ERICSSON

ENM Basic Network Surveillance Interface Protocol Specification

Interwork Description

1/15519-CNA 403 3034 Uen K

Contents

1 ENM Basic Network Surveillance Interface Protocol Specification

This document covers all the current versions of BNSI, BNSI versions 1,2 (BNSIv2) and version 3 (BNSIv3). It describes the complete BNSI protocol.

Standard Compliance

The BNSI protocols are in line with the standard alarm management definitions. The degree of compliance is also dependent on the implementations of agents and managers.

- Telecom. Mgmt Network, Generic Network Information Model ITU-T Recommendation M.3100
- Data Comm. Networks, Definition of Management Information ITU-T Recommendation X.721 (ISO/IEC 10165-2)
- Data Communication Networks, Alarm Reporting Function ITU-T Recommendation X.733 (ISO/IEC 10164-4)
- Data Communication Networks, Security Alarm Reporting Function ITU-T Recommendation X.736 (ISO/IEC 10164-7)
- Fault Management of the Base Station System ETSI EN 301 251 (GSM 12.11)
- Q3 Interface for Alarm Surveillance ITU-T Recommendation Q.82
- Transmission Control Protocol/Internet Protocol TCP/IP, RFC793/RFC791
- Information Processing, 8 bit single-byte coded, Latin alphabet no 1 ISO 8859-1:1987

Target Groups

This document is intended for users and developers of systems that are using the Basic Network Surveillance Interface (BNSI). BNSI is a protocol primary dedicated to forward alarm information from various network sources to a fault management system.

Requirements

It is assumed that the user of this document:

- Has knowledge of fault management.

- Has basic knowledge of telecommunications and data communication.

Typographical Conventions

The typographic conventions for all Customer Product Information (CPI) in ENM are reported in *[2]*.

2 Introduction to BNSI Protocol Specification

The purpose of the BNSI is to simplify the integration of new element management or network element systems with a centralized network fault management system. The protocol is also used in other alarm gathering situations.

To achieve an interface which is easy to implement, the BNSI has a text-based format for the information that passes through the connection. BNSI is also, in its basic form, a unidirectional interface.

The forwarded alarm records are coded as a sequence of lines with readable text. The transport mechanism (the interface adapters) may vary for different implementations of the BNSI.

The BNSI is intended to be implemented in a layered management system environment. This means that records sent over the BNSI should originate from connected systems, which include basic alarm handling capabilities.

The protocol is data-driven and no polling mechanism is needed. No special development or test equipment is necessary to be used when implementing a new BNSI agent. The BNSI is designed according to several international telecommunication standards (ITU-T, ISO).

When the fault management system connects to the alarm source, it usually starts a synchronization. This means that the alarm source sends all currently active alarms as synchronization alarm records to the fault management system. The fault management operator gets a consistent view of all alarms for the connected systems.

A manager is the receiving and controlling side of a BNSI interface, and an agent is the unit responsible for forwarding alarm information from the sources to the fault management system. There are no upgrade dependencies between the managers and the agents. The protocol is backward compatible in its different versions.

Notes on Terminology

In this document, a fault is something that occurs in the network or in the network equipment; for example, a power supply failure. A fault can also be a malfunction of a process; for example, a software bug or a miscalculation.

The fault is detected as an error by a supervision device, which emits a notification. The notification is of type alarm or type event. If it is possible for the device to detect the absence of an error - this is reported as a notification of type alarm clear. Between the alarm and the alarm clear, the alarm is called an active alarm.

As a response to the alarm or event, the manager may send an action. The action can be to find more information about the fault, or to correct a service affected by the fault (for example to re-route traffic or switch over to back-up equipment).

For more terminology, see *Object Identifier Attribute Types* on page 42.

2.1 Environment

The main purpose of BNSI is to support the functionality of a network fault management system.

The function of a network fault management system is to gather and present alarms from different types of network elements and other types of network objects, either directly or through their element managers, as shown in the following figure.

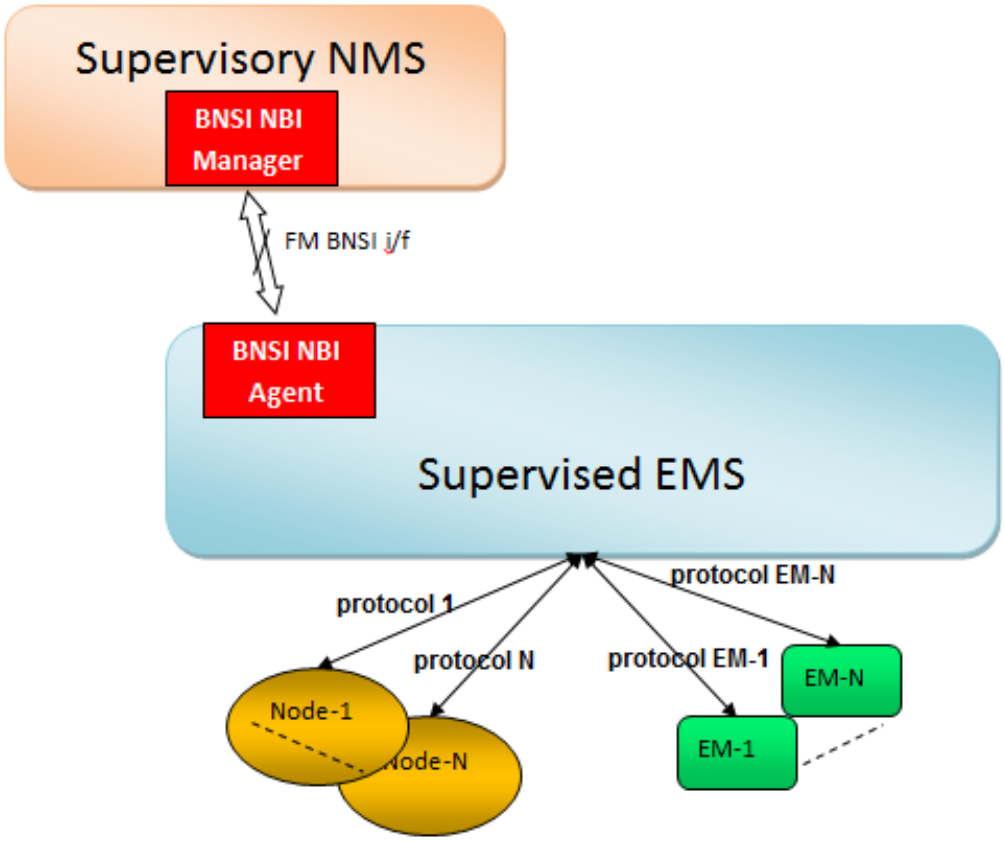

Figure 1 Example of Alarm Flow in a BNSI Environment

The BNSI was original intended to be used in a layered management solution. This means that the records sent over the BNSI originate from a connected management system which includes basic alarm handling capabilities. This system can, for example, be an element manager or network element manager. An agent system is normally used as a software adaptation in networks using Network Elements (NE) that cannot forward information

directly to the management system. In this case the supervised system is an Element Manager (EM).

In addition, BNSI can be used between layers of Network Management (NM) systems. The protocol can be used as an interface between, and within, the TMN layers NE, EM and NM. The protocol can also be used between network management systems or components in a network management system.

3 Communication Session Control

The BNSI agent receives alarm and event information from the network, translates it into the BNSI format, and forwards it to the management system. The management system receives information from the agent and handles it as it comes. The management system can, in the simplest case, only start and stop the BNSI agent, and the information flows from the BNSI agent to the management system. Therefore, in this case, the interface is unidirectional.

