.domain` will be displayed. If the number is restricted or not present, this tag will be replaces by the contents of the translation `unknownsender`	Mail
SIZE	The size of the message. If the message type (TYPE) is an e-mail message, the size is in bytes. If it is a voice or video message, then it is represented in seconds. If it is a fax message, it is represented in pages. The format of the size information can be customized with a subtemplate. (See Table 14 on page 43)	Mail
NUM_ATTACHMENTS	If the message is not a MIME message then the number of attachments is 0. [1] If the message is a MIME message and the content type of the message indicates it is a multi-part message then each part to the multi-part message is counted. Note that this multi-part message can contain recursive multi-parts. These recursive parts are not counted.	Mail
EMAIL_TEXT	The text portion of the e-mail.[2]	Mail
QUOTA_TEXT	This tag is a subtemplate that adds a warning about a full inbox, to the normal notification text. If the inbox is not full, this tag is replaced by an empty string. See QUTOA_TEXT in Table 14 on page 43	Inbox

Tag	Description	Based-on
PHONE	The telephone number of the subscriber is inserted into the message. This will in some telephones let the user call back to the messaging system with a single click. Using CONVERTED_PHONE, the telephone number can be modified before it is inserted into the text. The beginning and end of the telephone number can be replaced with other characters. The prefix, the suffix and their replacements is configurable (Section 5.7.8 Converting the Telephone Number on page 44.)	Inbox
CONVERTED_PHONE		Inbox
QUOTE	Inserts a double quote (“) into the message.	None
UNICODE=	Inserts a unicode character. I.e UNICODE=000a is equivalent to \u000a wich is a newline.	None

(1) NOTE. Do not use this parameter unnecessarily. Heavy load is put on NTF.

(2) NOTE. Do not use this parameter unnecessarily. Heavy load is put on NTF.

5.7.5.1 Special Date Format

The date tag can include a format specification and then has the form

```
DATE=<format>
```

where <format> is a format specification with the syntax described in Section 5.7.5.2 DATE= format tag on page 38

Example:will be replaced by the date like 2008-01-20, independent of the subscribers preferred date format.

```
DATE=yyyy-MM-dd
```

Note: One template can include several date tags with different special date formats, and different templates can also have different special date formats.

5.7.5.2 DATE= format tag

In slamdown information, the subscribers preferred date and time formats are ignored. To allow a compact time representation, it is separately configured, with the `DATE=` tag.

The tag is followed by a string composed of the characters yMdEahHm,.:;-_=+ with the following interpretation:

yy	year - two-digit format.
yyyy	year - fourdigit format.
M	month - one or two digits.
MM	month - two digits.
MMM	month - abbreviated name.
MMMM	month - full name
d	date - one or two digits
dd	date - two digits
EEE	weekday - abbreviated name.
EEEE	weekday - full name.
a	am or pm.
h	hour - one or two digits in 12-hour format.
hh	hour - two digits in 12-hour format.
H	hour - one or two digits in 24-hour format.
HH	hour - two digits in 24-hour format.
m	minute - one or two digits.
mm	minute - two digits.

, . : ; - = + can be used anywhere in the string without special meaning. A space is represented by _ character

The names of months and weekdays are shown in the users preferred language.

5.7.6 Adapting Template Strings to the Number of New Messages

Sometimes the quality of text messages can be improved by adapting the text to the number of messages. As an example, you probably want to avoid messages like “You have 0 new voice message(s) and 1 new fax message(s).” NTF allows this through *Count Specific Template Strings*. Count Specific Template Strings extends the syntax for Template Strings.

A part of a Template String that starts with a parenthesis on a new line is treated as a Count Specific Template String.

It can handle zero, one, two, three, four, five, six, seven, eight, nine and many. It can also handle a range of numbers such as numbers between two and four, or all numbers greater than or equal to five.

A Count Specific Template String shall consist of several lines with an optional mail count condition in front. The mail count condition can have 4 numbers or 1 number. If it is 4 number they are treated as voice, fax, email, video. Such as:

(1,0,0,0) " one new voice message "

or if it is 1 number it is treated as the total count of new mails (voice + fax + email + video) such as:

(2) "you have 2 new messages in"

The first line above has the condition (1,0,0,0) for one voice, zero fax, zero e-mail and zero video messages. The second line matches 2 new messages of any kind.

See Table 1 for valid values in condition.

Table 13 Valid values for condition

Condition	Description
d	Condition is true when the number of mail in INBOX of type voice, fax, e-mail or video is: - equal to d where d is an integer between 0 and 9.
d-	Condition is true when the number of mail in INBOX of type voice, fax, e-mail or video is: - equal to d - greater than d where d is an integer between 0 and 9.
-d	Condition is true when the number of mail in INBOX of type voice, fax, e-mail or video is: - equal to d - less than d where d is an integer between 0 and 9.

Condition	Description
d1-d2	Condition is true when the number of mail in INBOX of type voice, fax, e-mail or video is: - equal to d1 or d2 - less than d2 - greater than d1 where d1 and d2 are an integer between 0 and 9.
*	Condition is always true.

A generated text from a template is the concatenation of all lines where condition matches the actual number of messages of different types. Several text combinations can thus be generated with a minimum number of lines in the template.

There are two examples below that show how to build up a template. One c template is defined in English for voice and fax messages and one c template is defined in Croatian also for voice and fax messages.

```
1. c = {
2.   “You have ”
3. (0,0,*,*) “no new messages.
4. (1,*,*,*) “one”
5. (2-,*,*,*) VCOUNT
6. (1-,*,*,*) “ new voice message”
7. (2-,*,*,*) “s”
8. (1-,0,0,0) “.”
9. (1-,1-,*,*) “ and “
10. (*,1,*,*) “one”
11. (*,2-,*,*) FCOUNT
12. (*,1-,*,*) “ new fax”
13. (*,2-,*,*) “es”
14. (*,1-,0,0) “.”
15. }
```

Example 7 Configure a count (c) template in English for voice and fax messages

Note that the line numbers on the left side in the example shall not be included in the .cphr file.

A user with no new messages in the INBOX, condition (0,0,0,0), would give the phrase: “You have no new messages.”. Lines 2 and 3 in the c template have a condition that matches the INBOX.

A user with one new voice message and one new fax in the INBOX, condition (1,1,0,0), would give the phrase: ”You have one new voice message and one new fax.”. Lines 2, 4, 6, 9, 10, 12 and 14 in the c template have a condition that matches the INBOX.

A user with 5 new voice messages and 20 new faxes in the INBOX, condition (5,20,0,0), would give the phrase: ”You have 5 new voice messages and 20 new faxes.”. Lines 2, 5, 6, 7, 9, 11, 12, 13 and 14 in the c template have a condition that matches the INBOX.

A user with no voice messages and 3 new faxes in the INBOX, condition (0,3,0,0), would give the phrase: “You have 3 new faxes.”. Lines 2, 11, 12, 13 and 14 in the c template have a condition that matches the INBOX.

```
1. c = {
2.  ‟Imate ″
3. (1-,*,*,*) VCOUNT
4. (1,*,*,*) ‟ novu glasovnu poruku‟
5. (2-4,*,*,*) ‟ nove glasovne poruke‟
6. (5-,*,*,*) ‟ novih glasovnih poruka″
7. (1-,0,0,0) ″.″
8. (1-,1-,*,*) ‟ i ‟
9. (*,1-,*,*) FCOUNT
10. (*,1,*,*) ‟ novu fax poruku″
11. (*,2-4,*,*) ‟ nove fax poruke″
12. (*,5-,*,*) ‟ novih fax poruka″
13. (*,1-,0,0) ‟.″
14. }
```

Example 8 Configure a count (c) template in Croatian for voice and fax messages

Note that the line numbers on the left side in the example shall not be included in the .cphr file.

A user with one voice message in the INBOX, condition (1,0,0,0), would give the phrase: “Imate 1 novu glasovnu poruku.”. Lines 2, 3, 4 and 7 in the c template have a condition that matches the INBOX.

A user with 2 voice message in the INBOX, condition (2,0,0,0), would give the phrase: “Imate 2 nove glasovne poruke.”. Lines 2, 3, 5 and 7 in the c template have a condition that matches the INBOX.

A user with 4 voice message in the INBOX, condition (4,0,0,0), would give the phrase: “Imate 4 nove glasovne poruke.”. Lines 2, 3, 5 and 7 in the c template have a condition that matches the INBOX.

A user with 5 voice message in the INBOX, condition (5,0,0,0), would give the phrase: “Imate 5 novih glasovnih poruka.”. Lines 2, 3, 6 and 7 in the c template have a condition that matches the INBOX.

A user with one new voice message and one fax message in the INBOX, condition (1,1,0,0), would give the phrase: “Imate 1 novu glasovnu poruku i 1 novu fax poruku.”. Lines 2, 3, 4, 8, 10 and 13 in the c template have a condition that matches the INBOX.

```
1. unreadmessagereminder = {
2. (*) "This is a reminder that you still have "
3. (2-) TCOUNT
4. (2-) " new messages"
5. (1) "one new message from "
6. (1) FROM
7. (*) " in your mailbox"
8. }
```

Example 9 Configure reminder SMS template for English language

Note that the line numbers on the left side in the example shall not be included in the .cphr file.

A user with one unseen message from “123456” in the INBOX matches lines 2, 5, 6, 7 and will get a notification with the text “This is a reminder that you still have one new message from 123456 in your mailbox”.

A user with one unseen voice message, two unseen fax messages and one unseen video message in the INBOX matches the lines 2, 3, 4, 7 and will get a notification with the text “This is a reminder that you still have 4 new messages in your mailbox”

5.7.7 Subtemplate Strings

Subtemplate strings are strings that can be defined in the phrase file, and then be included in one or more template strings. The set of possible subtemplates is fixed.

The purpose of the subtemplate strings is to allow message type dependent size information.

A *subtemplate* is a string that can be used to format the size information for a particular message type. It can be any string which may include the special tag `SIZE`. If `SIZE` is present, it is replaced by the appropriate number when the notification text is generated. Note that this is the only tag that can be used in a subtemplate, and that it cannot be used in one of the main template strings. The table below shows the available message size subtemplates.

Table 14 Subtemplates

Subtemplate String	Description
FSIZE_TEXT	String for information about the size of a fax. Example: `FSIZE_TEXT= {”SIZE pages”}`

Subtemplate String	Description
ESIZE_TEXT	String for information about the size of an e-mail. Example: `ESIZE_TEXT= {”SIZE kilobytes”}`
VSIZE_TEXT	String for information about the size of a voice mail. Since the size information for voice messages is already formatted by the telephony system, this subtemplate should normally be omitted from the phrase files. Example: `VSIZE_TEXT= {”SIZE seconds”}`
MSIZE_TEXT	String for information about the size of an video message. Example: `MSIZE_TEXT={”length SIZE seconds”}`
fromnumberprefix	String that prefixes the FROM tag if the sender is a number. Example: `fromnumberprefix={”+”}`
AM	String for the text used as AM in time or date formats. Example: `AM={”am”}`
PM	String for the text used as PM in time or date formats. Example: `PM={”pm”}`
QUOTA_TEXT	String for information that user's mailbox is full. It can not contain tags, only constant text. Example: `QUOTA_TEXT= {” Your mailbox is full.”}`

5.7.8 Converting the Telephone Number

How the subscribers telephone number for the tag “CONVERTED_PHONE” shall be converted, is defined by the parameter `CONVERTED_PHONE` in the phrase file. This parameter specifies one or more conversions.

Note: This tag concerns the phone number of the **subscriber**, not the caller.

5.7.8.1 Prefix

The conversion string kan specify how to convert the beginning of a telephone number. The format for this is:

`prefix>replacement`

The prefix is separated from the replacement by a greater-than character, which you can see as an arrow from the old prefix to the new prefix.

```
CONVERTED_PHONE={"46>0"}
in the conversion specification will change 46701234567 into 0701234567
```

Example 10 Changing a swedish international number to a national number

5.7.8.2 Suffix

The conversion string kan specify how to convert the end of a telephone number. The format for this is:

`,suffix>replacement`

The conversion starts with a comma character to differentiate it from a prefix conversion. The suffix is separated from the replacement by a greater-than character.

```
CONVERTED_PHONE={",67>89"}
in the conversion specification will change 46701234567 into 46701234589
```

Example 11 Changing the end of a telephone number

5.7.8.3 Prefix and Suffix

One conversion can combine both a prefix conversion and a suffix conversion. This is applied to all telephone numbers that match both the prefix and the suffix. The format for the combined conversion is the prefix conversion followed by the suffix conversion:

`prefix>replacement,suffix>replacement`

```
CONVERTED_PHONE={"0>46,7>99"}
in the conversion specification will change 0701234567 into 467012345699,
but will not change 0701234568 or 1701234567,
since they do not match the prefix and suffix.
```

Example 12 Changing both ends of a telephone number

5.7.8.4 Multiple conversions

You can specify several different conversions in the same CONVERTED_PHO NE parameter. This is done by concatenating several conversions, separated by a slash character, i.e. the format is:

`p1>r1,s1>r2/p2>r3,s2>r4/...` where p1 and p2 specify prefixes, s1 and s2 are suffixes and r1, r2, r3 and r4 are replacements. The conversions are searched from the beginning and the first conversions that has a prefix and suffix that matches the telephone number, is used. The remaining conversions are skipped, i.e. at most one conversion is applied to each telephone number.

```
CONVERTED_PHONE={"46>0/47>0/358>0"}
in the conversion specification will change the country codes
for Sweden, Norway and Finland to the digit 0.

CONVERTED_PHONE={",1>11/,3>31/,5>51/,7>71/,9>91"}
in the conversion specification will append the digit 1 to all odd
telephone numbers.

CONVERTED_PHONE= {
"46>46,1>11/46>46,3>31/46>46,5>51/46>46,7>71/46>46,9>91"
}
in the conversion specification will append the digit
1 to all odd swedish telephone numbers.

CONVERTED_PHONE={ " 46>46555/358>358555"}
in the conversion specification will insert the digits 555 after the
country code in swedish and finnish international telephone numbers.
```

Example 13 Multiple conversions

5.7.8.5 Empty prefix or suffix

You can have an empty prefix or suffix, which will replace “nothing” with the replacement string, i.e. add something to the beginning or end of the telephone number.

```
CONVERTED_PHONE={ ">+" } in the conversion specification will add a
plus character to the beginning of all telephone numbers.

CONVERTED_PHONE={"358>358,>99/47>47,>98"} in the conversion
specification will append the digits 99 to the end of
all finnish telephone numbers and 98 to the end of all
norwegian telephone numbers.
```

Example 14 Adding to the telephone number

5.7.8.6 Empty replacement

You can have an empty replacement, which will replace the prefix or suffix with “nothing”, i.e. remove something from the beginning or end of the telephone number.

```
CONVERTED_PHONE={"46>/47>/358>"} in the conversion specification
will remove the country code from swedish, norwegian and finish
international telephone numbers.

CONVERTED_PHONE={ ",0>/,2>/,4>/,6>/,8>" } in the conversion
specification will delete the last digit from the
telephone number if it is even.
```

Example 15 Removing from the telephone number

5.7.8.7 Summary

The rules above can be summarized in these points:

- The greater-than character is used to separate the prefix and its replacement and to separate the suffix and its replacement.

- The comma character is used to separate the prefix rule from the suffix rule.
- The prefix, suffix and replacement can not contain a comma character or a greater-than character or a slash character.
- The prefix may be empty, to specify that the replacement shall be added to the beginning of the telephone number.
- The suffix may be empty, to specify that the replacement shall be added to the end of the telephone number.
- The replacement may be empty, to specify that the prefix or suffix shall be removed from the telephone number.
- If there is no need for a suffix rule, it can be omitted.
- If there is no need for a prefix rule, it can be omitted, but you must keep the comma before the suffix rule.
- If there is both a prefix and a suffix, the conversion will be applied only to telephone numbers that have both the prefix and suffix. The rule `+46>0,7>9` will not change the telehone numbers “+46701234568” or “46701234567”, but will convert “+46701234567” to “0701234569”

5.7.9 Example of Phrase File

This section describes how a phrase file can be configured to generate a certain message text in notifications.

Consider the following example, showing the currently available information to use for a notification.

```
Subject: Meeting next week
From: john@ericsson.se
Date: 2002-01-30
Time: 12.00

Shall we meet next week?
```

Example 16 Email contents for a notification

```
Number of new voice messages: 2
Number of new faxes:          0
Number of new emails:         4
Number of new video messages: 0
Mailbox full?                   no
```

Example 17 Inbox information for the notified user

```
.
.
.
s = {
FROM " sent you a message about " QUOTE SUBJECT QUOTE
". Your mailbox has " TCOUNT " new messages "
"(" VCOUNT " voice messages and " ECOUNT " emails)."
}
.
.
.
```

Example 18   Templates in the phrase file

This will produce the following notification text:

```
john@ericsson.se sent you a message about "Meeting
next week". Your mailbox has 6 new messages (2 voice
mails and 4 emails).
```

Example 19   Notification text

In producing this result, NTF goes through the following steps:

- NTF looks at the notification filters, and determines that the user has selected SMS notification with the “s” content.
- NTF looks at the users preferences and determines that the users preferred language is “en” (english).
- NTF finds the template string shown above.
- NTF replaces FROM with the sender address in the email (“john@ericsson.se”)
- NTF replaces SUBJECT with the email subject (“Meeting next week”)
- NTF replaces TCOUNT, VCOUNT and ECOUNT with the numbers (6, 2 and 4).

5.7.10 Command Line Test Tool

Test your phrase file configuration before it’s loaded in the NTF application. A text is printed to stdout, generated from the phrase files when you run the test tool with arguments. If anything went wrong in the creation of phrase files such as syntax errors, check the output from the NTF log file.

To activate the new phrase file configuration, start or restart NTF.

To test your phrase file configuration:

1. Change directory to <BASEDIR>/bin

```
# cd <BASEDIR>/bin
```

2. Start the command with arguments

#./generatetext –d <NTF_HOME> –c <content> –l <language> –v <voice> –f <fax> –e <e-mail> –m <video>

Example:

```
#./generatetext -d /global/ms1/ntf -c c -l en -v 1 -f 3
-e 0 -m 0
```

```
#Generated text:  You have one new voice message and 3 new
faxes.
```

Table 15 Test Tool Argument List

Arguments	Valid values
<content>	Notification content for example c for count, s for subject h for header and e0 for first Notification by Email template.
<language>	User preferred language. Use the language code for a language ISO 639, e.g. en for English.
<voice>	Number of voice mail in INBOX. Positive integer.
<fax>	Number of fax mail in INBOX. Positive integer.
<e-mail>	Number of voice e-mail in INBOX. Positive integer.
<video>	Number of video mail in INBOX. Positive integer.

5.8 Configuring Notification by E-mail

The Notification by E-mail functionality is configured very much like the SMS. If enabled, the quota warnings, slamdown information and fax print failed messages will also be sent using e-mail according to delivery profile. Notification by E-mail also uses templates for configuration (*.cphr). Template Strings **e0** to **e9** are reserved for e-mail notifications. In addition to these, the SMS Template Strings **s**, **c** and **h** can be used for email body content if content should be the same for SMS and E-mail notifications.

For all Template Strings it is recommended to add a subject header for the E-mail notification. A set of other e-mail headers can be associated with the used template strings, these are described below.

5.8.1 Notification by E-mail Header Configuration

The following headers can be set for each Template in a notification e-mail. It is possible to have, for example, one subject for e-mail template **e0** and another for e-mail template **e1**. All e-mail header templates are associated with its parent e-mail template (e.g. “e0” or “s”). For e0 to e9 the last digit is noted as <n> in the table below. For single character templates (c, s, h) <n> below is the character. The part of the e-mail header template noted as <CustomHeaderName> is any user defined header (usually starting with X-). Any number of emailnotification-header-x<n><CustomHeaderName> templates can be specified for each parent e-mail template.

Table 16 Optional E-mail Header Template

E-mail Header Template	**Valid values**
emailnotification-header-subject-<n>	Any subject, it may even be a count specific template. Example: emailnotification-header-subject-0 = { “E-mail Notification” } gives the email header for template e0: SUBJECT : “E-mail Notification”
emailnotification-header-from-<n>	A from address.
emailnotification-header-from-envelope-<n>	A valid envelope from address. Note: this is empty as default to avoid returned E-mails if subscriber is not found. Some SMTP servers does not accept this to be empty.
emailnotification-header-cc-<n>	A valid CC address.
emailnotification-header-bcc-<n>	A valid BCC address.
emailnotification-header-reply-to-<n>	A valid reply-to address.
emailnotification-header-x-<n><CustomHeaderName>	Any user defined header. Example: emailnotification-header-x-0X-myHeader = { “Off” } gives the email header for template e0: X-myHeader : Off

5.8.2 Example Configuration for Notification by E-mail

The following is an example part of a phrase file used to configure template “s” for notification by E-mail. Template “s” is used for a CoS where some users want SMS notifications, some E-mail and some both E-mail and SMS notifications. If content of the e-mail notifications should be the same as for the SMS notifications there is no use configuring the e0 template for this, instead the specific Notification by E-mail headers for “s” are configured.

```
s = {
"You have a new " SIZE " message with "
 NUM_ATTACHMENTS " attachments, regarding "
 QUOTE SUBJECT QUOTE "."
}

emailnotification-header-reply-to-s = {
 "please.reply.to@this.address.com"
}

emailnotification-header-subject-s = {
 "You have a new message with the subject "
 QUOTE SUBJECT QUOTE "."
}

emailnotification-header-from-s = {
 "from@some.address.com"
}
```

This will generate the following example e-mail:

```
SUBJECT:    You have a new message with the subject "testemail".
REPLY-TO:   please.reply.to@this.address.com
FROM:       from@some.address.com
You have a new size 1 kilobytes message
with 0 attachments, regarding "testemail".
```

5.9 Configuring System Notifications

The system notification functionality is general and flexible and can have many uses. It makes NTF convert some emails into SMS-messages, bypassing normal notification functionality.

Note: It is not really necessary that the mail arrives in the users mailbox. The important thing is that the user is in the ‘‘To:’’ list in the mail and that the mail arrives in one of NTFs gnotification mailboxes.

It is possible to configure which emails that shall be converted, and which SMS message shall be sent. The original purpose is to let the MoIP system notify a user about an important event, e.g. that the users mailbox is full. The system sends a normal email to the user, but NTF looks at the mail headers and recognizes that it is a system message and sends a special notification instead of the normal “You have new mail message” type of notification.

- System notifications are always sent with SMS.
- System notifications ignore filter settings.
- System notifications are sent even if the user has disabled notifications.
- System notifications use their own special SMS templates, one for each system notification.

Each system notification has a unique name. This is also the name of the SMS template used for that system message. Configuring system notifications is done in two steps:

- You tell NTF how to recognize a system message.
- You configure the SMS template for the system message.

5.9.1 systemnotification.cfg

The file `systemnotification.cfg` tells NTF which system notifications there are, and how to recognize an e-mail as a system notification.

To disable system notifications completely, you remove systemnotification.cfg (or rename it to e.g. systemnotification.off).

The rules for matching a system notification against an email are:

- A system notification is identified by one or more header name-header value pairs.
- A pair is considered to “match” an email if the value in the pair is a substring of the corresponding mail header.

  E.g. the pair “Subject=hello” matches an email with the header “Subject: You must say hello to John”.
- If all pairs configured for a system notification match the email, the email will generate a system notification, otherwise it will be a normal notification.
- Case is ignored both in the header name and the value.

A systemnotification.cfg can look like this:

```
#System notification configuration

#This matches the mail sent to the user when
#his mailbox is full
[ mailquotaexceeded ]
From=Mail Administrator
Subject=WARNING: Quota exceeded

#This matches the mail sent when a caller sends a
#cut-through-paging
[ cutthroughpaging ]
From=sink
Ipms-Notification-Type=sendshortmessage
```

Example 20 systemnotification.cfg

In this example, you can see that:

- The file consists of sections, one for each system notification.
- Each section starts with the name of the system notification in brackets (“[mailquotaexceeded]”).
- Each section has some configuration lines with a header name and a value (“From=Mail_Administrator”).
- You can have blank lines.
- Lines starting with # are comments (“#his mailbox is full.”).

This file can be configured for each system to the needs of the customer, but keep a few things in mind:

- Processing a system notification takes a little time for each arriving mail, so having hundreds of system notifications is not a good idea.

WARNING: Make sure that the header-value combinations are unique enough, so that not ordinary mails are misinterpreted as system notifications. “Subject=WARNING” is not a good alternative in mailquotaexceeded in the example above.

5.9.2 Configuring SMS Templates for System Notifications

The name of a system notification, which is defined within brackets in `systemnotification.cfg`, is also the name of a template string. A template string for each system notification should be added to the phrase files for all languages (en.cphr, sv.cphr etc.) to give the users the notifications in their preferred language.

Configuration of template strings is described in Section 5.7 Configuring Notification Templates for Different Languages on page 30. System notifications can use all the tags and subtemplate strings available in normal template strings.

See the following sections for examples of configuration of system notifications for various services.

5.9.3 Examples

5.9.3.1 Mail quota exceeded

The following two files exemplifies how NTF can be configured to warn users with full mailboxes.

```
[mailquoatexceeded]
From=Mail_Administrator
Subject=WARNING: Quota exceeded
```

Example 21 Part of systemnotification.cfg

```
mailquotaexceeded={
"Your mailbox is full. Please delete old mails."
}
```

Example 22 Part of phrase file en.cphr (language en)

If `systemnotification.cfg` and `en.cphr` contain the values shown in Example 21 on page 54 and Example 22 on page 54, the system could send an e-mail to the user with the specified From and Subject, and the user would receive an SMS message saying that the mailbox is full, and asking him to delete old mails, and if he has unread messages, the SMS will tell how many.

5.9.3.2 CLI notifications

When listening to messages in the inbox, the subscriber can request that the callers number is sent as an SMS.

The following two files exemplify how NTF can be configured to send CLI messages

```
[smscli]
From=sink
Ipms-Notification-Type=mvas.subscriber.caller_cli
```

Example 23 Part of systemnotification.cfg

```
smscli={"The number you requested: +" SUBJECT "."}
```

Example 24 Part of phrase file en.cphr (language en)

Note: The example shows how a plus character can be entered into the template, to indicate that the number is an international number. If the system uses national numbers, there should be no plus character.

5.9.3.3 Cut Through Paging notifications

The following two files exemplify how NTF can be configured to send Cut Through Paging notifications.

```
[cutthroughpaging]
From=sink
Ipms-Notification-Type=sendshortmessage
```

Example 25 Part of systemnotification.cfg

```
cutthroughpaging={
"+" EMAIL_TEXT " called you at " TIME ", "
DATE ". Please call back."
}
```

Example 26 Part of phrase files en.cphr (language en)

Note: The callers telephone number in a cut-through-paging request is sent as the body of an email, thus “EMAIL_TEXT” in the example.

Note: The example shows how a plus character can be entered into the template, to indicate that the number is an international number. If the system uses national numbers, there should be no plus character.

5.10 Configuring Missed Call Information Notification

The missed call information service is used to send SMS notification to a user that is not a voice mail subscriber. This service is also known as ivrtosms.

```
ivrtosms= {
"+" EMAIL_TEXT " called you at " TIME ", "
DATE ". Please call back."
}
```

Example 27 Part of phrase files en.cphr (language en)

The config parameter sourceAddress_IvrToSms can be configured to include the callers phone number to make the SMS appear like it is being sent from the caller phone number, see Section 7.1.12.41 SourceAddress_* on page 122 for parameter description.

Note: The callers telephone number in a ivr-to-sms request is sent as the body of an email, thus “EMAIL_TEXT” in the example.

Note: The example shows how a plus character can be entered into the template, to indicate that the number is an international number. If the system uses national numbers, there should be no plus character.

Since the receiver is not a subscriber, NTF can not find out what SMSC to use when sending the SMS. Instead, NTF distributes ivrtosms notifications over the SMSCs listed in the parameter `AllowedSmsc`. If this parameter is missing or empty, NTF uses all SMSCs it can find in the Messaging Component Register.

5.11 Configuring Outdial Notification

This section describes how to make the outdial notification feature work, and how to customize it for each customer.

5.11.1 Pre-requisites

To use outdial notification, a number of things outside NTF must be set up correctly:

- The user must have the service outdial notification in the COS.
- The users notification filter (or the notification filter in the COS if the user has none) must be set up to do outdial notification.
- The notification type outdial must not be disabled for the user.
- There must be at least one running MVAS or MAS, registered in MCR to provide the service OutdialNotification.

If NTF shall check if the phone is on before doing outdial, the following is required:

- The SMSC must support the use of SMS type 0.
- NTF must have an account in the SMSC that allows bind as a transceiver. If there are more NTFs using the same SMSC, each NTF needs its own account to ensure that delivery receipts are returned to the NTF that sent the SMS type 0.
- The users phone must support SMS type 0.

5.11.2 Enabling the Function

To turn outdial notification on, you must set the following parameters in `<NTF_HOME>/cfg/notification.cfg`:

DoOutdial Set this parameter to yes, to enable the outdial function.

If NTF shall check if the phone is on before doing outdial, the following parameter must be set in `notification.cfg`:

SmppBindType Set this parameter to **`transceiver`**, to allow NTF to receive SMS messages.

SmeSourceAddress, SmeSourceTon, SmeSourceNpi Set these parameters to correspond to NTFs SMS-sender address, so the SMSC knows where to return the receipt when the phone is turned on. These values are provided by the operator.

SetReplyPath This parameter must be set to match the SMSCs way of routing the receipt to NTF. This value is provided by the operator.

5.11.3 Customizing the Function

To tailor the function to each customers preferences, you primarily edit the outdial sequence as describer in Section 5.11.5 Configuring the Outdial Sequence on page 57. You can also set the following parameters in `notification.cfg`:

DataDirectory This is the directory where NTF writes files needed to recreate pending outdial notification if NTF is restarted.

5.11.4 Other parameters

The following parameters in `<NTF_HOME>/cfg/notification.cfg` affect the outdial notification function, but rarely need changing:

- XmpValidity
- XmpTimeout
- MaxXmpConnections

5.11.5 Configuring the Outdial Sequence

The outdial sequence is determined by a state machine which is completely configurable. Thus, almost any conceivable outdial sequence can be achieved by configuration. Each class of service can have its own state machine defined by the attribute emOutdialSequence in MUR. NTF looks for configuration files in `<NTF_HOME>/cfg` that are in the format of `outdial-<name>.cfg` where `<name>` is the name in emOutdialSequence. If no name exists in MUR or there is no matching configuration file, `outdial-default.cfg` is used.

Each outdial starts with the state machine in an initial state, doing some initial actions. When a call attempt is made the result code from the call, together with the current state that the outdial sequence is in, is used to determine the next operation sequence to be used and the state to go to after the sequence has been run.

The reason for using a state machine is that an ongoing call can have different behavior depending both on the outcome of the call attempt and how many tries that have already been done.

An operation sequence can have zero or more operations. An operation sequence that does not end with a call attempt should lead to the final state of the state machine. The result code from the call attempt will be used to select the next operation sequence and state.

5.11.5.1 Result Codes

The purpose of the outdial sequence is to call the subscriber, and the operation of the state machine is controlled entirely by the result of the call attempts.

Status codes from 900 and up, are internal NTF codes, while the other codes are status codes related to the outdial notification service in the XMP protocol. 6xx codes are used in some CA-versions of MVAS/MAS, for all other MVAS/MAS versions 4xx codes are used. The 6xx codes can be configured in MVAS/MAS and will replace the corresponding 4xx codes, however some 4xx codes like 408,421,450 will continue to exist. The following status codes are available:

200 The call attempt was successful and someone answered the call.

202 The call attempt was successful and someone answered the call.

401 The call attempt failed because the phone number was blocked.

402 The call attempt failed because the phone was busy in a call.

404 The call attempt failed because no-one answered the call.

405 The call attempt failed because the user is not reachable.

408 The call attempt failed because the request timed out.

421 The call attempt failed because the XMP server does not currently provide the service.

450 The call attempt failed temporarily.

500 The call attempt failed because the outdial request is unknown by the XMP server.

501 The call attempt failed because the parameters in the request were rejected by the XMP server.

502 The call attempt failed because resource limits in the XMP server were exceeded.

512 The call attempt failed because the phone number is not valid.

513 The call attempt failed for an unspecified reason.

514 The call attempt failed because the mailbox used for billing does not exist.

603 The call attempt failed because the user was busy in call.

610 The call attempt failed because no-one answered the call.

613 The call attempt failed because the user was not reachable.

614 The call attempt failed because the user suppressed the call.

620 The call attempt failed because there was congestion in the network.

621 The call attempt failed because of other calling failure.

900 Start of Outdial handling

910 Outdial notification is disabled by user for users current location (often because user is roaming). This occurs if the user has changed location since outdial was started.