In a network supervised by a BNSI management system, the agent systems are responsible for handling communication problems between the agent system and its network elements. If the communication between an agent system and a network element is lost and regained, the agent system needs to synchronize with the network element. When this is done, the agent system updates the management system with the current alarm situation.

3.1 Normal Communication Scenario

The command option for alarm synchronization is used to instruct the supervised system to send synchronization alarms prior to spontaneous alarms.

In the example, the command option for the heartbeat interval is set to 30 which instructs the supervised system to send heartbeat messages with an interval of 30 seconds if no alarms are sent. If no record or heartbeat message has been sent during the past 30 seconds it sends a heartbeat message.

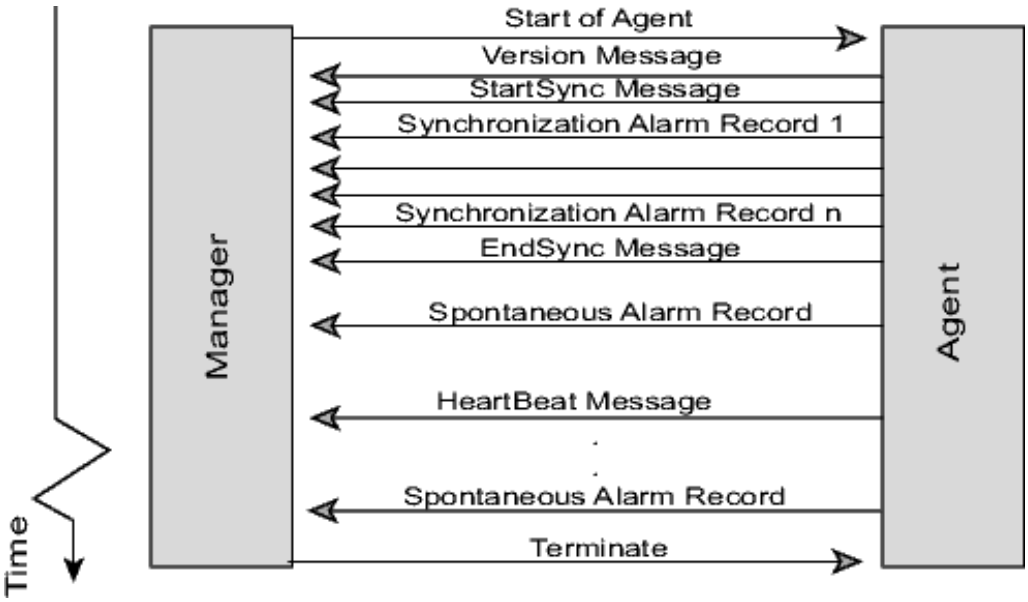

Figure 2 Basic Communication Scenario

3.2 Communication Disturbance Scenario

The interface mechanism chosen is characterized by a high degree of robustness. The basic steps used in most error cases are to start a new alarm sending session and terminate the previous one if it is alive in any sense.

A short disturbance in the physical communication does not have to be handled at all. The chosen transport protocol must be able to buffer the text-coded alarms on the sending side, and forward them when the link is working. This is valid when the disturbance is shorter than the heartbeat timeout monitored by the management system. This timeout is normally about twice as long as the heartbeat interval specified in the BNSI.

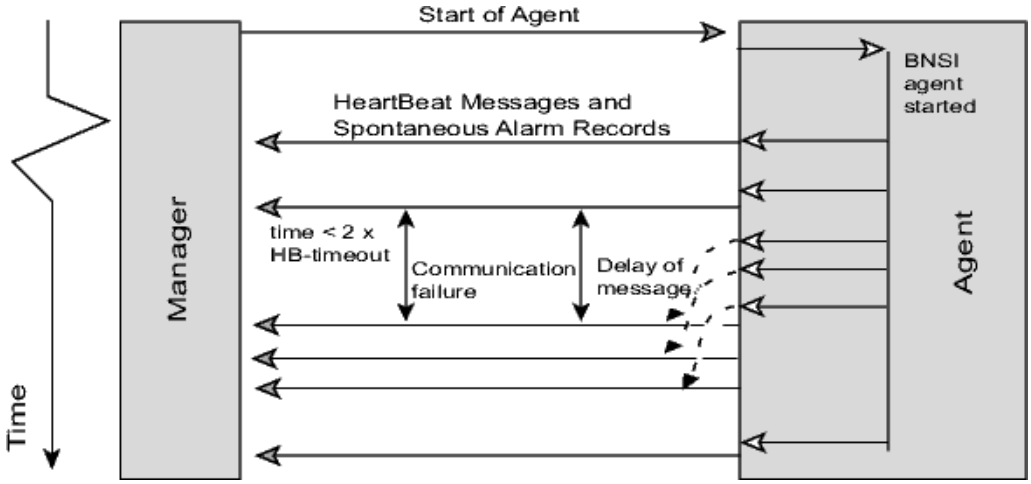

Figure 3 Short Communication Disturbance

3.3 Communication Failure Scenario

The management system tries to start a new communication session, and continues to do so until it succeeds. The communication software can terminate the previous session in the supervised system. If not, it means that the management system is not listening any more, since the supervised system has not been able to send any records for more than twice the heartbeat interval.

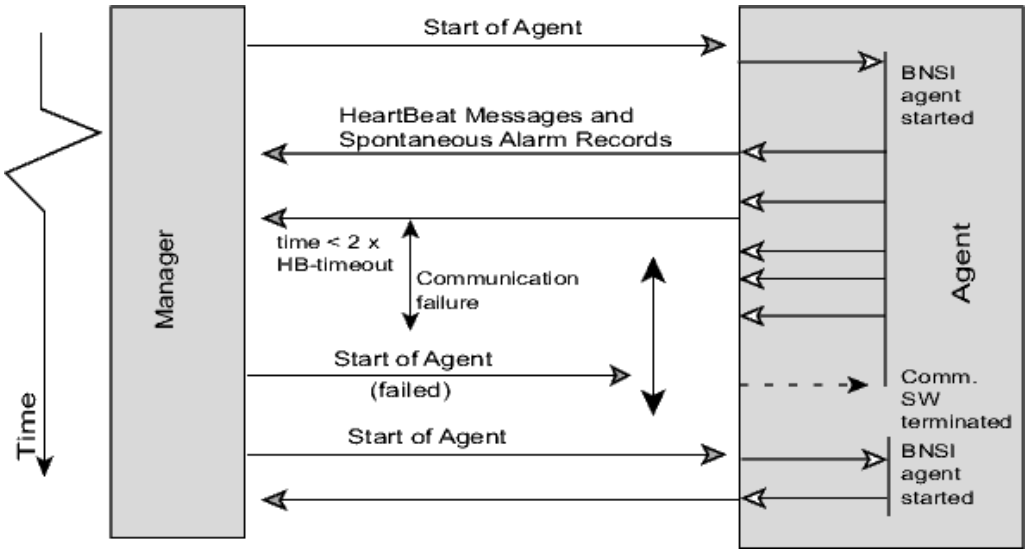

Figure 4 Long Communication Disturbance

4 Configuration

There are few items that need to be considered when configuring the manager and the agent side; for example, the transport mechanism must be decided and the user account in the agent system and ff or not the default translation map must be used.

Characteristics

BNSI is based on and utilizes TCP/IP for fast and reliable network communication.