915 It is not possible to determine the users current location.

920 The user has unconditional call forwarding set.

925 It is not possible to determine if user had unconditional call forwarding set.

930 Communication with ESI interrupted or other error.

940 Return after SMS or SMS0 (will probably not be used).

default What to do if an code that is not handled is returned.

Note: The meaning of result codes in the range 600-699 is not fixed. They are produced in the telephony interface by a configurable mapping from status codes in the telephony network. Thus it is possible to customize the outdial functions reaction on telephony status codes in great detail.

The values above is the default representation.

5.11.5.2 Operations

The result of a call attempt determines a sequence of operations that should be done. The available operations are:

WAITON Wait for the phone to be on (done by sending a SMS type 0). If the phone is not mobile, this action is ignored.

WAIT <t> Wait for **<t>** seconds, then go to the next operation

SMS <templatename> Send an SMS using the template **<templatename>**.

CALL Try to call.

The last operation in each sequence should be CALL, unless the outdial sequence ends with the next state.

5.11.5.3 States

A state is represented by an integer number, where the initial state by convention has number 0. The state is changed after the operation sequence is completed

5.11.5.4 Outdial Configuration File

The configuration file for the outdial sequence consists of a number of lines, each of the form **`param=value`**. There can also be comment lines, starting with **`'#'.`** Most of the configuration file describes the states and transitions but some parts of it define some basic operation for outdial. It is possible to define default transitions that apply for any state that does not have its own transition for an event explicitly set. This is useful for error codes that should be the same in any state but may be used for all reply codes. This is an example of an outdial configuration file:

Note: The file is not complete, and does not handle all codes. It just illustrates the type of lines that can occur in the configuration file.

```
# Example state machine
# Parameters not shown in diagram but needed
# Maximum time for call to stay in machine is 48 hours,
# if an outdial takes longer a sms with template
# outdial_replace will be sent
# The initial state is state 0 and total number of
# states is three (state 0 , state 1 and state 2).
maxwaithours = 48
initialstate = 0
numberofstates = 3
longtimesms = outdial_replace

# In any state that does not specifiy code 402,
# wait 120 seconds, do a call, then go to state 1
# This covers both the arc from state1 back to state1
# and the arc from state2 to state1 in the diagram
# It also defines an arc from state 0 to state 1 that is
# not in the diagram, but that arc will never be travelled.
default.402 = 1/wait 120; call

# In any state, if a call succeeds go to the final state
# and do nothing further
default.200 = END/


#
# Action to do when starting
state.0.900 = 1/waiton; wait 20; call

#
# Most of state 1 is given by default, but we must
# define what to do when no answer
state.1.404 = 2/wait 900; call

# If an unknown code comes continue with another call attempt.
# Known errors should be taken care of by setting default.<CODE>.
# The other errors that might come comes from making calls
# through MVAS/MAS.
state.1.default = 2/wait 900; call

# Most of state 1 is default, when no answer
# send sms and go to end state
state.2.404 = END/sms outdial_replace
```

5.11.5.5 Properties

There are some general properties that can be set in the file:

maxwaithours This is the maximum time outdial is to be attempted. When that time has passed the outdial will be stopped, regardless of the current state. An sms is sent if `longtimesms` is defined.

initialstate This is the state to start the outdial sequence in. The initial action is determined from this state. This should normally have the value 0. The states are numbered; 0, 1, 2, 3, n.

numberofstates This is the total number of states used. Note that if the last state number is n, then number of states must be n+1.

longtimesms This is the SMS template to use when sending SMS to notify about a call attempt that has taken longer than `maxwaithours`. The template is also used when a call gives a return code that is not defined for the current state.

5.11.5.6 Default Handling

Since many reply codes can be handled in the same way, regardless of the state, it is possible to define default handling of a result code. Similarly, there can be a default handling in a state, independent of the result code.

The properties defining the default handling are named **`default.<code>`** and **`<state>.default`** where <code> and <state> correspond to some code or state number.

The default handling means that there can be conflicting rules for a state and code. The rules are seleced in the following order:

- Correct state and correct code.

- Default state and correct code.
- Correct state and default code.
- If neither the state nor code match, an SMS is sent with the template defined by the `longtimesms` parameter.

The value specifies the next state and a sequence of operations in the format described in Section 5.11.5.8 Syntax for State and Actions on page 62 below.

Example:Meaning that the outdial sequence will by default terminate in all states when the call succeeds.

```
default.200=END/
```

5.11.5.7 Individual State Handling

Each combination of state and result code that needs an action or next state different from the default, needs a parameter in the configuration file. The parameter is named **`state.<stateno>.<code>`**. **`Stateno`** is the number of the state (0 up to numberofstates - 1) and **`code`** is the result code.

The value specifies the next state and a sequence of operations in the format described in Section 5.11.5.8 Syntax for State and Actions on page 62 below.

Example: Meaning that if a call attempt in state 3 gives the result code 404, another call attempt should be done after 15 minutes and the next state is 4.

```
state.3.404 = 4/wait 900; call
```

5.11.5.8 Syntax for State and Actions

The syntax for the property values is the same for default handling and for individual states. The format is **`<next state>/[<operation>[;<operation...]]`**, i.e. the value consists of two parts; the next state and the action. The parts are separated by a forward slash '/'. The next state is either a state number or the string “END”. “END” defines that the machine should go to its final state. The action is a sequence of zero or more operations. If there are more than one operation the operations must be separated with semicolon “;”. If the operation has a parameter the parameter is separated from the operation name with a blank “”.

5.12 Configuring MWI Notification for IP networks

Message waiting indicator (MWI) notification can be sent to both mobile and IP-based telephones. MWI for mobile phones are sent as SMS and the configuration can be found in Section 7.1.6 SMS Configuration Parameters on page 101. This section will handle MWI for IP-based telephones.

NTF sends a MWINotification XMP-request to MAS that sends a SIP NOTIFY request to the end user. The XMP config parameters in Section 7.1.9 XMP Configuration Parameters on page 109 are therefore used in the call towards MAS.

Except from the XMP configuration MWI for IP networks has all configuration in the file `<NTFHOME>/cfg/MWIforIP.cfg`. This file must exist or MWI for IP will not work. It is possible to configure the max time a notification is valid and how long to wait before making a new notification retry. It is possible to have different retry times for different retry attempts. For example the first 50 attempts could be done with 60 seconds interval and the rest with 300 seconds interval to decrease the load on the system and network.

NTF can count the users mail before every call attempt or NTF can count the mails at incoming deposit or when the subscriber logs out. This is controlled by the parameter alwayscheckcount in `MWIforIP.cfg`. The advantage by checking count at every call attempt is that the count will be correct even if the user reads mail from an external imapclient or via MEC but the disadvantage is increased IMAP traffic and load on the MS/NTF machine. The default value is to not check count at every call.

Example on MWIforIP.cfg

```
maxtimehours=168
waittime = 60
waittime.50-100 = 120
waittime.101-  = 300
alwayscheckcount = false
```

Wait 60 seconds for the first 50 attempts, 120 seconds for retry attempt 50 to 100 and wait 300 seconds for all other retry attempts. The notification is only valid for 168 hours.

Note: The answer from MAS can contain a retry-time that is the minimum time to wait before making a new attempt. This time will override the time configured.

5.13 Configuring Slamdown Information

There are two forms of slamdown information. When basic slamdown information is selected, an SMS and or E-mail with slamdown information is sent to the subscriber immediately, for each slamdown call. When slamdown information list is selected, NTF collects slamdown information until the subscribers phone is turned on. Then NTF sends one or more SMS containing a list of information about several slamdown calls. For Notification by E-mail, slamdown list is not supported. When slamdown list is configured for SMS, the E-mail notifications fall back to basic slamdown information. The configuration options differ depending on which slamdown information is selected.

5.13.1 Pre-requisites

To use slamdown information, a number of things outside NTF must be set up correctly:

- The slamdown function must be enabled in the users COS.
- The slamdown function must be enabled in the MVAS/MAS configuration.
- The user must have the service slamdown information (i.e. slamdown_notification in the COS).
- The user must not have disabled slamdown information.

If slamdown information list is used, the following must be set:

- NTF must have an account in the SMSC that allows bind as a transceiver. If there are more NTFs using the same SMSC, each NTF needs its own account or source address, to ensure that delivery receipts are returned to the NTF that sent the SMS type 0.

5.13.2 Enabling the Function

When all pre-requisites are fulfilled, slamdown information in some form will be sent, without further configuration

5.13.3 Customizing the Function

The slamdown information function can be customized in many ways. Most important is the choice between basic slamdown and slamdown information list.

You customize the function with the following parameters in NTFs configuration file (`<NTF_HOME>/cfg/notification.cfg`):

SlamdownList This parameter selects between basic slamdown and slamdown information list.

For the slamdown information list function, NTF needs some special communincation with the SMSC. The following parameters must be set:

SmppBindType Set this parameter to **`transceiver`**, to allow NTF to receive SMS messages.

SmeSourceAddress, SmeSourceTon, SmeSourceNpi Set these parameters to correspond to NTFs SMS-sender address, so the SMSC knows where to return the receipt when the phone is turned on. These values are provided by the operator.

SourceAddress_slamdown Source address that is set instead of SmeSourceAddress if source addresses should be differentiated for different types of SMS. This can contain the callers number.

SourceAddress_smsType0 Source address that is set instead of SmeSourceAddress if source addresses should be differentiated for different types of SMS.

SetReplyPath This parameter must be set to match the SMSCs way of routing the receipt to NTF. This value is provided by the operator.

The following settings control the content of the slamdown information list:

SmsStringLength Set this parameter to control the maximum length of a slamdown information message. If the slamdown information does not fit into one SMS, it will be sent in several SMS.

SlamdownMaxCallers Set this parameter to force NTF to include fewer callers in the slamdown information SMS than there is room for.

SlamdownMaxCallsPerCaller Set this parameter to 9, to restrict the call count to one digit even if a caller has called 10 or more times.

SlamdownMaxDigitsInNumber Set this parameter to truncate the callers number, if you need to fit more callers in one SMS

SlamdownTimeOfLastCall Set this parameter show the last or first call to the user.

SlamdownTruncatedNumberIndication If you choose to truncate caller numbers, you may also want to set this parameter, so that it is apparent that the number has been truncated.

5.13.4 Customizing SMS Content for Slamdown Information

The phrase files for language customizing (<NTF_HOME>/templates/*.cphr) have a number of parameters that control the content of slamdown information sent to the subscriber. These parameters should always be customized, the default values will provide some information, but probably not good enough.

The customization of content is completely different depending on whether the information is presented as a list or per slamdown call.

5.13.4.1 Content for Basic Slamdown Information

The template `slamdown` in the language file determines the content of the message sent, when slamdown information is sent for each slamdown call. The template is any string, and these tags can be embedded in the text:

FROM	This tag is replaced by the callers telephone number.
TYPE	This tag is replaced by the call type (voice or video, translated to the users preferred language).

5.13.4.2 Content for Slamdown Information Lists

Each slamdown information message will consist of a header, one or more body parts and a footer. Any of these may be empty. The content of the message is determined by the templates:

- slamdownheader
- slamdownheader1
- slamdownfooter
- slamdownbody

5.13.4.2.1 slamdownheader

Every slamdown information list SMS begins with this text. The slamdown header can not contain any tags.

The default value is “Callers:” \u000a

Note: To insert a special character in a phrase file, enter a backslash followed by the character u and four-digit hexadecimal Unicode code. In particular, a newline can be entered as \u000a.

5.13.4.2.2 slamdownheader1

If the slamdown information list has calls from a single caller, this template is used instead of slamdownheader.

The default value is the value defined by `slamdownheader`

5.13.4.2.3 slamdownfooter

Every slamdown information list message ends with this text. The slamdown header can not contain any tags.

The default value is empty.

5.13.4.2.4 slamdownbody

Between the header and footer in a slamdown information list message, information about one or more callers is inserted. This template determines the content of the information for each caller. In the slamdown body, the following tags can be used:

FROM This tag is replaced by the callers telephone number.

COUNT This tag is replaced by the number of times the caller has made a slamdown call.

DATE= This tag is replaced by the time (and date) of the first/last slamdown call from the caller. See Section 5.7.5.2 DATE= format tag on page 38

The default value is FROM \u000a.

5.13.5 Examples

These examples assume that we have a user that has received a lot of slamdown calls:

- 1 call from 111222
- 46 calls from 333444
- 5 calls from 555666
- 1 call from 888999999999999
- 7 calls from 123456

Table 17 notification.cfg

notification.cfg
SMSStringLength=140
SlamdownMaxCallers=0
SlamdownMaxCallsPerCaller=0
SlamdownMaxDigitsInNumber=0

Table 18 en.cphr

en.cphr
slamdownheader={”Callers:” \u000a }
slamdownbody={ FROM \u000a }
slamdownfooter={”“}

Table 19 charconv.cfg

charconv.cfg
Pack=No

The different examples are described in relation to the following settings, which are the default settings:

5.13.5.1 Default

This example shows what happens if all parameters have their default values, with the exception of SlamdownList, of course.

Table 20 notification.cfg

notification.cfg
SlamdownList=Yes

The slamdown information sent becomes:

```
Callers:
111222
333444
555666
888999999999999
123456
```

5.13.5.2 Extended Templates

This example uses templates with more information than the default settings:

Table 21 en.cphr

en.cphr
slamdownheader={”Calls with no message left:” \u000a } slamdownbody={”[“ COUNT “] ” FROM “, “ DATE=dMMM_HH:mm \u000a }

There will be two slamdown information SMS:

```
Calls with no message left:
[1] 111222, 24May 17:00
[46] 333444, 24May 16:00
[5] 555666, 24May 15:00
[1] 999999123456, 24May 14:00
```

and

```
Calls with no message left:
[7] 123456, 23May 17:00
```

5.13.5.3 Compact Format

By truncating long numbers, limiting call count to 9 and removing unecessary characters from the templates, we can squeeze 5 callers in one SMS

Table 22 en.cphr

en.cphr
slamdownheader= { “Calls with no message left:” \u000a }
slamdownbody={”<[1]” COUNT “> “ FROM “,” DATE=dMMM_HH:mm \u000a }

[1] In the default GSM character set, the character < uses one byte while [uses two bytes.

Table 23 notification.cfg

notification.cfg
SlamdownList=Yes
SlamdownMaxDigitsInNumber=6
SlamdownMaxCallsPerCaller=9

Now, we get everything in one SMS:

```
Calls with no message left:
<1> 111222,24May 1700
<9> 333444,24May 1600
<5> 555666,24May 1500
<1> *23456,24May 1400
<7> 123456,23May 1700
```

5.14 Configuring Mailbox Quota Handling

It is possible to limit how much mail a users mailbox may contain. The limit can be a number of megabytes, a number of messages or both. NTF can be configured to send warning messages when a new message arrives and the mailbox is full or almost full. The warning can be sent even when the mailbox is so full that there is not room for the new message. NTFs quota handling is very flexible and can be configured in many ways.

NTFs quota handling, which is activated when new messages arrive, should not be confused with the daily quota check in the message store. This quota check scans the users mailbox once a day and sends a warning message if the mailbox is full.

When NTF detects that the users mailbox is full or almost full, it can do one of these actions:

- Notify as usual.
- Do not notify at all.
- Notify as usual, but with a quota warning added to the notification (only for SMS notification).

- Send a special quota warning message.
- Notify as usual, and send a special quota warning message.

NTFs quota handling is triggered every time a message arrives to the users mailbox when it is full.

To find out if the users mailbox is full, NTF has to log into the users inbox, so using this function will put extra load on both NTF and MS. If the system has message count enabled, this function does not cost anything extra though, since NTF then logs in to the inbox anyway.

The following parameters in notification.cfg are involved in this functionality:

- CheckQuota
- NumberOfMailQuotaLevel
- QuotaAction
- QuotaTemplate

The phrase files for different languages (e.g. en.cphr and sv.cphr) must also be tailored to the customers needs.

For quota based on number of messages, there are two levels; one when the mailbox contains the maximum number of messages and one when it is full to the level set with `NumberOfMailQuotaLevel`. The handling is the same, but different templates are used to generate the warning message. For quota based on the number of megabytes in the inbox, there is only one level. This level is set in the message store.

5.14.1 Examples

5.14.1.1 Configuring NTF to notify as usual even if the mailbox is full.

- Set CheckQuota to *Off*.

5.14.1.2 Configuring NTF to stop notifications when the mailbox is full.

- Set CheckQuota to *On*.
- Set QuotaAction to *Discard*.

5.14.1.3 Configuring NTF to send a warning message instead of notifications when the mailbox is full.

- Set CheckQuota to *On*.
- Set QuotaAction to *Warn*.
- Configure the template for the warning message:

- Set the content for the template mailquotaexceeded in en.cphr and other phrase files, e.g.

  ```
  mailquotaexceeded={"Your mailbox is nearly full,
  you may be missing messages."}
  ```

5.14.1.4 Configuring NTF to send a warning message with some information about the new message when the mailbox is full.

- Set CheckQuota to *On*.
- Set QuotaAction to *Warn*.
- Create a new template in en.cphr and other phrase files with tags available for notification templates, e.g.

  ```
  quotawarning={"
  You received a message from " FROM " , but your mailbox
  is full, so it may not have been stored."
  }
  ```

- Set QuotaTemplate to the name of the new template e.g.

  ```
  QuotaTemplate=quotawarning
  ```

5.14.1.5 Configuring NTF to send a warning message with some information about the new message when the mailbox contains the maximum (or almost the maximum) number of messages.

- Set CheckQuota to *On*.
- Set QuotaAction to *Warn*.
- Set NumberOfMailQuotaLevel to 90
- Create new templates in en.cphr and other phrase files with tags available for notification templates, e.g.

  ```
  quotawarning={
  "You received a message from "
   FROM ", but your mailbox is full,
  so it was not stored in your inbox."
  }
  ```

  ```
  mailquotahighlevelexceeded={
  "You received a message
  from " FROM ". Your mailbox is almost full,
  please make room for new messages."
  }
  ```

- Set QuotaTemplate to the name of the new template e.g.

  ```
  QuotaTemplate=quotawarning
  ```

5.14.1.6 Configuring NTF to send a warning message and normal notifications when the mailbox is full.

- Set CheckQuota to *On*.
- Set QuotaAction to *NotifyAndWarn*.
- Configure the template for the warning message:
- Set the content for the template mailquotaexceeded in en.cphr and other phrase files, e.g.

```
mailquotaexceeded= {
"Your mailbox is nearly full, you may be missing messages."
}
```

5.14.1.7 Configuring NTF to send normal notifications, but with a quota warning added to messages when the mailbox is full.

- Set CheckQuota to *On*.
- Set QuotaAction to *Notify*.
- Configure a subtemplate called QuotaText in en.cphr and other phrase files, e.g.

```
QuotaText= {
 " (Your mailbox is full)"
}
```

- Add the quota tag (QUOTA_TEXT) to all your templates (usually called c, s h, or e0). Do **not** add the quota tag to any subtemplates.

If the mailbox is full, the text in the quota subtemplate will be inserted into the message at the position of the quota tag. If the mailbox is not full, the quota tag will just be removed.

5.14.2 Daily quota check

The mail server will scan mailboxes daily and warn users about full mailboxes. This functionality has nothing to do with NTFs quota check functionality, but to send the warnings, NTFs general system notification functionality is used.

By default, NTF is delivered with a system notification called mailquotaexceeded, which can be used for this functionality. The specification for this system notification is:

```
[ mailquotaexceeded ]
 Subject=WARNING: Quota exceeded
 Received=[(x.x.x.x)]
```

Example 28 Excerpt from systemnotification.cfg

You should replace x.x.x.x with the IP-address of the mailserver. This is to prevent malicious users outside MoIP from creating mail quota messages.

The default phrase files for different languages are delivered with a system notification template called mailquotaexceeded. As you may notice, this is the same name as the default template for NTFs quota warnings. Thus, the default values will give the users the same warning text when NTF detects a full mailbox when a message arrives, as when the message store detects a full mailbox in its daily scan.

Note: System Notifications like Daily Quota Check are available only for SMS.

5.15 Configuring Call-MWI Notification

The following parameters affect call-MWI notification:

5.15.1 CallMwiCaller

Selects the sender that will appear on the subscribers calling line presentation device.

System The number of the messaging system is used.

Caller The number of the user that left the message is used. If no number is available due to calling line presentation restriction, or because the message is not telephone related (i.e. an ordinary email), the system number is used instead.

Subscriber The subscribers own telephone number is used.

5.15.2 NumberToMessagingSystemForCallMwi

Defines the number of the messaging system that appears on the subscribers calling line presentation device when `CallMwiCaller` is set to “System”

By default, this number is the same as that of `NumberToMessagingSystem`. Since that number is used in mobile notification types, and call-MWI notification is primarily intended for fixed telephones, there may be a need for another number, which is set in this parameter.

5.15.3 SMSTypeOfNumber and SMSNumberingPlanIndicator

The type of number and numbering plan indicator for the called and calling number are taken from these parameters.

5.15.4 Prerequisites For Call-MWI Notification

This notification type is available for a user if the service is active in the users COS and at least one MVAS/MAS that provides the needed XMP service, is registered in MCR.

The user will get a call-MWI notification if

- “CMW” is one of the notification types in the notification filter (filter settings in COS).
- “CMW” is allowed for the terminal type the notification is sent to (NTF parameters FixedTypes and MobileTypes).
- “CMW” is not disabled by the user for the telephone number the notification is sent to (attribute on the billingnumber entry)..

Note: “The notification filter” is short for “the first notification filter that matches the filter requirements on time and mail headers”.

5.16 Configuring character conversion

Internally, NTF uses the Unicode character set. This character set handles almost any character, but requires two bytes per character. Many languages only need a character set with 8 or 7 bits per character, such as the GSM default character set, and can then have messages twice as long. To convert Unicode into some other character set, NTF provides a character conversion that can convert a Unicode character to output bytes. NTF is delivered with general functionality that can convert a character to 0, 1, 2 or more bytes. In addition, it is possible to plug in customized converters if the need arises.

The default converter that comes with the NTF component is configurable and can convert a character to any sequence of output bytes. The conversion is controlled by a configuration file. This file should be called “charconv.<protocol>” where <protocol> is the lower-case name of the protocol used for communication with the SMSC. If that file is not found, the file “charconv.cfg” is used instead.

NTF can try to convert to one character set, but if that fails because of “impossible” characters, reconvert the entire message to UCS2 (SMSC version of Unicode). This is useful, e.g. if you want full-length messages in english, but also want to allow chinese messages in the same system.

5.16.1 charconv.cfg

The NTF converter functionality is controlled by a character conversion configuration file, “charconv.cfg”. This file could look like this:

```
DataCodingScheme=0
Converter=Converter

#Convert unsupported characters to reverse question-marks.
0000-FFFF=60
#Just chop characters that are the same as ASCII
000A,000D,0020-007A=Chop

0009=3C7461623C

0060=Delete
#Currency symbols and accented characters
0040=Keep
00A3=01
00A4=24
20AC=1B65
```

Example 29 charconv.cfg

From this example, you can see that the file contains parameters (DataCodingScheme=0, Converter=Converter) and conversion rules (0040=Keep, 20AC=1B65). There can also be blank lines and comments in the file.

5.16.2 Parameters

The parameters that can be set in `charconv.cfg` are:

DataCodingScheme A number between 0 (default) and 255 defining the character set after conversion. This is normally 0 (SMSC default alphabet). Other values can be used, as defined by the SMSC protocol.

DataCodingSchemeForUcs2 When the converter failed to convert to the configured character set, and had to revert to UCS2 conversion, another data coding scheme must be set. The available values are the same a for `DataCodingScheme`, but for UCS2, the value is usually 8..

Converter Specifies the name of a plug-in converter to replace the general converter. The default is "Converter", which means there is no plug-in and the general converter is used.

Pack NTF normally puts one character in each byte of the converted message. If `Pack` is set to “yes”, NTF will instead put 8 characters in 7 bytes. The value can be “yes” or “no”.

5.16.3 Basic Conversion Rules

The rest of consists of character conversion rules. The most basic conversion rule is of the form:where you specify character with one four-digit hexadecimal number and the output bytes with 0 or more pairs of hexadecimal digits. 20AC=1B65 specifies a conversion from a Unicode euro currency character to the byte sequence giving the euro character from the extension table in the

GSM default character set. You can specify any byte sequence you want, 0009=3C7461623C in makes the tab-character in a message show as the five characters <tab> in the telephone if the default GSM character set is used.

```
character>=<output bytes>
```

5.16.4 Extended Conversion Rules

To make the file smaller and more convenient to write, there are some extensions.

First, you can specify a “character list” of characters that shall be handled the same way. In Example 29 on page 75, “0000-FFFF=60” specifies a default for all characters, so that any characters not mentioned in another conversion rule will give a reverse question mark in the default GSM character set. In the same example it is also shown that character lists can be more complex, like the one in “000A,000D,0020-007A=Chop”.

Note: A character list can not contain spaces.

You can also specify “actions” for a character or character list instead of output bytes, like the line “0060=Delete” in Example 29 on page 75. There are three actions and they are described in this table:

Table 24 Actions

Action	Description
chop	The most significant byte of the unicode character is chopped off and the least significant byte is output.
delete	No byte at all is output, i.e. the character is deleted.
keep	The two bytes of the unicode character are output unchanged.
ucs2	Conversion is aborted, and the entire message is recoded into UCS2, i.e. the “keep” rule is applied to all characters in the message.

Hexadecimal digits and actions are not case sensitive.

5.16.5 Conversion Rule Priority

To make the file convenient to write, the same character can occur many times in different overlapping rules. To resolve the ambiguities this leads to, the following priorities are used:

1 If there are several rules with identical character lists, the last one is used. (Identical means that the sequence of hexadecimal digits, commas and hyphens is identical including case, not just that the lists specify the same characters). Except from that, the order between rules is not significant.

2 A rule specified for a single character has higher priority than any rule where the character is part of a list.

3 Rules with the “chop” action have the highest priority of all rules for a list of characters.

4 Rules specifying an “output byte sequence” have second highest priority of all rules for a list of characters.

5 Rules specifying the “keep” action have the third highest priority of all rules for a list of characters.

6 Rules specifying the “delete” action have the second lowest priority of all rules for a list of characters.

7 Rules specifying the “ucs2” action have the lowest priority of all rules for a list of characters.

8 If no rule at all is specified for a character, the default action is chop which gives a reasonable result for digits and letters and many punctuation characters.

5.16.6 Advice

Multi-byte output sequences take more time to process and require more memory, so use actions or single-byte output sequences if possible. If many multi-byte output sequences are needed, consider using a customized converter if performance or memory usage are unacceptable.

5.17 Configuring MMS notifications

The connection to the MMSC must be configured if MM7 is used as protocol towards the MMSC. The needed configuration parameter are listed in Section 7.1.5 MMS Configuration Parameters on page 100.

NTF can use smil templates for the notification for video and voice messages. The parameter usesmil must be set to true to activate smil templates.

The smiltemplate is located in a zip file together with additional media content.

5.17.1 MMS zip file

The templates for MMS media content are located in one zip file for each supported language and message type, so the user gets notification texts and sounds in his/hers preferred language.

NTF uses different files for video and voice mms notifications. Video content is stored in a file named video-<language>.zip, voice content i stored in a file named <language>.zip. NTF uses the default language if a zip file for the users language can't be found. The default language is english (en).

All language files must be stored in the template directory under <NTFHOME> and have a file name consisting of the language name with the file extension .zip, for example en.zip for the English phrase file for voice and "video-sv.zip" for the Swedish phrase file for video.

Note: The MMS zip files must be located in the specified directory and have the required file names. Otherwise, the NTF component will not find the files.

The zip file is mere a container of one or more different media files.

Consider the following rules when you create a MMS zip file:

- The zip file name must not contain any spaces.
- The zip file MUST contain one and only one file with the extention smil. I.e. <any_name>.smil.

5.17.2 SMIL template

Note: It is important to read and understand how SMIL works before any configuration is done with MMS templates. To obtain information about how SMIL work please consult www.w3c.org

The MMS messages use a subset of the Synchronized Multimedia Integration Language (SMIL) as the presentation language, specified by the World Wide Web Consortium (W3C).

The MMS notifications consist of a slide show of one or several slides.

The templates for an MMS notification (smil document and media files, such as pictures, audio and text files) are located in a zip file. When a notification should be sent, a MMS notification is generated using the media content in the zip file.

The smil document defines how different types of media, e.g. text, sound and pictures should be presented on receiving device.

SMIL stands for Synchronized Multimedia Integration Language and is tag based language (xml). It is the SMIL template, located in the zip file that determines how the different media files in the zip file should be presented on the receiving device. There are SMIL tags available to specify timing, sound and positioning on the screen. However, the actual appearance of the presentation is dependent on the characteristics of the receiving device. A presentation created for one device, may appear differently on another.

A slide in a SMIL presentation contains information about how different multimedia elements should appear on a display. Due to the characteristics of todays mobile devices, such as their screen sizes, audio limitations etc..., there are limits to how MMS messages can be reproduced on the receiving devices. Since NTF doesnt have a terminal database, it is very important that the SMIL templates are configured so that the template agrees with as many mobile devices as possible.

To obtain more information on how SMIL work please consult www.w3c.org. Further more, since different mobile devices handle and implement SMIL differently, please consult the different white papers that are available for different mobile devices if the SMIL template should be modified.

5.17.2.1 An example of a SMIL template

The following example shows how an MMS SMIL template can look like.

```
<smil xmlns="http://www.w3.org/2001/SMIL20/Language">
<head>
       <layout>
               <root-layout width="60" height="60"
                background-color="#CEF6F0"/>
               <region id="From" top="0" left="0"
                height="15" width="60"/>
               <region id="Image" top="15" left="0"
                height="30" width="60"/>
               <region id="Image" top="15" left="0"
                height="30" width="60"/>
       </layout>
</head>
<body>
       <par dur="__LENGTH__s">
               <text src="__FROM__" region="From">
                       <param name="foreground-color"
                        value="#00ccff"/>
                       <param name="textsize"
                        value="small"/>
               </text>
               <img src="image.gif" region="Image"/>
               <text src="__DATE__" region="Date">
                       <param name="foreground-color"
                        value="#ff00ff"/>
                       <param name="textsize"
                        value="large"/>
               </text>
               <audio src="__MESSAGE__"/>
```

```
        </par>
</body>
</smil>
```

- This SMIL example defines a one slide show presentation for text, image and audio.
- This slide is displayed for __LENGTH__ seconds and includes the image en_image.jpg. Below the image, the text defined by __FROM__ is displayed, in small size and with colour #0000ff. Below that text, the text defined by __DATE__ is displayed, in small size and with colour #0000ff. Simultaneously an audio file is played.
- __LENGTH__, __FROM__ and __DATE__ are SMIL template strings.

These will be explained in the following chapter:

5.17.2.2 SMIL template strings

A SMIL template string is a string that defines parts of the content in SMIL template and in the MMS content that is displayed on the receiving device.

The following SMIL template strings can be inserted into the SMIL template (the file with extension file with extension <any_name>.smil

Table 25 SMIL Template Strings

Tag	Description
__DATE__	Indicates that information about when the message was deposited should be displayed on the receiving device. The format is based on the subscriber's preference.
__FROM__	Indicates that information about the senders phone number should be displayed on the receiving device . If the senders phonenumber could not be determined the default system number will be used.
__MESSAGE__	Indicates that the message should be transcoded and should be included in MMS notification. Voice messages is encoded to amr and Video messages are transcoded to 3gp.
__LENGTH__	Duration of a slide. If present within the smil template, __LENGTH__ will be replaced with the length of the voice or video message in milliseconds.