Both capacity and performance values are affected by the hardware used and the software configuration.

Limitations

The interface is restricted to fault management, meaning that it is restricted to alarm surveillance and actions related to fault management.

5 Interface Adapter

The communication in BNSI is based on TCP/IP, and the interface adapter handles the transport protocol.

The basic idea is that the management system where the BNSI manager is running initiates the alarm transfer by starting a process in the supervised system (the BNSI agent). The manager system silently receives the records that the supervised system sends. The records are transferred using the BNSI agent output to the manager system.

The functionality of the protocol is grouped into three major areas: Alarm Management, Action Management and Interface Adapter. See Major Functional Parts figure.

Note: The selection of interface adapter mechanism must be agreed and selected before the communication is started between the management system and the supervised systems. This is a task for the system integration projects.

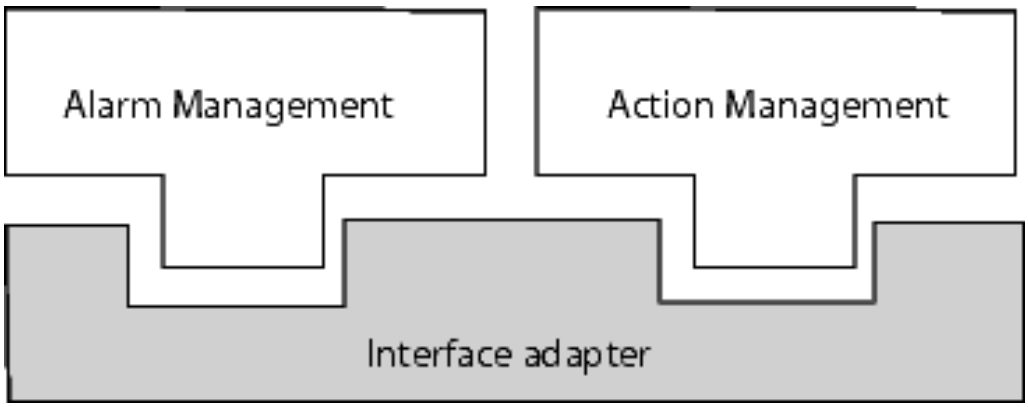

Figure 5 Major Functional Parts

5.1 Interface Adapter Definition

The BNSI manager unit implements the receiving and controlling side of the BNSI. The received text-coded alarm records are analysed and converted to an internal representation used by the management system. The basic interface mechanism is implemented in the interface adapters. The purpose of these adapters is to make it easy to adapt the system to different transport mechanisms.

The requirements for the interface adapter mechanisms are that these are reliable, support full-duplex, and supports a byte-stream. Whether they are connection-oriented or connection-less is not important. TCP/IP fulfils these requirements. When starting a remote process by means of an interface

adapter, there is one bi-directional TCP/IP connections established. One for the agent input and one for the agent output. There is also one for the transferring of signals.

A simple sketch of a BNSI-based fault management system in its environment is shown in figure *Interface Adapter Role*.

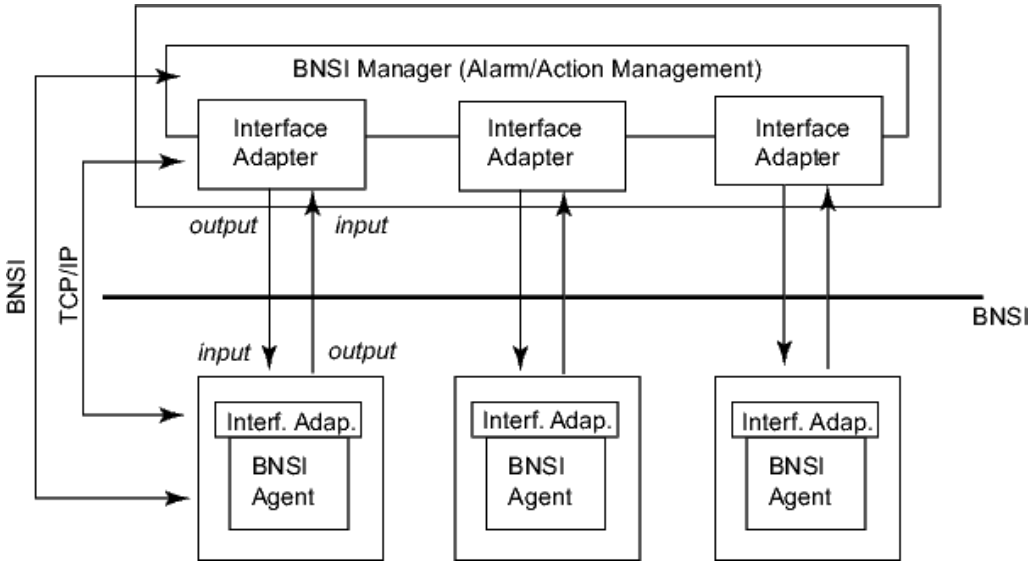

Figure 6 Interface Adapter Role

The interface between the interface adapter and the control part of the manager/agent consists of a bi-directional character stream and these two 'commands':

- Start of surveillance: Initiate the remote execution of the agent
- Termination of surveillance: Stop the execution of the agent

Commands to suspend and resume the alarm flow are not supported and not needed since the TCP/IP network communication buffers and suspends further alarms if the manager system temporary cannot cope with the alarm flow.

If possible, any disturbance on the communication link is handled by the interface adapter without notifying the controlling part of the manager/agent. This, for example, means that the interface adapter uses hand-shaking mechanisms and acknowledgement of records or both (that is, the interface adapter on the manager side sends a message back to the interface adapter on the agent side for each record it has received correctly).

The current publicly defined interface adapters are listed in table *Transport Mechanisms in BNSI*. In addition to these interface adapters, there are other proprietary or special-solution adapters used.

Table 1 Transport Mechanisms in BNSI

Interface Adapter	Surveillance Command: Start	Surveillance Command: Terminate	Comments
REXEC	Described in Request for Start of Surveillance section	Described in Request for Termination of Surveillance section	See UNIX documentation. Advantages: Reserved TCP port. Limitations: Cleartext password.
RSH	Described in Request for Start of Surveillance section	Described in Request for Termination of Surveillance section	See UNIX documentation. Also called REMSH and RCMD. Advantages: Reserved TCP port, trusted user support. Limitations: No signals passed.
SSH	Described in Request for Start of Surveillance section	Described in Request for Termination of Surveillance section	See UNIX documentation. Advantages: Reserved TCP Port, trusted user support, encrypted information, more commands can be sent separately on an opened session Integration agent can be supported; see Implementation Specification section.

Netware

Communication is implemented with the network functions for the TCP/IP protocol.

The transport layer is implemented differently depending on what is available on the remote machine. An example of the interoperation of BNSI with other parts of the protocol suite is shown in figure *Example of the BNSI Protocol Suite*.