Note:

- The __LENGTH__ template string can only be present within a SMIL dur parameter. I.e. <par dur="__LENGTH__s">
- The __DATE__, __FROM__, __MESSAGE__ and __COUNT__ template strings can only be present within a SMIL src parameter. I.e. <audio src="__MESSAGE__"/>
- The __DATE__, __FROM__, __MESSAGE__, __COUNT__ and __LENGTH__ template string can occur several times within the smil template.

5.17.3 Update a MMS zip template for a specific language

Simply create a new zip file with a new set of media files that should be be a part of the MMS notification.

Add or remove media files to the zip file and/or modify the SMIL template (if necessary) so that it corresponds to media files that should be included in the MMS zip file and to the MMS template strings. Copy the new zip to template directory and restart NTF.

5.18 Configuring Autoprint of New faxes

When a new fax arrives in a subscribers inbox, NTF can automatically forward it to a fax machine for printing. The fax is printed after all notifications have been sent. When the fax has been printed, NTF marks it as seen, in the subscribers inbox.

5.18.1 Pre-requisites

- The user must have autoprint of fax active. Autoprint is active if it is enabled in the user entry, or blank in the user entry and enabled in the CoS.
- The user must have set a default fax print number, and the number must be another number than the users inbound fax number.
- If subscriber external prefixes are used, a prefix for the subscriber must be available via ESI.
- The MS must be set up so that mails to the host "mfc" arrive to the MFC..

5.18.2 Enabling the Function

When all pre-requisites are fulfilled, faxes will be autoprinted.

5.18.3 Customizing the Function

You customize the function with the following parameters in NTFs configuration file (`<NTF_HOME>/cfg/notification.cfg`):

FaxPrintIp Internal prefix which is added to the fax number.

FaxPrinSepSize The number of digits in the subscriber external prefix.

6 Traffic Data Handling

Events are sent as Accounting Request packets following the RADIUS-MA protocol.

The attributes field in the packet contains a number of attributes with accounting details. Each attribute contains the following data: attribute number (one octet), length (one octet) and value (0-253 octets).

The Accounting-Status-Type attribute is always set to Start-Stop (12), to indicate an instantaneous event, combining the start and end of service delivery.

The following attributes are reused from the RADIUS protocol and are required in the request packet, that is common for all events.

Table 26 RADIUS Attributes

Attribute name	Description	Value {example}
MA-Mailbox-Id	Identifies the mailbox.	{1234567}
MA-SAS-Identifier	Name of the component as registered in MCR.	{ntf1@hostname.domain}
Acct-Status-Type	The type of event.	12 (Start-Stop event)
Acct-Session-Id	Random value that is used to identify a session.	<random value>
Vendor-Specific	The MoIP specific attributes are part of the vendor specific attribute.	193

The MoIP specific attributes are part of the Vendor-Specific attribute. The following MoIP specific attributes are required in the request packet, that is common for all events.

Table 27 MOIP Specific Attributes

Attribute name	Description	Value {example}
Escape-To-Product	This attribute tells that the following attributes are specific for the product.	4
MAE-Time-Of-Event	Timestamp for the event.	<timestamp>

Note: The complete set of attributes in a RADIUS-MA request packet is made up of the attributes described for an event together with the required (common) attributes.

6.1 Notification Events

6.1.1 SMS Notification

Table 28 SMS Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Description	Telephone number of caller for voice message notifications.	Default: not present. If configured: 12345678

6.1.2 E-mail Notification

Table 29 E-mail Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	14 (Notification by E-mail)
MAE-Event-Description	Telephone number of caller for voice message notifications.	Default: not present. If configured: 12345678

6.1.3 Slamdown Information

Table 30 Slamdown Information Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)

Attribute name	Description	Value {example}
MAE-SAS-Port-Type	Type of port.	1 (SMS) 14 (Notification by E-mail)
MAE-Event-Reason	Why the event was sent	1 (slamdown information)

6.1.4 System Reminder

Table 31 System Reminder Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	2 (reminder)

6.1.5 SMS Replacing Outdial

Table 32 SMS Replacing Outdial Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	3 (outdial replacement)

6.1.6 Fax print failed notification

Table 33 Fax Print Failed Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS) 14 (Notification by E-mail)
MAE-Event-Reason	Why the event was sent	4 (fax print has failed)

6.1.7 SMS Phone On (SMS type 0)

Table 34 SMS Phone On Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	5 (PhoneOn)

6.1.8 Reminder SMS

Table 35 Reminder SMS Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	13 (Reminder SMS)

6.1.9 Update SMS

Table 36 Update SMS Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	14 (Update SMS)

6.1.10 Cancel SMS

Table 37 Cancel SMS Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	15 (Cancel SMS)

6.1.11 Missed Call Information

Table 38 Missed Call Information Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	1 (SMS)
MAE-Event-Reason	Why the event was sent	6 (MissedCallInformation)

6.1.12 WAP Push Notification

Table 39 WAP Push Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	2 (WAP)

6.1.13 MMS Notification

Table 40 MMS Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	4 (MMS)

6.1.14 MWI Notification

Note: The MWI function applies only to Messaging-over-IP solutions that include a voice application.

Table 41 MWI Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	5 (MWI)

6.1.15 Outdial Notification

Note: The Outdial function applies only to Messaging-over-IP solutions that include a voice application.

Table 42 Outdial Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	3 (Outdial)

6.1.16 Pager Notification

Table 43 Pager Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	8 (Pager)

6.1.17 Call MWI Notification

Table 44 Call MWI Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	9 (Call MWI)

6.1.18 Flash Notification (SMS class 0)

Table 45 Flash Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	11 (Flash)

6.1.19 Wireline Notification

Table 46 Wireline Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	12 (WMW)

6.1.20 Fax Autoprinted

Table 47 Fax Autoprinted Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	13 (Fax)

6.1.21 Failed Notification

Table 48 Failed Notification Event

Attribute name	Description	Value {example}
MAE-Event-Type	Type of operation.	7 (Send)
MAE-Object-Type	Type of object.	3 (Notifier)
MAE-SAS-Port-Type	Type of port.	One of the port numbers above, depending on what type of send that failed
MAE-Terminate-Cause	Type of failure	5 (notification expired) 6 (general error)
MAE-Event-Description	Failure description	discarding bad message: ERROR CODE [0x0B] FROM SMSC (invalid destination address) when sending SMS to {SMSAddress: 1,1,12ggH89}

7 Appendix

7.1 Configuration Parameters

This section describes parameters in `notification.cfg`.

The parameters are grouped in tables for different types of parameters. Some parameters have a short description in the table. If there is no description in the table, it can be found in Section 7.1.12 Parameter Descriptions on page 110.

7.1.1 Miscellaneous Configuration Parameters

Table 49 Miscellaneous Configuration Parameters

Parameter	Description	Values
LogicalZone	Name of the logical zone this NTF belongs to (if any)	String Default: "unspecified".
Netmask	The netmask of the the local host network. This parameter is used when determining whether a component (e.g. MVAS) is local or non-local, but only if LogicalZone is not set.	a.b.c.d where a,b c and d are numbers 0..255. Default: `255.255.255.0`
NumberToMessagingSystem	This should be a telephone number that can be used to call the messaging system. It may be used in e.g. pager notifications.	Range: NA Default: `133`
InternalQueueSize	Number of e-mail messages that NTF can buffer before being processed.	Range: 1..100 Default: `5`
LogLevel	The log level to be used by the notification process, i.e. level of detail written to the log file (LogFile):	Range: 0 no logging (off) 1 error level 2 verbose level 3 debug level Default: **1**

Table 49 Miscellaneous Configuration Parameters

Parameter	Description	Values
LogSize	The maximum size in bytes for the notification log file (LogFile). Once the size has been reached, the log file will be moved to a file with “.sav ” as file name extension.	Range: 100000..1000000000 Default: **10000000**
FlushLog	If true, NTF will flush the log output after each line, otherwise, it will only flush the log after error lines.	Range: true-false Default: **false**
WatchdogLogLevel	The log level to be used by the ntfWatchdog.	Range: 0 no logging 1 error level 2 warning level 3 information level Default: **1**
WatchdogLogSize	The maximum size of the logfile for the watchdog	Range: 100000..1000000000 Default: **10000000**
EsiFailureAction		See description in Section 7.1.12.8 EsiFailureAction on page 112
EsiSystem	Indicates wheather there exists a ESI in the system or not. At present this is used in CFU-check only. Other Esi-communication is determined by input to NTF.	Range: yes-no Default: **no**
PathToSnmpScripts	Where to find snmp scripts.	Range: valid port number Default: **/apps/snmp/scripts**
SnmpAgentPort	Port number for management and monitoring.	See Section 7.1.12 Parameter Descriptions on page 110
SnmpAgentTimeout	The time in milliseconds the ntfagent process waits for updated MIB parameter values.	Range: NA Default: **10**

Table 49 Miscellaneous Configuration Parameters

Parameter	Description	Values
VeryOldMessage	Messages older than this time will be deleted by a clean-up function, to ensure that messages do not linger in the notification mailboxes for ever. This parameter shall not be confused with the expiry time for SMS messages, that is stored in the user database.	Seconds Range: 3600..604800 Default: **259200** (3 days)
CheckUserHasMessage		See description in Section 7.1.12.4 CheckUserHasMessage on page 111
CheckUserHasMessage RetryCount		See description in Section 7.1.12.5 CheckUserHasMessag eRetryCount on page 111
CheckQuota		See description in Section 7.1.12.2 CheckQuota on page 110
QuotaAction		See description in Section 7.1.12.18 QuotaAction on page 115
QuotaTemplate		See description in Section 7.1.12.19 QuotaTemplate on page 116
NumberOfMailQuotaLev el		See description in Section 7.1.12.17 NumberOfMailQuotaLevel on page 115
NotifThreads	Maximum number of notifications that should be processed simultaneously.	Range: 3..30 Default: **10**

Table 49 Miscellaneous Configuration Parameters

Parameter	Description	Values
RetryInterval	If NTF fails to send a notification, e.g. because it can not contact the SMSC, the first few notifications are queued in NTF but the rest are cancelled and later retrieved from the store again to be retried. This parameter defines the interval between retries. This parameter should be long enough that it is likely that the error has been corrected, but not too long, since then the accumulated messages will cause temporarily very high load on NTF.	Seconds Range: 60..3600 Default: **600**
MailMemoryMegabyte	This parameter limits the amount of memory used to store notifications that could not be delivered due to communication problems with an external system. If the limit is exceeded, NTF stops checking for new notifications until the communication with the external system is OK again and memory has been freed.	Range: 10-50 Default: **20**
FaxPrintIp		See description in Section 7.1.12.9 FaxPrintIp on page 113
FaxPrintSepSize		See description in Section 7.1.12.10 FaxPrintSepSize on page 113
FaxProvisioningPrefixSize		See description in Section 7.1.12.11 FaxProvisioningPrefixSize on page 113
FaxPrintANumberPrefixSize	Defines the size of a prefix (e.g. size of country code) that is removed from the subscribers telephone number, when used as A-number in fax print requests	Range: 0-5 Default: **0**

Table 49 Miscellaneous Configuration Parameters

Parameter	Description	Values
UseCallerInEventDescription		See description in Section 7.1.12.46 UseCallerInEventDescription on page 125
MessageIdValidate	The time in seconds an autoprint fax notification shall be delayed.	Range: yes-no Default: yes
DelayAutoprintFaxSms		See description in Section 7.1.12.12 DelayAutoprintFaxSms on page 114
AutoForwardedMessages	If set to true NTF will check for receivers that are autoforwarded from other mail accounts.	Range: true|false Default: **false**
DefaultNotificationFilter	The notification filter to use if no filter in the user entry or COS matches. It is not recommended to change this parameter.	Range: N/A Default: **1;n;a;evfm;;;997;;;;;OFF;;**
DefaultNotificationFilter2	The notification filter for slamdowns. It is not recommended to change this parameter.	Range: N/A Default: **1;y;a;s;SMS,EML;slamdown,slamdown;998;;;;;SLAMDOWN;;**
DefaultNotificationFilter3	The notification filter for faxprint fail messages. It is not recommended to change this parameter	Range: N/A Default: **1;y;a;p;SMS,EML;faxprintfail,faxprintfail;999;;;;;FAXPRINTFAIL;;**

7.1.2 Directory Server Configuration Parameters

Table 50 Directory Server Configuration Parameters

Parameter	Description	Values
MURPort	The LDAP port number for the directory server (MUR).	Range: valid port number Default: **389**
MURHost	Full DNS name of the host where the directory server (MUR) is installed.	Range: NA Default: **murhost**

Table 50 Directory Server Configuration Parameters

Parameter	**Description**	**Values**
MURUserName	The user that does reading and writing to MUR. Enter the full DN or one of the two predefined names. **Admin** = uid=admin,ou=Directory Administrators,<SEARCH BASE> **Directory Manager** = cn=Directory Manager	Range: NA Default: **Directory Manager**
MURPassword	Password for the MUR user.	Range: NA Default: **emmanager**
MurRetryInterval	If the connection to MUR is lost, then NTF tries to re-establish the connection with this interval.	Seconds Range: 1..60 Default: **10**
MCRPort	The LDAP port number for the configuration directory server (MCR).	Range: valid port number Default: **389**
MCRHost	Full DNS name of the host where the configuration directory server (MCR) is installed.	Range: NA Default: **mcrhost**
MCROrg	The search base for MCR.	Range: NA Default: **o=config**
MCRUserName	The name of the MCR admin.	Range: NA Default: **IComponent**
MCRPassword	The password for the MCR admin.	Range: NA Default: **abc123**
MCR_INSTANCE_NAME	Name of the NTF component instance as registered in MCR. This name should be defined as follows: **ntf1@hostname.domainname**, where **hostname.domainname** must be possible to look up by use of DNS from the clients utilizing this service. Note that a host with several network interfaces has several hostnames.	Range: NA Default: **ntf1@host1. yourcompany.com**

Table 50 Directory Server Configuration Parameters

Parameter	Description	Values
SearchBase	This value must be changed for each customer. The search base used for all LDAP searches in MUR.	Range: NA Default: **ou=site1,o=userdb**
McrExpiryTime	The NTF component caches the information fetched from MCR about other components in the system. This parameter determines how often this information is refreshed. This parameter also determines how often NTF reconnects to MUR.	Seconds Range: 60..36000 Default: **300**
MurMaxConn	Maximum number of simultaneous connections to the user directory. This value should be the same as NotifThreads or a little less.	Range: 1..30 Default: **10**
DefaultDateFormat	The format used for date information in notifications, if the information is missing in the user entry, or if the notification is for a user that is not a subscriber.	Range: mm/dd/yyyy dd/mm/yyyy dd/mmm/yyyy yyyy/mm/dd yyyy-mm-dd dd/mmm/yyyy dd.mm.yyyy yyyy_mm_dd Default: **yyyy/mm/dd**
DefaultTimeFormat	The format used for time information in notifications, if the information is missing in the user entry, or if the notification is for a user that is not a subscriber.	Range: 12 or 24 Default: **24**
DefaultLanguage		See description in Section 7.1.12.6 DefaultLanguage on page 111
MurConnectionJNDI	NTF can use Netscape LDAP or Sun JNDI to connect to MUR. Set this parameter to true for JNDI and false for Netscape LDAP.	Range: true|false Default: **true**

7.1.3 IMAP Configuration Parameters

Table 51 IMAP Configuration Parameters

Parameter	Description	Values
IMAPRetryInterval	How long time NTF will wait until it retries to connect to the notification mailbox (in MS). This parameter will have effect only when NTF has lost its IMAP connection to MS.	Seconds Range: 1..999 Default: **30**
IMAPThreads	The number of `IMAPThreads` connecting to the notification mailboxes. There should be one thread per mailbox, so if you define 6 mailboxes in the message store, you should define 6 `IMAPThreads` in NTF. The number of threads are directly related to the notification mailbox name. For example: `IMAPThreads = 6 gnotification1_0` ... `gnotification1_5`	Range: 3..30 Default: **30**
IMAPHost	Full DNS name of the host where the message store (MS) is installed.	Range: NA **No Default**
IMAPUserName	This value must be changed for all NTFs in a system, but the first. This is the basename for the notification mailboxes. The actual mailbox names will be of the format: <IMAPUserName> _<Thread_no> For example, if the IMAPUserName = gnotification1 and IMAPThreads = 2 then the mailbox names will be: gnotification1_0 gnotification1_1	Range: NA Default: **gnotification1**
IMAPPassword	The password to be used for the notification mailbox (IMAPUserName).	Range: NA Default: **system**
IMAPPostmaster	The postmaster address. Messages sent to this address will not get notification. The format of the address is: **`postmaster@<host name>.<domain name>`**	Range: NA Default: **postmaster**

Table 51 IMAP Configuration Parameters

Parameter	Description	Values
IMAPPollInterval	The interval NTF uses to poll the notification mailboxes.	Seconds Range: 1..99 Default: **10**
IMAPPort	The IMAP port number.	Range: valid port number Default: **143**
ImapRootUserName	The user ID that will be used for logging in to the Messaging Server. This user is the ServiceAdministrator that was defined during MS installation. This user can be found in the user directory server, MUR.	Range: NA Default: **ServiceAdmin**
ImapRootPassWord	The password for the ImapRootUserName.	Range: NA **admin**
ImapTimeout	How long time NTF will try to connect to the IMAP server, and also wait for a response.	Milliseconds Range: 1000..30000 Default: **5000**
IMAPBatchSize	The number of message identities read from the store in one operation.	Range: 1-1000 Default: **100**
DeletesBeforeExpunge	When NTF has deleted this many messages, an expunge operation is done.	Range: 1-1000 Default: **50**
MaxTimeBeforeExpunge	If this number of seconds have passed since the last expunge, a new expunge operation is done even if fewer than deletesBeforeExpunge messages have been deleted.	Range: 10-3600 Default: **300**
NewMailFolders		See description in Section 7.1.12.16 on page 115

7.1.4 WAP Push Configuration Parameters

Table 52 WAP Push Configuration Parameters

Parameter	Description	Values
WAPPushUserName	User ID to be used for logging into the WAP gateway.	Range: NA **No default**

Table 52 WAP Push Configuration Parameters

Parameter	Description	Values
WAPPushPasswd	Password to be used (by WAPPushUserName) for logging into the WAP gateway.	Range: NA **No default**
WAPPushUrlSuffix	`WAP Push` application on the WAP gateway. For example, if the `WAP Push` application is named **`wap_push_appl`**, and is located in the `wap_push_dir` directory on the host **`host10.yourcompany.com`**, then the URL suffix is: `/wap_push_dir/wap_push_appl` (and the URL will become: `http://host10.yourcompany.com/wap_push_dir/wap_push_appl`).	Range: NA Default: **/wap_push_dir/ wap_push_appl**
WAPPushRetrievalHost	The name of the host used for WAP push retrieval. This should be a name that can be reached by the users, i.e. from an external network. It can be the name of an MWS host or a load balancer for example.	Range: NA **No Default**

7.1.5 MMS Configuration Parameters

Table 53 MMS Configuration Parameters

Parameter	Description	Values
MMSWhiteList	The domain that will be used in the `Mail From:` field of the envelope when MMS is sent via SMTP. Some MMSCs need to be configured to accept this domain.	Range: NA Default: An IP address of the NTF host.
MMSVersion	MMSheader that states which MMS version that should be used. This header will be interpreted by the MMS-C.	Range: NA Default: Single space character.
MMSUserName	The user name for the external MMS-C. (Only valid for MM7 enabled servers that uses HTTP) If one single NTF instance shall use several MMS-Cs , the same NTF user ID and password (as specified by the MMSUserName and MMSCPassword parameters) must be used for all MMS-Cs.	Range: NA **No default**

Table 53 MMS Configuration Parameters

Parameter	**Description**	**Values**
MMSCPassword	The password for the external MMS-C. (Only valid for MM7 enabled servers that uses HTTP)	Range: NA **No default**
UseSmil	If the MMS voice notification should include smil formatting.	Range: true|false Default: false
MMSMaxVideoLength	Limits the max length of the Video sent in the MMS notification.	Range: -1 = no limit 0 = only first frame 1- = limit to X seconds Default: -1
MMSMaxConnections	Max number of outgoing MMS notifications.	Range: 1-1000 Default: 10
MmscVaspId	The VASP ID parameter for MM7	Range: NA Default: NTF
MmscVasId	The VAS ID parameter for MM7	Range: NA Default: ntf@<hostname>
MarkAsRead	Marks a message for the user as seen if the complete message has been sent out as MMS.	Range: true|false Default: false
MMSPostMaster	Email address to use as receiver of all MMS notifications.	Range: N/A Default: " "
UseMMSPostMaster	If set to YES MMSPostMaster has to be set with a valid email address. All MMS notifications will now be sent to this address.	Range: true|false Default: **false**

7.1.6 SMS Configuration Parameters

Table 54 SMS Configuration Parameters

Parameter	**Description**	**Values**
SMSCTimeout	The time for the SMSC to respond to an SMPP message.	Seconds Range: 1..300 Default: **30**

Table 54 SMS Configuration Parameters

Parameter	Description	Values
ReplaceNotifications	This parameter lists the SMS templates[1] that are replaceable. An SMS generated from a replaceable template will replace all previous SMS from the same template to the same mobile. The default value is suitable if the templates for c and mailquotaexceeded generate messages like `"You have 3 new messages"` and `"Your mailbox is almost full"`	Range: Up to seven different notification content values. Default: **c, mailquotaexceeded**
DisableSmscReplace	If DisableSmscReplace is set to yes, replace will only be in the telephone, not in the SMSC. NTF uses the service type to control replace in the SMSC. If this is not properly supported by the SMSC, set DisableSmscReplace to yes.	Range:yes, no, true, false, on and off. Default: **no**
MessageCount	This is a global parameter that will turn the message count feature on or off. [2]	Range: ON/OFF Default: **ON**
MsgCntByType	If `MessageCount` is “ON”, the `MsgCntByType` parameter determines whether or not the count of new messages in the mailbox will regard the message type (voice, fax, e-mail, video). If `MsgCntByType` is set to “YES”, the number of new messages will be counted by message type (voice, fax, e-mail, video). The total number of new messages will also be counted. If `MsgCntByType` is set to “NO”, the total number of new messages will be counted, regardless of message type.	Range: YES/NO Default: **YES**
SMSString Length	The maximum number of bytes allowed for one SMS message.	Range: 0..140 Default: **140**

Table 54 SMS Configuration Parameters

Parameter	Description	Values
SMESystemID	The system name for the External Short Message Entity (ESME), i.e the SMSC. If a single NTF instance shall use several SMSCs, you can only specify one account with the SMESystemID and SMEPassword parameters. To use different credentials on different SMSCs, you must use the MCRAddon file (Section 5.6.2 MCRAddon.xml on page 28).	Range: NA **No default**
SMEPassword	The password for the ESME used for connecting to the SMSC.	Range: NA **No default**
SMESystemType	Identifies the type of ESME requesting to bind with the SMSC.	Range: NA Default: **MOIP**
SMEServiceType	Used to specify the type of service an SMS belongs to[3]. The default value “VMN” (voice-mail notification) should normally be used	Range: NA Default: **VMN**
SMEServiceTypeForMwi	Used to specify the type of service for MWI on and MWI off messages.	Range: NA Default: **not set**, meaning that the value for SMEServiceType is used
NumberOfSMS	Maximum number of SMS messages that can sent for one notification. If a message is too large for one SMS, it can be split up into several SMS messages.	Range: 1..10 Default: **5**
BearingNetwork	Identifies the bearer network, CDMA2000 or GSM.	Range: GSM/CDMA2000 Default: **GSM**
SMSTypeOfNumber	Describes the default type of telephone number used for SMS and MWI notifications. [4].	Range: NA Default: **1**
SMSNumberingPlanIndicator	Indicates the default numbering plan of the telephone number used for SMS and MWI notifications. [4]	Range: NA Default: **1**
SMESourceAddress	See description in Section 7.1.12.32 SMESourceAddress on page 119	

Table 54 SMS Configuration Parameters

Parameter	Description	Values
SourceAddress_<type>	See description in Section 7.1.12.41 SourceAddress_* on page 122	
SourceAddress_<type>_<cosname>	See description in Section 7.1.12.41 SourceAddress_* on page 122	
SourceAddress_<system notification name>	See description in Section 7.1.12.41 SourceAddress_* on page 122	
SourceAddress_<system notification name>_<cosname>	See description in Section 7.1.12.41 SourceAddress_* on page 122	
SMESourceTON	See description in Section 7.1.12.34 SMESourceTON on page 120	
SMESourceNPI	See description in Section 7.1.12.33 SMESourceNPI on page 120	
SMSMaxConn	The maximum number of connections NTF is allowed to make to one SMSC This number should not exceed the maximum allowed by the SMSC.	Range: 1..30 Default: 10
SMSQueueSize	See description in Section 7.1.12.40 SmsQueueSize on page 122	
AllowedSmsc	When a subscribers user entry specifies the SMSC “auto”, or when SMS shall be sent to non-subscribers, NTF selects an SMSC from those found in MCR. This parameter can be set to restrict the selection to a subset of the SMSCs, e.g. to avoid using a test SMSC that is in MCR.	Range: a list of space-separated SMSC names, e.g. **`alfa beta gamma`** Default: an empty list, which means NTF is allowed to use all SMSCs registered in the component register.
DiscardSmsWhenCountIs0	If this parameter is true, NTF will discard an SMS notification if the users new messages have been counted and that count is 0. This currently applies to all SMS messages, also those not related to mail arriving to the users inbox, so it should normally be left at the default value.	Range: YES/NO Default: **NO**

Table 54 SMS Configuration Parameters

Parameter	Description	Values
KeepSmscConnections	If this parameter is false, NTF will drop connections to the SMSC if the traffic is low. If the parameter is set to true, NTF will keep the connections open even without traffic	Range: YES/NO Default: **NO**
MwiOffCheckCount	See description in Section 7.1.12.14 MwiOffCheckCount on page 114	
MwiServers	See description in Section 7.1.12.15 MwiServers on page 114	
smscbackup	Defines backup-SMSCs. The value is a list of pairs of SMSC names, in this format: smsc1>smsc2 **or** smsc1>smsc2;smsc2>smsc3 The character **>** can be seen as an arrow redirecting traffic from smsc1 to smsc2 and from smsc2 to smsc3.The SMSC pairs are separated with a **;** character.	Range: NA **No default**
SmppBindType	See description in Section 7.1.12.35 SmppBindType on page 120	
SmppErrorCodesIgnored	See description in Section 7.1.12.36 SmppErrorCodesIgnored on page 120	
SmppVersion	The version NTF sends during bind. This does not affect the actual version of smpp that is used. Note that version is in decimal-format.	Range: NA Default: 52
Smshandlerloadbalancing	Loadbalancing the handle of SMS between the SMSC for name and the backup SMSC. When this parameter is true, NTF will switch to the backup SMSC not only when there is a failure, but also when the SMSC is temporarily overloaded.	Range: true|false Default: false
SetReplyPath	Tells how the routing of replies from the telephone shall be routed to NTF.	See description in Section 7.1.12.22 SetReplyPath on page 116

Table 54 SMS Configuration Parameters

Parameter	Description	Values
SmscPollInterval	Tells how often NTF should poll the SMSC, to keep the connection up.	See description in Section 7.1.12.38 SMSCPollInterval on page 121
SmsPriority	See description in Section 7.1.12.39 SmsPriority on page 121	
SmscErrorAction	This is an emergency parameter that can be set if NTF and the SMSC do not agree on error codes. Normally, the value “handle” is used, and NTF does its best to handle the error, by retrying temporary errors and discarding messages that give permanent errors. Some SMSCs generate vendor-specific error codes or give a temporary error code to permanent errors. To stop NTF from retrying permanently failing messages, this parameter can be set to “log” or “ignore”, which makes NTF ignore error codes from the SMSC. “log” logs the error code though.	Range:handlelloglignore Default:handle
SplitMwiAndSms	Setting this parameter to true forces NTF to send one sms-message and one mwi-message if the user has both mwi and sms.	Range: truelfalse Default: false
UseAlternativeFlashDcs	See description in Section 7.1.12.45 UseAlternativeFlashDcs on page 124	
CancelSMSAtRetrieval	See description in Section 7.1.12.1 CancelSmsAtRetrieval on page 110	
CheckTerminalCapability	See description in Section 7.1.12.3 CheckTerminalCapability on page 110	
DefaultTerminalCapability	See description in Section 7.1.12.7 DefaultTerminalCapability on page 111	
SendUpdateAfterRetrieval	See description in Section 7.1.12.20 on page 116	
SendUpdateAfterTerminalChange	See description in Section 7.1.12.21 SendUpdateAfterTerminalChange on page 116	

Table 54 SMS Configuration Parameters

Parameter	Description	Values
UnreadMessageReminderInterval	See description in Section 7.1.12.43 UnreadMessageReminderInterval on page 124	
UnreadMessageReminderMaxTimes	See description in Section 7.1.12.44 UnreadMessageReminderMaxTimes on page 124	
UnreadMessageReminderType	See description in Section 7.1.12.42 UnreadMessageReminderType on page 124	
Validity_Flash	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_SmsType0	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_MwiOn	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_MwiOff	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_MailQuotaExceeded	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_TemporaryGreetingOnReminder	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_VoiceMailOffReminder	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_CfuOnReminder	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_Slamdown	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_UnreadMessageReminder	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_UpdateSMS	See description in Section 7.1.12.47 Validity_* on page 125	
Validity_<system notification name>	See description in Section 7.1.12.47 Validity_* on page 125	

(1) The templates are defined in the phrase files for different languages, e.g. h, s, slamdown, general and ivrtosms.

(2) If both MessageCount and CheckQuota are active, they will be combined so the quota is checked in the process of finding the message count.

(3) For replaceable templates, NTF assigns the service type to other, non-standard values

(4) Defined in Short Message Peer to Peer (SMPP) Specification (http://www.smpp.org)

7.1.7 Slamdown Configuration Parameters

Table 55 Slamdown Configuration Parameters

Parameter	Description	Values
SlamdownList	See description in Section 7.1.12.23 SlamdownList on page 117	
SlamdownMaxCallers	See description in Section 7.1.12.24 SlamdownMaxCallers on page 117	
SlamdownMaxCallsPerCaller	See description in Section 7.1.12.25 SlamdownMaxCallsPerCaller on page 117	
SlamdownMaxDigitsInNumber	See description in Section 7.1.12.26 SlamdownMaxDigitsInNumber on page 118	
SlamdownNumberOfRetries	See description in Section 7.1.12.27 SlamdownNumberOfRetries on page 118	
SlamdownOldestFirst	See description in Section 7.1.12.28 SlamdownOldestFirst on page 118	
SlamdownTimeOfLastCall	See description in Section 7.1.12.29 SlamdownTimeOfLastCall on page 118	
SlamdownRetryInterval	See description in Section 7.1.12.30 SlamdownRetryInterval on page 119	
SlamdownTruncatedNumberIndication	See description in Section 7.1.12.31 SlamdownTruncatedNumberIndication on page 119	
JournalRefresh	See description in Section 7.1.12.13 JournalRefresh on page 114	

7.1.8 Parameters for Outdial Notification

Some parameters in the `notification.cfg` configuration file are used to configure the outdial notification function.

Table 56 Parameters for Outdial Notification

Parameter	Description	Values
DoOutdial	This parameter enables (YES) or disables (NO) the out-dial notification function.	Range: YES/NO Default: **YES**

7.1.9 XMP Configuration Parameters

The XMP protocol is used to communicate with other components, e.g. MVAS/MAS for outdial notification and ESI for external subscriber information. This section defines general XMP parameters. If there are specific parameters for a certain notification type, that parameter overrides these general parameters.