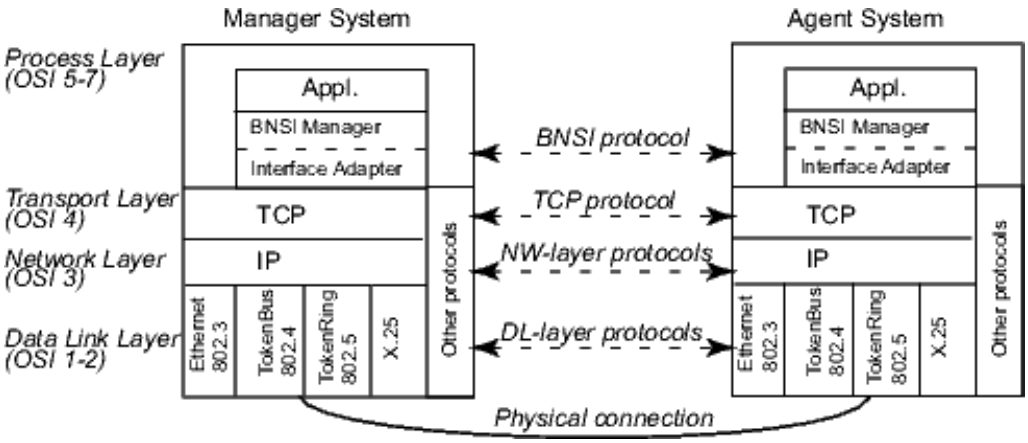

Figure 7 Example of the BNSI Protocol Suite

Note: In BNSI the supervised systems is accessible from the management system via a TCP/IP network. To exemplify it, the manager communicates with one supervised systems by a WAN connection with X.25, and one with a LAN connection by an Ethernet connection using TCP/IP directly.

5.2 Request for Start of Surveillance

When any of the interface adapters `rsh/remsh/ssh` or `rexec` are used, the alarm forwarding process (the BNSI agent) should be implemented as a program which receives start-up parameters in the argument list. The resulting text-coded alarms are sent to the output channel of the program. The start command instructs the agent system to start sending alarms to the manager system. It must be named SendAlarms and have the following parameters:

```
SendAlarms <name> [ -sync | -xsync ] \      [ -hbint <sec> ] [ -ver <ver> ] \   →
    [ -toor ] [ -filter ] [ -autoack ]      [ -ml ]
```

Note: In a character case sensitive operating system the capital-S and capital-A in the word SendAlarms are significant.

Argument	Reference
<name>(1)	See AgentName parameter.
-sync	See Synchronization parameter.
-xsync	See Extended Synchronization parameter.
-hbint <sec>	See Heartbeat parameter. The heartbeat timeout is given in the <sec> time specification.
-ver <ver>	See Version parameter. The <ver> is a number for the highest supported version by the manager that is BNSIv3.
-toor	See TypeOOR parameter.
-filter	See Filter parameter.
-autoack	See AutoAck parameter.
-ml	See MultipleLine parameter.

(1) The <name> argument is mandatory and must come as the first argument, the other arguments can be in any order and additional parameters can be added. If an agent does not understand them, they will be ignored (except for <name> and —hbint that the agent must be able to support.

When the SendAlarms process starts, it displays the highest supported version info on the console and the process waits on the console for alarms and action requests. So the user cannot press the Enter key more than three times.For the management system to use the rexec or rsh service, there must be a user account in the supervised system that the management system can use. The user name and password for this account have to be stored in a configuration file in the management system. A special account for this

purpose should be provided in the supervised system. Preferably, this account should be set up so that only the BNSI agent command can be executed.

5.2.1 Configuration Parameters for BNSI Agent

The BNSI agent is a process that is started (or activated) by the BNSI manager, and the agent receives configuration parameters.

Parameter	Description
AgentName	The name of the supervised system (agent system) used in the management system. This parameter is mandatory. It is used as a base for the attribute ObjectOfReference for the Network Elements. See Object Identification section.
Synchronization	If this parameter is present, the supervised system sends all currently active alarms to the management system before it sends new alarms.
Extended Synchronization	If this parameter is present, the supervised system sends an extended synchronization list.
Heartbeat	If this parameter is present, and if no alarms are sent within the time specification, the supervised system sends a heartbeat message. The heartbeat timeout is specified in seconds with a typical value range of 30 to 300 seconds.
Version	The highest supported BNSI version by the manager is given as a parameter to the agent. The agent responds with the version number it supports, and if no response is given from the agent, the default version is used. The agent may not respond with a higher value than what the manager requested. If this argument is missing, the agent may not respond with any version number, and the default version is used. In version 1, it is not possible to send a sync_start followed by a sync_end for a network element. When a network element is deleted, to notify the manager that the alarm list has been cleared, the agent exits. In version 1, it is also not possible to do actions towards the agent. So restart of the agent process is necessary to synchronize.
TypeOOR	If this parameter is present, the agent uses, if possible, the syntax of ObjectOfReference and NetworkOfReference as described in Object Identification section.
Filter	If this parameter is present, and the selected version supports it, the predefined filter in the agent is used to suppress alarms. The absence of this flag does not mean that no filter may be used by the agent. This flag says nothing about how the filter is defined.
AutoAck	If this parameter is present, and the selected version supports it, the predefined auto-acknowledge filter in the agent is used to acknowledge all forwarded alarms in the agent system.
MultipleLine	If this parameter is present, the manager tells the agent that it is capable of handling several rows of text for a single attribute. This is only applicable to text-value attribute. Also note that the so-called free text attribute may be multiple rows even if this parameter is not present.

5.3 Request for Termination of Surveillance

Signals are not a mandatory part of a BNSI protocol, but the communication session with a BNSI agent is usually terminated by the management system with a Terminate Signal, signal SIGTERM (15).

The agent process then terminates its execution as quickly as possible. The agent process also terminates if it discovers that the communication link to the manager is lost (if it is able to detect this). In some systems this could be detected by the signal SIGHUP (1) or signal SIGPIPE (13) or EOF character found on input channel. In this case, the agent must not try to write any exit-message to the manager.

Table 2 Signals (from the Manager) to the Agent

Signals	Action/Behavior
1, 2, 3, 13, 15	Terminate the agent.
Others	Not defined: Ignored, terminate or default behavior.

6 Alarm Management

This section gives a formal syntax description of the messages sent over the BNSI in the alarm sending session.

The simplest form of BNSI is basically unidirectional; the BNSI manager system is the receiver and the connected supervised systems are the senders. The manager system initiates the communication when it is ready to receive alarm information and handles the received information as it comes.

In the basic case (if only the basic parts of the alarm management functionality are used and none of the action management functionality is used), three services must be provided by the supervised system:

- Send Alarm, Alarm Clearing and Event records (until the session is ended).
- Send all currently active alarms (to synchronize the management system alarm list).
- Send Heartbeat messages (to know that the supervised system and the communication channel are working).

In addition to this basic functionality there are several optional functions in BNSI.

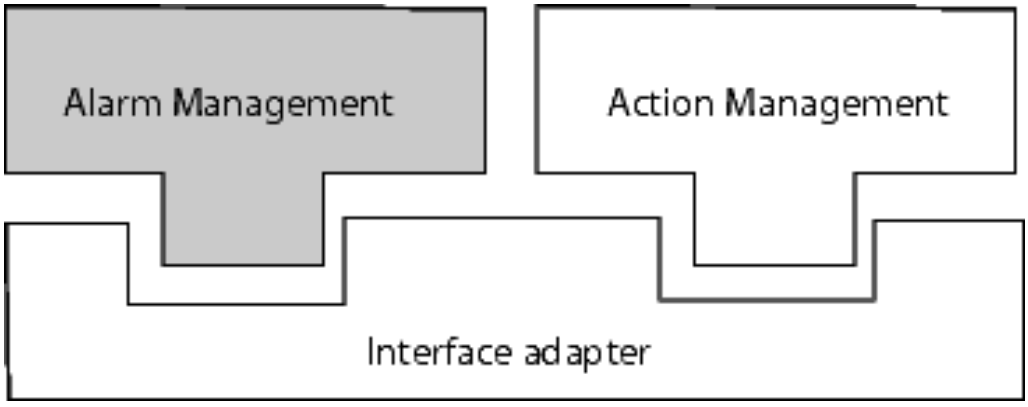

Figure 8 Functional Parts — Alarm Management

The following types of messages can be sent from the supervised system:

- Alarm, AlarmClear and Event records
- HeartBeat message
- SyncStart, SyncAlarm, SyncEvent and SyncEnd records
- Version message

- Comment, Error and Termination messages

6.1 Alarm Management Syntax

This section describes the syntax of the BNSI alarm event flow expressed in the BNF notification.