Table 57 XMP Configuration Parameters

Parameter	Description	Values
XmpTimeout	Timeout for connection and response for XMP.	Seconds Range: 5..120 Default: **30**
MaxXmpConnections	The maximum number of simultaneous connections to one host/port combination.	Range: this number should not have to be large, since each connection supports multiple simultaneous transactions Default: **3**
XmpValidity	How long an XMP transaction is useful, i.e. if this time has passed, the transactions is not valid any more.	Seconds Range: 30..3600 Default: **90**
XmpPollInterval	See description in Section 7.1.12.48 XmpPollInterval on page 127	

7.1.10 Pager Notification Configuration Parameters

Table 58 Pager Notification Configuration Parameters

Parameter	Description	Values
PagerPauseTime	The string of digits used to send notifications to a pager system may contain pauses. This parameter defines how long each pause shall be.	Milliseconds Range: 100..10000 Default: **1000**

7.1.11 Call-MWI Notification Configuration Parameters

Table 59 Call-MWI Notification Configuration Parameters

Parameter	Description	Values
CallMwiCaller	See description in Section 5.15.1 CallMwiCaller on page 73	
NumberToMessagingSystemForCallMwi	See description in Section 5.15.2 NumberToMessagingSystemForCallMwi on page 73	

7.1.12 Parameter Descriptions

7.1.12.1 CancelSmsAtRetrieval

This parameter enables or disables cancelling of pending notification messages at retrieval.

On, Yes, True Cancel of pending notification messages at retrieval is enabled.

Off, No, False Cancel of pending notification messages at retrieval is disabled.

The default value is *Off*.

7.1.12.2 CheckQuota

This parameter is the main switch for quota check functionality. If it is *off*, NTF will never even check the status of the users mailbox, so there will never be any special quota handling. Turning it on will enable the check, and increase the load on NTF and MS. The value is not case-sensitive

On, Yes, True Log in to the users mailbox and check for quota warning.

Off, No, False Do not check for quota warning.

The default value is *Off*.

7.1.12.3 CheckTerminalCapability

This parameter enables or disables terminal awareness functionality.

On, Yes, True Terminal awareness functionality is enabled.

Off, No, False Terminal awareness functionality is disabled.

The default value is *Off*.

7.1.12.4 CheckUserHasMessage

If this parameter is true, NTF will check every notification by logging in to the users mailbox and verifying that the message exists and is unread. If not, the notification is cancelled. The advantage with this check is that the user is not notified when the message has already been read, or when the message was never stored in the users mailbox because it would exceed the users quota. The disadvantage is that it puts extra load on the message store and user registry.

On, Yes, True Only notify about unread messages that are in the users mailbox.

Off, No, False Notify for all messages that have arrived to the system.

The default value is *Off*.

7.1.12.5 CheckUserHasMessageRetryCount

This parameter tells how many times NTF will try to find a message in the subscribers inbox before the message is finally removed from the store.

-1 Retry until the message has arrived in the users mailbox or the message is too old.

0 The message is removed with no retry.

199 The message is retried X times until it is removed.

The default value is **0**.

7.1.12.6 DefaultLanguage

If the user has no preferred language, the language in this parameter is used instead.

<valid language code> The language is used for user that do not have a preferred language.

<not valid language code> English is used for users that do not have a preferred language.

A valid language code is one for which there is a phrase file in <NTF_HOME/templates.

The default value is *en*.

7.1.12.7 DefaultTerminalCapability

This parameter indicates the default terminal capabilties that shall be used if the terminal capability can not be determined. This parameter is only valid if CheckTerminalCapability is set to On. It is a string with three settings in a comma separated list. Each setting has a name, an equals sign and a value. The names are.

MWI Terminal MWI capability.

Flash Terminal flash capability.

Replace Terminal replace capability.

And the value is one of:

On The terminal supports this feature

Off The terminal does not support this feature

The default value is *mwi=off,flash=on,replace=on*.

7.1.12.8 EsiFailureAction

Determines what to NTF shall do in case the notification is affected by roaming status and the roaming status of the subscriber can not be determined. The value is not case sensitive.

Retry Retry the notification later, by keeping the mail in the store and activating it again within 10 minutes (configurable). If some notification type that is not dependent on location succeeds, this retry is cancelled.

Home Assume the subscriber is home, i.e. disable the notification types the subscriber wants disabled in the home network.

Roam Assume the subscriber is roaming, i.e. disable the notification types the subscriber wants disabled when roaming.

Both Assume the subscriber is both home and roaming, i.e. disable all notification types that depend on roaming status.

None Assume the subscriber is neither home nor roaming, i.e. enable all notification types that depend on roaming status.

AllowNotificationTypes=<list of types> Of the notification types that depend on roaming status, allow those in the list, i.e. types in the list are enabled and all other types are disabled.

Example:

```
EsiFailureAction=AllowNotificationTypes=SMS,MWI
```

The default value is *Retry*.

7.1.12.9 FaxPrintIp

This parameter specifies the Internal Prefix for faxprint.

<any string>	This string is prepended to the number of the destination fax machine. This should normally be a number, but any string is allowed.
<empty string>	No prefix is added.

The default value is the empty string.

7.1.12.10 FaxPrintSepSize

This parameter specifies the size of the Subscriber External Prefix for faxprint.

0	No subscriber external prefix is added.
1 and up	The subscriber external prefix is retrieved from an external subscriber information system. Then it is made exactly the number of digits specified by this parameter by zero padding to the left, or truncating to the right. Finally, it is prepended to the number of the destination fax machine.

The default value is *0*.

7.1.12.11 FaxProvisioningPrefixSize

Some customers have fax-only subscriptions. To let one subscriber have both a voice and a fax-only subscription (with the same telephone number), a provisioning prefix is added to the telephone number of all fax-only subscriptions. When communicating with external systems, NTF must use the telephone number of the real telephone, i.e. remove the provisioning prefix. This parameter specifies the size of the prefix to remove.

This parameter is only effective for subscribers that have fax enabled but not the voice service.

0	No provisioning prefix is removed.
1 and up	This many characters in the beginning of the provisioned telephone number are removed, before use in communication with external systems.

The default value is *0*.

7.1.12.12 DelayAutoprintFaxSms

The time an SMS notification about a new fax message is delayed, if the fax is also autoprinted.

The time is set in seconds.

The default value is *3*.

7.1.12.13 JournalRefresh

Specifies how often the slamdown and emaillist data files should be refreshed. The datafiles are trimmed and information not longer needed are removed when a refresh is done.

The refreshtime is set in seconds.

The default value is *1800*.

7.1.12.14 MwiOffCheckCount

When a user logs out of his mailbox, NTF normally sends an MWI off (if the user has MWI). If this parameter is on, NTF counts the unread messages in the users inbox and sends MWI off if the count is 0, but an updated MWI on if not all messages have been read.

On, Yes, True	Count the users unread messages and send MWI off only when the count is 0.
Off, No, False	Always send MWI off when the user logs out from the mailbox.

The default value is *Off*.

7.1.12.15 MwiServers

This parameter tells NTF to send MWI requests to certain SMPP servers. The parameter connects regular SMSC to MWI servers. When an MWI notification is sent, NTF first checks which SMSC the user has. NTF checks this parameter for a MWI server to use. If no MWI server is defined, the regular SMSC is used, otherwise the MWI server is used.

The format of the parameter is a list of many to many connections.

```
SMSCA>MWIA,MWIB;SMSCB,SMSCC>MWIC
```

Example 30 MwiServers example

This means that SMSCA will use the mwi servers MWIA and MWIB. SMSCB and SMSCC will use MWIC.

The default value is "".

7.1.12.16 NewMailFolders

This parameter determines if NTF will look in the inbox only or in the Voice, Video, Fax and Inbox.

The parameter can have the following values and are case sensitive:

single NTF looks in the inbox only.

multiple NTF looks in different folders. All messages in the inbox are treated as e-mails

unknown NTF looks in different folders. Messages in the inbox are checked to see the type of message.

The default value is **single**.

7.1.12.17 NumberOfMailQuotaLevel

The CoS of a user can limit the number of messages the user can have. If `CheckQuota` is set, NTF will warn when the mailbox is full. If `NumberOfMailQuotaLevel` is set, NTF will also warn (with another message) when the mailbox is partly full. This parameter defines the percentage for “partly”.See Section 5.14 Configuring Mailbox Quota Handling on page 69

0-99 When the mailbox contains at least this many percent of the allowed number of messages, a quota warning will be sent for each new message that arrives.

-1, 100 and up Quota warning only when the malbox is full.

7.1.12.18 QuotaAction

QuotaAction controls NTFs handling of a notification if the users mailbox is full. It is a string that can be *Notify*, *Discard*, *Warn* or *NotifyAndWarn*. The value is not case-sensitive.

Notify means NTF will send normal notifications to the user.

Discard means NTF will neither send any normal notifications nor quota warnings to the user.

Warn means NTF will send a quota warning message to the user instead of the normal notification.

NotifyAndWarn means NTF will send normal notifications to the user, plus a quota warning message. If the normal notification also uses SMS (or E-mail), the user will receive two messages.

The default value is *Warn*.

7.1.12.19 QuotaTemplate

QuotaTemplate tells NTF the name of the template to use for the quota warning message sent when the mailbox is full. The template is defined in the phrase files for the different languages. It can be any valid template name (see Section 5.7.4 Template Strings on page 33) The value is case-sensitive.

Note: The name of the other template (when the mailbox is not quite full), `mailquotahighlevelexceeded`, is not configurable. The name of the “full” template is configurable to allow it both to be any template, and to be the same as the template used when the daily quota check finds a full mailbox.

The default value is *mailquotaexceeded*.

7.1.12.20 SendUpdateAfterRetrieval

This parameter enables or disables sending of an update SMS after retrieval.

On, Yes, True	Update SMS after retrieval is enabled
Off, No, False	Update SMS after retrieval is disabled.

The default value is **off**

7.1.12.21 SendUpdateAfterTerminalChange

This parameter enables or disables sending of an update SMS after terminal change.

On, Yes, True	Update SMS after terminal change is enabled
Off, No, False	Update SMS after terminal change is disabled.

The default value is **off**

7.1.12.22 SetReplyPath

This parameter is used to enable or disable the ReplyPath feature in the SMSC. NTF sends this flag to the SMSC to tell it how a response from the phone shall be routed.

The value must be provided by the operator.

On, Yes, True, 1	Responses to SMS type 0 will be routed "back to where it came from", i.e. the SMSCs involved will keep track of how the response should come back to NTF.
Off, No, False, 0	The responses will be routed to a default SMSC, stored in the users SIM card.

The default value is *No*.

7.1.12.23 SlamdownList

The slamdownlist parameter determines if an SMS with slamdown information should be sent for each call, or if slamdown information should be collected in a list that is sent as one or more SMS when the phone is turned on. The value is not case sensitive.

On, Yes, True, 1	Slamdown information is collected in a list which is sent as a number of SMS when the phone is turned on.
Off, No, False, 0	Slamdown information is sent as one message per slamdown call.

The default value is *No*.

Note: This configuration is not applicable for Notification by E-mail.

7.1.12.24 SlamdownMaxCallers

SlamdownMaxCallers is the maximum number of callers in one slamdown information message. The value must be a number:

0	Put as many callers as possible in each SMS.
1 - about 5	Put at most this many callers in each SMS, there will be fever callers if the SMS size would be exceeded.
about 6 and up	Un-necessarily large number with the same effect as 0.

The default value is *0*.

7.1.12.25 SlamdownMaxCallsPerCaller

SlamdownMaxCallsPerCaller limits the count of slamdown calls from one caller. This is useful primarily for minimizing the size of the caller information in the SMS by not counting more than e.g. 9 slamdown calls from the same caller.

0	Count all slamdown calls
9	Limit the count to one digit.
99	Limit the count to two digits
other numbers	Limit the count to other, perhaps less useful numbers.

The default value is *0*.

7.1.12.26 SlamdownMaxDigitsInNumber

SlamdownMaxDigitsInNumber can be set if long caller numbers should be truncated.

0	Show all digits in the caller number.
other numbers	Remove leading digits from the caller numbers, so at most this many digits remain.

The default value is 0.

7.1.12.27 SlamdownNumberOfRetries

If sending of the SMS type 0 or a slamdown information message fails, it is retried periodically. This parameter determines how many retries will be made before the slamdown information is discarded.

number	NTF tries to send the SMS type 0 and the slamdown information this many times, before giving up.

The default value is *5*.

7.1.12.28 SlamdownOldestFirst

This parameter determines the order in slamdown information messages , of the callers that made slamdown calls.

On, Yes, True, 1	The caller that called least recently is sent first.
Off, No, False, 0	The caller that called most recently is sent first.

The default value is *Yes*.

7.1.12.29 SlamdownTimeOfLastCall

In case there are several slamdown calls from the same caller, this parameter lets you select if the slamdown information shall show the time of the first or last call.

On, Yes, True, 1	The time of the callers last call is shown in the slamdown information.
Off, No, False, 0	The time of the callers first call is shown in the slamdown information.

The default value is *Yes*.

7.1.12.30 SlamdownRetryInterval

If sending of the SMS type 0 or a slamdown information message fails, it is retried periodically. This parameter determines the length of the period.

number	How many seconds shall pass before NTF tries sending a slamdown-related SMS again.

The default value is *300*.

7.1.12.31 SlamdownTruncatedNumberIndication

SlamdownMaxDigitsInNumber can be used with SlamdownTruncatedNum berIndication to replace the leftmost digit in a truncated number by another character, to clearly indicate that the number has been truncated.

empty	If a number has been truncated, there will be no special indication.
any string	The leftmost digit of truncated numbers is replaced with the first character of this string.

The default value is *.

7.1.12.32 SMESourceAddress

If this parameter is set, NTF sends it as originating address with SMS-requests, i.e. the SMS will be shown in the users mobile as if it came from this address. This address can often be configured in the SMSC instead, and then the value is omitted in notification.cfg.

It is possible to set a different source address for each type of SMS message, by setting one or more of the parameters below with names starting with `SourceAddress_`. The value of SMESourceAddress is the default for all those parameters.

<address>	This telephone number is used as source address, together with the type-of-number and numbering-plan-indicator set with the parameters SMESourceTON and SMESourceNPI.
<ton>,<npi>,<address>	This type-of-number, numbering-plan-indicator and telephone number is used as source address.
callers_number	The callers telephone number is used as the sourceaddress, together with the type-of-number and numbering-plan-indicator set with the parameters SMSSourceTON and SMSSourceNPI. Callers_number are only valid for SourceAddress_slamdown and SourceAddress_ivrtosms.

<ton>,<npi>,callers_number This type-of-number and numbering-plan-indicator and the callers telephone number is used as the source address. This is only available to SourceAddress_slamdown and SourceAddress_ivrtosms.

The default value is the empty string, meaning the null source adress is sent..

7.1.12.33 SMESourceNPI

The default numbering-plan-indicator used for source address parameters with only the telephone number set.

0..255 Numbering-plan-indicator is set to this value for all source addresses that do not explicitly contain a numbering-plan-indicator.

The default value is *0*.

7.1.12.34 SMESourceTON

The default type-of-number used for source address parameters with only the telephone number set.

0..255 Type-of-number is set to this value for all source addresses that do not explicitly contain a type-of-number.

The default value is *0*.

7.1.12.35 SmppBindType

This parameter tells how NTF binds to the SMSC. If the function Slamdown Information List is used, or if the outdial sequence checks for the phone to be on, the value must be “transceiver”. Otherwise it should be “transmitter”.

transmitter NTF binds to the SMSC as a transmitter.

transceiver NTF binds to the SMSC as a transceiver.

The default value is *transmitter*.

7.1.12.36 SmppErrorCodesIgnored

This parameter tells which error codes from the smsc that should be ignored. An ignored error code will cause the sms-notification to be treated as if it was ok. Enter the codes seperated by “,” or “ “. The codes can be in hex-format (e.g. 0xAF) or decimal-format.

The default value is the empty string, meaning that no error codes ignored.

7.1.12.37 SnmpAgentPort

This parameter defines the port where NTF listens to requests from the SNMP agent. If the value only specifies a port number, NTF listens to that port on the wildcard address. If the value includes an IP address, NTF listens to the port on that interface.

<local-ip-address>: <port number>	NTF listens to the port on the specified interface.
<port number>	NTF listens to the port on all interfaces.

The default value is *18001*.

7.1.12.38 SMSCPollInterval

This parameter is only used if NTF shall wait until the phone is on before outdial starts. To ensure that a connection is always up, NTF sends periodic poll messages to avoid timeout in network components and the SMSC. The default is often a suitable value, but other values may be needed, depending on how the SMSC (and network components between NTF and the SMSC) is set up.

any number	The number of seconds between polls.

The default value is *60*.

7.1.12.39 SmsPriority

This parameter defines the priority for all SMS messages sent by NTF. The values are different depending on the protocol used for SMSC communcation. If one NTF is attached to SMSCs with different protocols, a value that is valid for all protocols should be used. Values outside the valid range have the same effect as setting the priority to 0.

7.1.12.39.1 SMPP

0-3	Priority, 0 is the lowest priority.

The default value is *0*.

7.1.12.39.2 CIMD2

0	No priority information is sent to the SMSC.
1-9	Priority, 9 is the lowest priority.

The default value is *0*.

7.1.12.40 SmsQueueSize

Defines the max length for a queue towards one smsc. When the queue is full no more requests to that smsc can be accepted and calls are directed to the backup if loadbalancing is set or fails if loadbalancing is not set.

The default value is **20**.

7.1.12.41 SourceAddress_*

It is possible to have different source addresses for different kinds of SMS messages. If one of the source addresses in this section are not set, the value of SMESourceAddress is used. All of the settings below support a CoS specific source address that is used if present. The <cosname> is COSname in MUR. If for example setting SourceAddress_fax_cos1 is set and the user's cosname is "cos2" the setting of SourceAddress_fax will be tried, and if this is not set either, the SMESourceAddress is used. There is one exception: SourceAddress_IvrToSms can only be set per NTF instance.

The format of source addresses is described in Section 7.1.12.32 SMESourceAddress on page 119.

7.1.12.41.1 SourceAddress_CfuOnReminder

Source address for reminders that the user has left unconditional call forwarding on.

7.1.12.41.2 SourceAddress_email

Source address for email notifications.

7.1.12.41.3 SourceAddress_fax

Source address for faxmail notifications.

7.1.12.41.4 SourceAddress_Flash

Source address for flash notifications (SMS class 0).

7.1.12.41.5 SourceAddress_MailQuotaExceeded

Source address for mailbox full warnings.

7.1.12.41.6 SourceAddress_MwiOff

Source address for MWI off messages.

7.1.12.41.7 SourceAddress_MwiOn

Source address for MWI on messages.

7.1.12.41.8 SourceAddress_Slamdown

Source address for slamdown information SMS. Can have callers_numbers as value.

7.1.12.41.9 SourceAddress_IvrtToSms

Source address for sending messages to non subscribers. Can have callers_numbers as value.

7.1.12.41.10 SourceAddress_SmsType0

Source address for the SMS type 0 message that may be sent for some types of slamdown information or outdial notification.

7.1.12.41.11 SourceAddress_TemporaryGreetingOnReminder

Source address for reminders that the user has left temporary greeting on.

7.1.12.41.12 SourceAddress_UnreadMessageReminder

Source address for reminder SMS.

7.1.12.41.13 SourceAddress_UpdateSms

Source address for update SMS. Used both for updateAfterRetrieval and updateAfterTerminalChange.

7.1.12.41.14 SourceAddress_video

Source address for videomail notifications.

7.1.12.41.15 SourceAddress_voice

Source address for voicemail notifications.

7.1.12.41.16 SourceAddress_VoiceMailOffReminder

Source address for reminders that the user has left voice mail off.

7.1.12.41.17 SourceAddress_<system notification name>

Source address for a named system notification. If a system notification is defined (see Section 5.9 Configuring System Notifications on page 51) a special source address can be defined for that system notification by adding a source address parameter named after the systemnotification.

E.g. a special source address for a system notification called cutthroughpaging can be specified by defining the configuration parameter `SourceAddress_cutthroughpaging` in `notification.cfg`

7.1.12.42 UnreadMessageReminderType

This parameter enables or disables periodic sending of a reminder SMS to inactive users, and selects the type of message.

none	Sending of reminder SMS is disabled
sms	Sending of reminder SMS is enabled and the message is an ordinary SMS.
flash	Sending of reminder SMS is enabled and the message is a flash SMS.

The default value is *none*.

7.1.12.43 UnreadMessageReminderInterval

This parameter defines with what interval, the reminder SMS is sent. The value shall be an integer number of seconds and it is recommended to use at least several hours.

<60	Illegal value interpreted as 60 seconds.
60-	The number of seconds before the first reminder SMS and between the following.

The default value is *86400 (one day)*.

7.1.12.44 UnreadMessageReminderMaxTimes

The maximum number of reminder SMS to send to a user. The value should be a small, positive integer.

1-	The number of times to send a reminder SMS before giving up.

The default value is *3*.

7.1.12.45 UseAlternativeFlashDcs

The flash SMS function is implemented by sending an SMS Class 0. SMS class 0 is controlled by the GSM parameter data coding scheme, but this can be done in two ways. Normally the DCS used is 16 for messages with the GSM default character set and 24 for messages that require the UCS2 character set. With this parameter, you can force DCS to be 240.

Note: The alternative flash DCS can not be used with the UCS2 character set. Make sure the UCS2 conversion is not used in the character conversion file (Section 5.16 Configuring character conversion on page 74).

On, Yes, True, 1	Force DCS=240.
Off, No, False, 0	Set DCS to 16 or 24 depending on the character set.

The default value is *off*.

7.1.12.46 UseCallerInEventDescription

If this parameter is true, NTF will include the telephone number of the caller in the event description of notification events Section 6.1.2 E-mail Notification on page 84. This applies only to successful SMS notification of voice messages.

On, Yes, True, 1	Include the caller number in the event description.
Off, No, False, 0	Do not include the caller number in event descriptions.

The default value is *off*.

7.1.12.47 Validity_*

These parameters specify the validity period for different types of SMS. The validity period is the time period during which NTF shall consider a short message of this type, to be valid.

This time is given to the SMSC, so that if it is not able to send the SMS within the validity period, the SMS will be discarded.

0	The SMSC will decide validity period
1-300	The validity period is set to this many hours.
301-	Valid values that are probably useless.

The default value is taken from the expiration time in the users service profile.

7.1.12.47.1 Validity_CfuOnReminder

Validity period for reminders that the user has left unconditional call forwarding on.

7.1.12.47.2 Validity_Flash

Validity period for flash notifications (SMS class 0).

7.1.12.47.3 Validity_MailQuotaExceeded

Validity period for mailbox full warnings.

7.1.12.47.4 Validity_MwiOff

Validity period for MWI off messages.

7.1.12.47.5 Validity_MwiOn

Validity period for MWI on messages.

7.1.12.47.6 Validity_Slamdown

Validity period for slamdown information SMS.

7.1.12.47.7 Validity_SmsType0

Validity period for the SMS type 0 message that may be sent for some types of slamdown information or outdial notification.

7.1.12.47.8 Validity_TemporaryGreetingOnReminder

Validity period for reminders that the user has left temporary greeting on.

7.1.12.47.9 Validity_VoiceMailOffReminder

Validity period for reminders that the user has left voice mail off.

7.1.12.47.10 Validity_UnreadMessageReminder

Validity period for unread message reminders.

7.1.12.47.11 Validity_UpdateSms

Validity period for update SMS. Used both for updateAfterTerminalChange and updateAfterRetrieval.

7.1.12.47.12 Validity_<system notification name>

Validity period for a named system notification. If a system notification is defined (by configuring `systemnotification.cfg` and `*.cphr`) a special validity period can be defined for that system notification by adding a validity period parameter named after the systemnotification.

E.g. a special validity period for a system notification called cutthroughpaging can be specified by defining the configuration parameter **`Validity_cutthroughpaging`** to `notification.cfg`

7.1.12.48 XmpPollInterval

NTF sends empty XMP requests with this interval to allow servers a possibility to send replies.

5-180	An empty requests is sent with this many seconds interval.
other	Allowed but not useful.

The default value is *90*.

7.2 Tools

This appendix describes some tools delivered with NTF.

All tools can be found in the `bin` directory of the NTF installation (`<NTF_HOME>/bin`).

7.2.1 checkconfig

checkconfig removes comments, illegal parameters, obsolete parameters and parameters with default values from an NTF configuration file and prints the result. The output is the smallest file that gives the same configuration as the input file.

The tool can be used to

- See just the most interesting parameters in a configuration file, i.e. those that have been modified from the default values.
- Find illegal or obsolete parameters (using the option “ a”).
- Find parameters that should be set (parameters that should be set will have the value “NoDefault” in the output).
- Update the original configuration file with the output (using the option “ c”, for change).

To run `checkconfig`, NTFs java archive (`ntf.jar`) must be available either in the current directory or in a child or sibling directory called `bin`. One way to fulfill this is to run `checkconfig` from `<NTF_HOME>/cfg` where the configuration files are stored:

NOTE that checkconfig deletes comments from the configuration file.

These examples show the ways the tool can be used:

```
#cd <NTF_HOME>/cfg
#../bin/checkconfig -h
Usage: checkconfig [-anc] [<configfilename>]
    -a include parameters with default values and \
              illegal or obsolete parameters as comments.
    -c replace the config file with the cleaned version.
    -n run in noninteractive mode, not asking questions \
       if cleaning.
    configfilename defaults to "./notification.cfg".
```

Example 31 Help information for checkconfig

```
#cd <NTF_HOME>/cfg
#../bin/checkconfig
```

Example 32 Checking the default file ./notification.cfg

```
#cd <NTF_HOME>/cfg
#../bin/checkconfig notification.cfg.old
```

Example 33 Checking another file

```
#cd <NTF_HOME>/cfg
#../bin/checkconfig -a
```

Example 34 Displaying all parameters, even obsolete and illegal ones, and those that just have the default value.

```
#cd <NTF_HOME>/cfg
#../bin/checkconfig -c
```

Example 35 Updating the original file.

```
#cd <NTF_HOME>/cfg
#../bin/checkconfig -anc outdial.cfg
```

Example 36 Everything at once - checking and updating a non-default file, keeping all parameters and not asking for confirmation.

7.2.2 verify-outdial

Verify-outdial checks the configuration file used for outdial. It must be run in the cfg directory.

```
../bin/verify-outdial outdial-default.cfg
```

Example 37 Running verify-outdial.

The tool examines the configuration file and prints out potential problems it finds in the file. If there is an error in the file that makes it impossible to create the outdial state machine, that error is reported and nothing else.

If the configuration file describes a valid state machine, there may still be problems that might lead to failures during outdial handling. Those problems are classified as warnings, with three different levels; high, medium and low; depending on the impact and the probability that it was intentional.

The tool checks the following things:

7.2.2.1 Error

- The state machine can not start, because no transition is defined from the initial state when the start code is received. This means that there will be a runtime error at the first transition for all outdial attempts. The outdial handling sends a replacement sms if this happens.

7.2.2.2 High Level Warning

- Some state does not handle all reply codes. This will give a runtime failure if the unhandled code is received. The outdial handling sends a replacement sms if this happens.
- Transition after a sucessful call does not lead to the end state. This means that more actions might be done after a successful call, including calling again.
- A command list does a call that is not at the end of the list. This means that the following operations will be done regardless of the result of the call.
- The state machine may stop, i.e. there is a command list not ending with a call, where the next state is not the end state. Without a call, no code will be received that can move the state machine to a new state.
- A state will never be reached. There are no transitions from the initial state to a state. This means that the state will never be reached. This is not in itself dangerous but indicates a mistake in the configuration file.

7.2.2.3 Medium Level Warning

- The state machine contains transitions for unknown codes. This is not dangerous in itself, the transitions for those codes just never will be followed, however this can indicate other errors.
- An SMS template that does not exist in the template files is used. This means that the default sms content will be used when that sms command is executed.
- A state does not define any transitions on its own, it only uses the defaults. This might be correct if the default is appropriate, but it might be a sign of a mistake in the configuration file.
- There is a loop of states that can occur for other codes than the busy code. This means that there are some conditions where the outdial will try to call indefinitely. When the phone is busy, it is often desirable to loop and call again frequently. For other codes, it may indicate a mistake in the configuration file. In any case, the number of calls will be limited by the general time limit in the outdial configuration file.

7.2.2.4 Low Level Warning

- A list of commands in one transition sends more than one SMS.

- An SMS is sent in one transition and the next state is not final. This means that more than one SMS might be sent.
- A list of commands in one transition both sends an SMS and tries to call.
- There are more transitions than the start transition from the initial state. If the state machine never goes back to the initial state these transitions are meaningless. They are not dangerous in themselves but this can be a symptom of other problems.
- The start transition is defined for other states than the initial state. Since the start transition code is only used in the initial state these transitions will never be followed. This can be a symptom of other problems.
- The initial state is not number zero. The convention is to have the initial state as zero. This might be a mistake in the configuration file.
- There are transitions back to the initial state. Normally the initial state is only used at the start. This might be a mistake in the configuration file.
- There is a loop of states, including the busy code. This means that if the phone is busy, the state machine can keep on trying indefinitely. This is often wanted, but not always.

7.2.3 smppbind

Experience shows that obtaining the correct user name/password settings for the SMSC is surprisingly difficult. To help in trouble-shooting a new NTF installation where SMS notifications do not work, the script **`smppbind`** can be used. It can be run on any host where perl is available (as `/usr/bin/perl`). Smppbind can be used to:

- Isolate notification troubles to the SMSC connection.
- Quickly try different user names and passwords.
- Localize network problems by testing from different hosts and networks.
- Get detailed diagnostic messages.

You can run smppbind in two modes; either you specify all parameters on the command line or you specify all parameters in one or more lines in a file. The latter mode is useful if you want to do many bind attempts, e.g. to try alternative spellings for system_id and password; one bind attempt is done for each line.

```
Usage
./smppbind [-v] host port system_id password system_type
  or
./smppbind [-v] -f filename
      -v verbose gives more messages
      system_type is optional
      system_id is the SMPP terminology for user name
      -f filename identifies a file with lines, where each
      line contains host port system_id password
      system_type, and a bind will be made for each line.
```

```
#./smppbind smpphost 5016 smsuser smspass
Failed SMPP Bind to smpphost:5016 as smsuser(smspass) \
status code 15 (Invalid system id (i.e. user name))
```

Example 38 Running smppbind with a bad user id.

7.2.4 addinstance

Addinstance is used in a HA-system to create NTF instances.

addinstance -h gives help

7.2.5 getconfig

Getconfig displays the parameter value used by NTF, whether it is a default value or in the configuration file.

getconfig -h gives help

7.2.6 interactivebind

Interactivebind is a simple text-based program that lets you try binding to an SMSC interactively.

Start interactivebind and then use the h command for help.

7.2.7 ntf_mcr_reg.sh

Ntf_mcr_reg registers NTF in MCR.

ntf_mcr_reg -h gives help

7.2.8 ntf_mcr_unreg.sh

Ntf_mcr_unreg unregisters NTF from MCR.

ntf_mcr_unreg -h gives help

7.2.9 ntf_mur_reg.sh

Ntf_mur_reg registers NTF users in MUR.

ntf_mur_reg -h gives help

7.2.10 ntf_mur_unreg.sh

Ntf_mur_unreg unregisters NTF from MUR.

ntf_mur_unreg -h gives help

7.2.11 phr2cphr.pl

Converts phrase files from old format (.phr) to new format (.cphr)

phr2cphr.pl -h gives help

7.2.12 rminstance

Rminstance can be used in HA-systems to remove NTF instances.

rminstance -h gives help

7.2.13 stopntf

Stopntf stops the NTF traffic processes. Normally, it will be started again by the NTF watchdog.

7.2.14 upgradeinstance

Upgradeinstance can be used in HA-systems to upgrade the configuration of NTF instances after the software has been upgraded.

upgradeinstance -h gives help

7.3 Copyright Notices

7.3.1 The Hypersonic SQL Group

Copyright (c) 1995-2000 by the Hypersonic SQL Group.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the Hypersonic SQL Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE HYPERSONIC SQL GROUP, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Hypersonic SQL Group.

7.3.2 The HSQL Development Group

Copyright (c) 2001-2004, The HSQL Development Group
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the HSQL Development Group nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HSQL DEVELOPMENT GROUP, HSQLDB.ORG, OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Reference List

[1] *Messaging-over-IP Technical Overview*, 5/1551-HDB 101 02 Uen

[2] *Glossary*, 1/0033-1/HDB 101 02 Uen

[3] *Installation Guide NTF*, 1/1531-CRH 109 127 Uen

[4] *MIB Management User Guide*, 1/1553-CRH 109 083 Uen