Example 1 BNSI Syntax in BNF Notification

```
eventflow ::=    [ comment | err ] [ neg ] [ { hb | err } ]
[ sync ] [ { record | hb | err | comment } ] [ exit ] eof
sync ::=    startsync [ { syncrecord | hb | err | comment } ]
endsync
record ::=    startline [ { comment } ] alarm-rec {
mandatory-attr } [ { optional-attr } ] [ { additional-attr }]
  [ free-text ] endline
startsync ::=    startline [ { comment } ] startsync-rec
oor-attr endline
endsync ::=    startline [ { comment } ] endsync-rec
oor-attr endline
syncrecord ::=    startline [ { comment } ] sync-attr{
mandatory-attr } [ { optional-attr } ][ { additional-attr } ]
[ free-text ] endline
mandatory-attr ::=    ( oor-attr | time-attr | et-attr |
ps-attr | pc-attr )
optional-attr ::=    ( sp-attr | an-attr | ni-attr | sc-attr |
  ocor-attr | boi-attr | bs-attr | pra-attr
| ti-attr | ack-attr )
alarm-rec ::=    &8220;-RecordType=&8221; ( 1|2|3|4|6 ) eol
sync-rec ::=    &8220;-RecordType=5&8221; eol
startsync-rec ::=    &8220;-RecordType=7&8221; eol
endsync-rec ::=    &8220;-RecordType=8&8221; eol
oor-attr ::=    &8220;-ObjectOfReference=&8221; chars eol
nor-attr ::=    "-NetworkOfReference=" chars eol
time-attr ::=    &8220;-EventTime=&8221; digits eol
et-attr ::=    &8220;-EventType=&8221; digits eol
ps-attr ::=    &8220;-PerceivedSeverity=&8221; ( 0|1|2|3|4|5 ) eol
pc-attr ::=    &8220;-ProbableCause=&8221; digits eol
sp-attr ::=    &8220;-SpecificProblem=&8221; digits eol
an-attr ::=    &8220;-AlarmNumber=&8221; digits eol
ni-attr ::=    &8220;-NotificationIdentifier=&8221; digits eol
sc-attr ::=    &8220;-StateChange=&8221; digits eol
ocor-attr ::=    &8220;-ObjectClassOfReference=&8221; chars eol
boi-attr ::=    &8220;-BackupObjectInstance=&8221; chars eol
bs-attr ::=    &8220;-BackedupStatus=&8221; ( 0|1 ) eol
pra-attr ::=    &8220;-ProposedRepairAction=&8221; digits eol
ti-attr ::=    &8220;-TrendIndication=&8221; ( 0|1|2 ) eol
ack-attr ::=    &8220;-Acknowledge=&8221; ( 0|1|2 ) eol
additional-attr ::=    &8220;-&8221; chars &8220;=&8221; chars eol
free-text ::=    { chars eol }
neg ::=    &8220;#Version=&8221; (1|2|3) eol
hb ::=    &8220;#HB&8221; eol
comment ::=    &8220;#&8221; &8220; &8221; chars eol
err ::=    &8220;#Error=&8221; chars eol
exit ::=    "#Exit=" digits eol
startline ::=    &8220;%a&8221; eol
endline ::=    &8220;%A&8221; eol
eol ::=    &8216;lf&8217; | &8216;cr&8217; &8216;lf&8217; | &8216;cr&8217;
eof ::=    ( &8216;termination-signal&8217; | &8216;end-of-file&8217; )
digits ::=    { 0|1|2|3|4|5|6|7|8|9 }
chars ::=    Any sequence of 8-bit visual character.
```

6.2 Alarm and Event Forwarding

The supervised system sends alarms for all its supervised equipment.

The equipment that has to be supervised is not explicitly defined. Some equipment might be 'hidden' by an internal configuration in the supervised

system. That is, alarms associated with some particular objects can be discarded from the forwarding process. See *Alarm and Event Pre-Filtering* on page 19.

The alarm information is transferred as records of the following types:

- Alarm records
- Alarm Clearing records
- Event records

Alarm and Event records basically transfer the same kind of information; the difference is that an Alarm record can be cleared. In the remainder of this document the term 'record' is used as the name for these different types of messages.

6.3 Alarm Synchronization

When an alarm sending session is started, a complete list of the currently active alarms is expected to be sent over the interface.

It is normally necessary to recover the alarm state of the connected network elements after a communication failure between the supervised system and the management system, and after restarts of the supervised system or the management system. When the management system requests an alarm summary to synchronize its internal alarm list, the ongoing alarm sending process in the supervised system is terminated, and a new one is started. The result is that the management system is supplied with a fresh alarm list followed by a continuous sequence of new alarms.

The expected behavior is that all currently active alarms are transferred as synchronization alarms between a start sync record and an end sync record. All spontaneously generated alarms and alarm clearings are buffered and sent after the synchronization phase is finished.

6.3.1 Extended Synchronization

The extended synchronization function is a variant of the alarm synchronization function.

As with the alarm synchronization function, this is requested by the manager at the start of a session. The manager can decide whether they want to use alarm synchronization or extended synchronization.

The expected behavior is that all active alarms (acknowledged or not), together with all events and cleared alarms which are not acknowledged, are sent in the extended synchronization.

The difference between alarm synchronization and extended synchronization is detailed in the *Extended Synchronization* table where (X) indicates that the

alarm or event is included in the synchronization list, and (-) indicates that it is not.

Table 3 Extended Synchronization

Alarm/Event type	Synchronization	Extended Synchronization
Active Alarms unacknowledged	X	X
Active Alarms acknowledged	X	X
Cleared Alarms unacknowledged	-	X
Events	-	X

6.4 Heartbeat Messaging

The system sends heartbeat messages at regular intervals.

Besides sending the different types of records, the supervised system must also be able to send heartbeat messages at regular intervals (if the manager requests it). These messages, or rather the absence of them, give the management system a way to determine when the information flow from the supervised system is interrupted.

If no alarms are sent to the agent system within a specific time (the heartbeat interval), the BNSI agent sends a heartbeat message to the management system to confirm that it is still alive.

6.5 Alarm and Event Pre-Filtering

The optioanl event or alarm filter enables event filtering so that only events that fulfill certain criteria are forwarded to the management system.

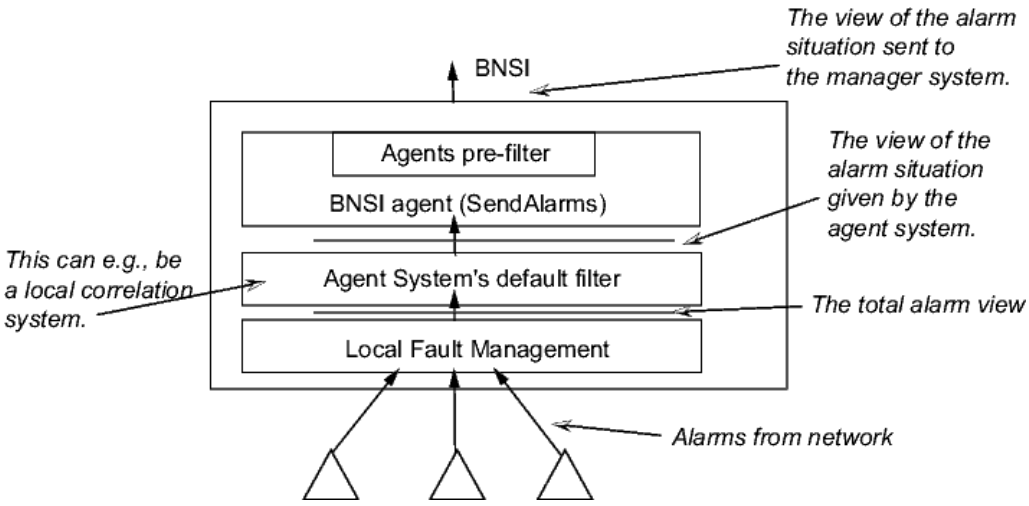

Figure 9 Filters in an Agent System

The setting of the filter is controlled, in the following order, by:

- The BNSI agent process.
- A configurable default setting - that is, a local configuration of the agent system.

If no filter is requested by the manager system and no default filter exists, then all events or alarms can be forwarded by the agent.

6.6 Start and Flow of Alarms Surveillance

The start and termination is controlled by the Interface Adapter.

Heartbeat supervision is achieved by having the alarm sending process in the supervised system regularly send dummy messages if there are no alarms to report.

The data flow is separated into lines, and each line is separated by a line feed character <LF>. An additional carriage return <CR> may be added in connection to the line feed (that is, support of both UNIX and PC style).

Empty lines may be inserted between records, messages or in the free text field of a record. Empty lines between records and messages are ignored by the manager.

Note: The manager does not treat empty lines as 'alive notifications'.

The sequence of different message types sent over the BNSI is shown in the following figure.

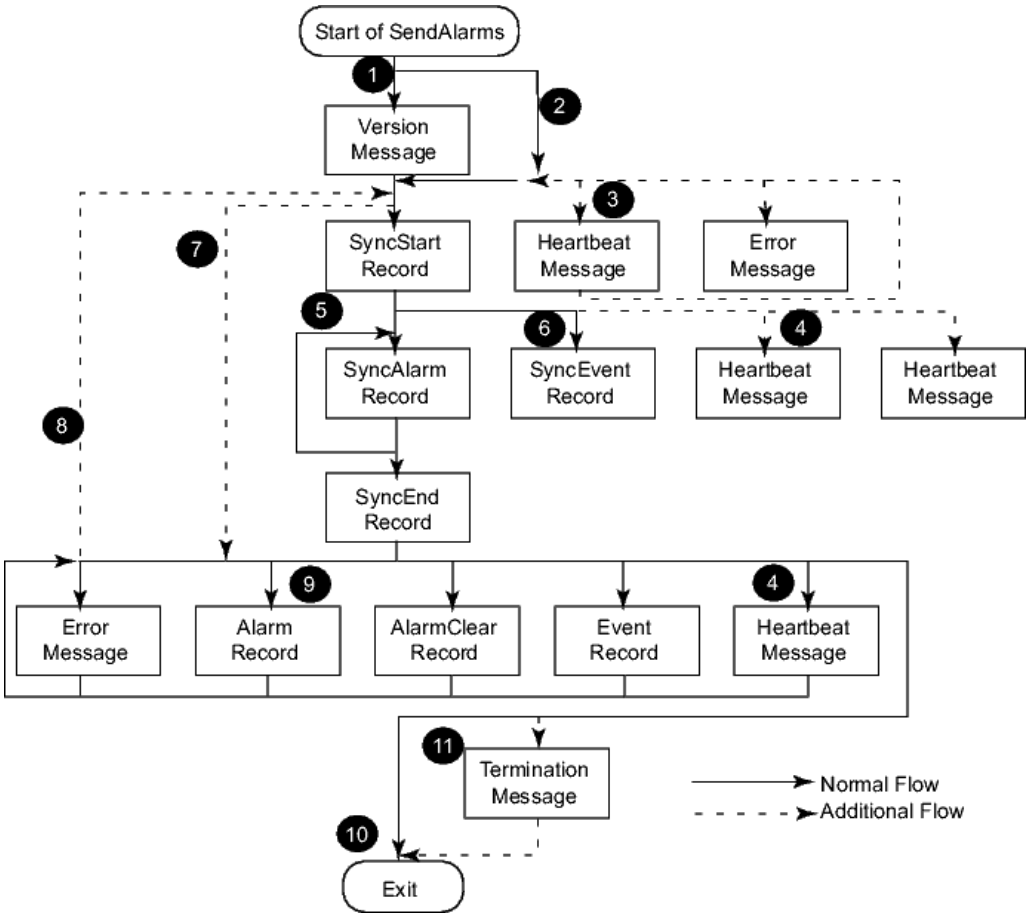

Figure 10 BNSI Upstream Message Flow (from Agent to Manager)

1. The manager starts the agent. The manager supplies the agent with start-up parameters. Some parameters are optional.

2. If requested by the manager, a message stating the version supported by the agent may be sent. This is optional.

3. The synchronization sequence may be delayed due to a heavy load on the supervised system. In this case, heartbeat messages have to be sent according to the heartbeat interval specified at the start of communication.

4. Heartbeat messages must always be sent according to the heartbeat interval when no other messages are sent. If possible, no heartbeat messages must be sent during the synchronization phase.

5. When an alarm sending session is started with the synchronization parameter or the extended-synchronization parameter, a complete list of currently active alarms is expected to be sent over the interface. All currently active alarms are transferred as synchronization alarms between a start sync record and an end sync record. All spontaneously generated alarms and alarm clearings are buffered and sent after the synchronization phase is finished. It is very important that no alarms are

lost in this process. It is also important, and a requirement, that alarms are sent only once during a synchronization phase.

6. If extended synchronization is used, the unacknowledged events must also be sent during the synchronization phase.
7. If the manager does not request a synchronization, or if the agent does not support the synchronization function.
8. The agent may perform a ‘spontaneous synchronization’.
9. Throughout the normal supervision period, the agent spontaneously sends alarms, alarm clears, error messages and event messages to the manager.
10. When there is a serious error in the supervised system making it unable to report alarms, it may close down the link.
11. It is possible to send a termination message to notify the manager of the reason for the termination.

6.7 Internal Faults

If the supervised system encounters a serious problem, the management system must be informed.

If supervised system is unable to forward alarms over the interface, the management system is informed by the termination the session. The management system recognizes that the communication channel is closed, and acts as it does upon communication failure: it tries to restart the alarm sending session.

As long as the supervised system cannot send alarms, these attempts to restart must be rejected.

6.8 Version Negotiation

When the alarm communication session is started, the manager can specify a desired BNSI version.

The manager specifies the BNSI version as an argument to the agent program. The agent side can then respond with the, by the agent, highest supported version BNSIv3.

If no version set is given by the agent or by the manager, then the default version BNSIv3 is used.

6.9 Messages

Messages are sent from the supervised system to the management system, and each message is characterized by a single text line starting with a hash character (#).

6.9.1 Comment Message

The simplest form of messages are comments.

Comments are outputs from the agent which the manager can ignore. However, during installation and error search these kind of messages can be useful. A good practice is for the agent software to start a session with comments about its product data, such as the version and date of release. To ensure that a comment is not interpreted in a future version of BNSI as a special message (such as a version message or HB message), the first character after the hashtag must be a space.

Example 2

```
# Start of BNSI agent for system XYZ
# Revision of agent: R1A
```

6.9.2 Version Message

The version message with which the agent responds is a single line of text.

`#Version=<ver>`

Where `<ver>` is the highest supported version set by the agent. In version 1, this line is treated as a comment by the manager. <ver> is a number.

The following values are allowed:

- #Version=1
- #Version=2
- #Version=3

6.9.3 Heartbeat Message

HeartBeat messages are formatted as a single line of text.

`#HB`

A heartbeat message is always displayed in capital letters. There is no space between the # and the letter H.

6.9.4 Error Message

Error messages are formatted as a single line of text.

```
#Error=<error message text>
```

Neither the message text nor the behavior of the manager is defined further. The manager may ignore the message, log the message in a log-text file or generate an alarm record by itself where the message text is included as additional text. In the latter case, no clearing of the error message is expected.

Note: The manager may ignore these messages. There is no way for an agent to know if the manager side handles these messages.

6.9.5 Termination Message

Termination or exit messages are formatted as a single line of text.

```
#Exit=<exit code>
```

Table 4 Exit Codes from Agent Process

Values	Description
0	Normal termination (No problems detected).
1	Termination due to a disturbance or problem in the communication with the management system.
2	Exit due to a serious internal communication error or communication error with NE.
3	Exit due to an environmental problem.
4	Termination after local shut down.
5	Exit due to a processing error.
6	Exit due to an authentication failure or lack of authority to use a resource.
7	Exit due to lack of support for this functionality by the current implementation.
8	Exit due to database or other storage problems.
9–31	Reserved for future definitions.
32–63	Agent implementation-specific numbers.

Other exit codes may be used by the agent program. If the same values are used as exit codes from the process, then only values in the range of 0 to 63 are allowed as defined exit codes.

An agent must not send a termination message if the reason is that the communication with the manager is lost.

It is optional for the manager to act on the value of the exit message. The manager usually acts in the same manner regardless of the exit value, and only uses the value to report to the operator.

Example 3 Three Message Lines

In the example the first line is a comment, the second an error message, and the last line is an exit message.

```
# Cannot login to alarm database
#Error=DB Failure - Access denied, code=147
#Exit=6
```

Note: Avoid relaying on exit codes from processes since the exit values are not always transferred to the manager side.

6.10 Records

Event and alarm records are a special type of messages, in that they consist of several lines of text.

The Alarm, AlarmClearing, Event and SyncAlarm records are represented with the same set of fields and attributes. Every record starts with a percentage sign and a lowercase 'a' (%a) on a single line and ends with a percentage sign and an uppercase 'a' (%A). A line starting with a hash character (#) is ignored and may be used for comments. These comment lines are lost after the alarm has been processed by the management system. They may still be useful for debugging purposes.

It is possible to include free text at the end of the alarm record. This text is very useful since the operator sees it. It must be used to represent all data relevant to the operator that is not represented with the standardized (or additional) attributes.

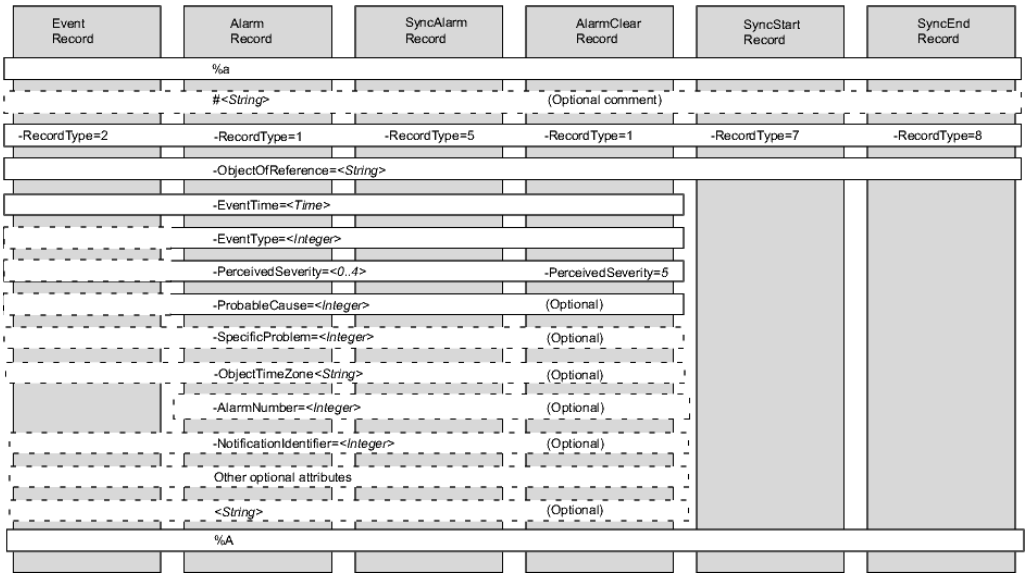

Figure 11 Record Syntax

6.10.1 Event Record

An Event record is identified with the attribute RecordType set to 2.

An Event record is handled and presented as an Alarm record. The difference is that an alarm cleaning is not expected for an Event record. The attribute AlarmNumber is of no interest for events since there is no need to correlate events with AlarmClearing records.

A variant of the event record is the internal alarm with the attribute RecordType set to 3.

6.10.2 Alarm Record

Alarm records are identified by having the attribute RecordType equal to 1 which means it is an alarm.

The attribute AlarmNumber can be used to correlate alarms with alarm clears. It is normally not necessary since the alarm should be unique based on the attributes ObjectOfReference and ProbableCause.

6.10.3 Alarm Clearing Record

The only syntax difference between an Alarm record and an AlarmClearing record is that the attribute PerceivedSeverity is set to the value 5, representing a clearing.

The following attributes must have identical values compared with what was sent in the alarm record to be cleared:

- EventType
- ObjectOfReference
- ProbableCause
- SpecificProblem
- AlarmNumber

Note: The last two attributes are optional but must always be sent in the alarm clear record if they were sent in the alarm record. Otherwise, they must not be sent.

6.10.4 Sync Alarm Record

SyncAlarm records are sent for all active alarms to make the alarm list in the management system consistent with reality.

SyncAlarm records are identified by having the RecordType attribute set to 5. Most of the other attributes have to be identical with the corresponding Alarm record. The alarm text at the end of the record and the EventTime attribute may, however, differ if it is not possible to reconstruct the values.

6.10.5 SyncStart and SyncEnd Records

The SyncStart and SyncEnd records are used to mark the beginning and end of a synchronization sequence.

The SyncStart and SyncEnd records consist of four lines.

There are two types of synchronization lists:

- For the entire agent system

  The string in the ObjectOfReference attribute should be the supervised system name in this case.

- For one sub-element at a time

  The synchronization may only contain the sub-element objects. The string in the ObjectOfReference attribute must be the supervised system name followed by a comma (,) and the sub-element name. For more information, see *Action Management* on page 52.

Note: At start of an alarm session the complete synchronization list is used. That is, when and if the manager requests a synchronization, the agent gives a synchronization list for the entire supervised system.

Example 4 Start Agent with Synchronization Phase

A short example of a start of an agent with a synchronization phase where only one alarm is currently active.

```
%a
-RecordType=7
-ObjectOfReference=SanFrancisco2
%A
%a
-RecordType=5
-ObjectOfReference=SanFrancisco2,Mic7-1,Mou8
-EventTime=19960228181214
-EventType=4
-PerceivedSeverity=1
-ProbableCause=56
-SpecificProblem=137
%A
%a
-RecordType=8
-ObjectOfReference=SanFrancisco2
%A
```

6.11 Attribute Types

All data in BNSI is in text format, however each attribute in the alarm records is of a certain type.

Attribute types:

- Number
- Text
- Time
- Object Identifier

This section provides a detailed description of each attribute that can be sent in the records described in the Records section. Some of the attributes with integers represent entries in a so-called translation map. Such a map contains cross-references between these numbers and text strings that are presented to the operator.

Most of the fields represent attributes defined in the ITU-T recommendation X.733. Indications are given below when appropriate.

Note: No line in the record can exceed the limit of 255 characters.

6.11.1 Number Attribute Types

Numbers are sequences of digits.

Numbers follow the decimal base and are usually within the range 0 to 32767 (215-1), while large numbers are within the range 0 to 2147483647 (231-1). Negative values are only allowed if specially stated for an attribute (leading zeros are allowed, but not recommended).

6.11.1.1 Perceived Severity

Type Number

Value 0-5

Description The PerceivedSeverity attribute defines the severity as described in ITU-T X.733.

Table 5 Perceived Severity Values

Value	Severity
-PerceivedSeverity=0	Indeterminate
-PerceivedSeverity=1	Critical

Value	Severity
-PerceivedSeverity=2	Major
-PerceivedSeverity=3	Minor
-PerceivedSeverity=4	Warning
-PerceivedSeverity=5	Cleared

Table 6 Perceived Severity Value Descriptions

Indeterminate (0)	Must not be used. If used, the severity is unknown.
Critical (1)	The critical severity level indicates that a condition that affects service has occurred and an immediate corrective action is required. Such a severity is reported, for example, if a managed network object becomes totally out of service and its capability must be restored. This requires an immediate action, even outside working hours.
Major (2)	The major severity level indicates that a condition that affects service has occurred and an urgent corrective action is required. Such a severity is reported, for example, when service degrades in the capacity of the managed network object and its full capability must be restored. This requires an immediate action, within working hours.
Minor (3)	The minor severity level indicates that a fault condition that does not affect service has occurred and that corrective action must be taken to prevent a more serious (for example, service-affecting) fault. Such a severity is reported when the detected alarm condition does not currently degrade the capacity of the managed network object. This requires an action at a suitable time, or at least that close observation of the situation continues.
Warning (4)	The warning severity level indicates there is a potential or impeding fault that affects service, before any significant effects have occurred. Corrective action is performed on a scheduled maintenance basis.
Cleared (5)	The clear severity level is used to clear a previously reported alarm; for more information, see the Alarm Clearing Record section in the Records topic.

6.11.1.2 Alarm Number

Type Large Number

Description The AlarmNumber attribute is optional, however the BNSI manager must be capable of handling this attribute. If alarms are numbered by the equipment that issues them or by the supervised system, this number may be entered here.

See also *Alarm Number vs Notification Identifier* on page 30.

Two alarms from the same agent and from the same network element (which normally means that they have the same object of reference) cannot have the same alarm number. An alarm number from an already cleared alarm may be reused for a new alarm.

The AlarmNumber attribute must not be used in an event record.

Note: If alarm-numbers are used, they are also used for clearing; meaning, to clear a numbered alarm, the number of both the alarm and the clearing must be the same. In other words, if AlarmNumber is present it is used as part of the unique identification of an alarm.

6.11.1.3 Notification Identifier

Type Large Number

Description This optional attribute provides an identifier for the record, used, for example, in an acknowledge action.

An agent may choose to reuse notification identifier values for the alarm records within a synchronization. These are alarm records with record type 5 and 9. The other record types must have unique notification identifier value (this also applies for repeated alarms).

Record types 7 and 8 do not have notification identifier attributes.

Note: Notification identifiers must be chosen to be unique across all records of an agent throughout the whole surveillance session.

Example 5

```
%a
-RecordType=1
-ObjectOfReference=M&252;nchen:2,b14,3:4
-NetworkOfReference=Bayern
-EventTime=19990408101722
-EventType=2
-PerceivedSeverity=2
-ProbableCause=320
-AlarmNumber=756
-NotificationIdentifier=384194
%A
```

6.11.1.4 Alarm Number vs Notification Identifier

A comparison between notification identifiers and alarm numbers.

Attribute	Used to identify	Scope of number sequence	Repeatincycleg
Alarm Number	An alarm and all repetition and the clearance of the alarm.	Unique within the scope of the managed object (object of reference).	May be repeated as soon as the previous alarm is cleared.
Notification Identifier	One single event or alarm record.	Unique within the scope of the whole agent.	Never repeated during a surveillance session.(1)

(1) The agent can restart with the first number when it reaches the last of the 2147483648 numbers. However, the agent may not use any numbers related to alarms that are still active.

6.11.1.5 Backup Status

Type Number

Value 0-1

Description The BackupStatus attribute is optional. It indicates whether the switch to the BackupObjectInstance attribute was successful or not.

Table 7 Backup Status Values

Value	Backup Status
-BackupStatus=0	Failure (False)
-BackupStatus=1	Success (True)

Table 8 Backup Status Value Definitions

Failure (0)	The switch to backup/protection object failed.
Success (1)	The switch to backup/protection object succeeded.

Example 6 Backup Object

An example of an alarm record with information about backup object.

```
%a
-RecordType=1
-ObjectOfReference=Rio:2,eopA,20,1
-EventTime=19980119193011
-EventType=4
-PerceivedSeverity=2
-ProbableCause=355
-SpecificProblem=29
-AlarmNumber=84631
-BackupObjectInstance=Rio:2,eopA,20,2
-BackupStatus=1
Automatic switch-over at Rio de Janeiro station 2 from
outgoing to S&227;o Paulo.
%A
```

6.11.1.6 Probable Cause

Type Large Number

Description The ProbableCause attribute further qualifies the alarm compared to the event type attribute. The translation map specifies a character string to be used by the user interfaces instead of the BNSI value given by the ProbableCause attribute.

The default mapping is based on the following ITU-T recommendations:

- M.3100
- X.733
- X.736 (security alarm cause)
- GSM 12.11

Though it is possible to use these translation maps in combination, it is recommended to work only with one as a basis for the default mapping. The preferred one is X.733.

Connections to some alarm sources may require additional entities for the mapping, in which case a new translation map must be created. The default mapping, which is shown in the Standard Translation Map section, is part of the standard BNSI.

Note: Avoid a new and specialized probable cause mapping, if possible. Use the standard probable causes and use the attribute SpecificProblem for further refinement of the problem specification.

Example 7 One Alarm with a Minimum of Attributes

```
%a
-RecordType=1
-ObjectOfReference=Perth3,ew:2
-EventTime=19980202231234
-EventType=3
-PerceivedSeverity=2
-ProbableCause=336
%A
```

Note: This attribute is not sent in an alarm record by BNSI agent when BNSIv3 is used. Instead BNSI agent sends a new attribute ProbableCauseText.

6.11.1.7 Acknowledge

Type Number

Value 0-2

Description The Acknowledge attribute is optional. It indicates whether the alarm is acknowledged in the supervised system. This attribute can be used in a repeated alarm to indicate that the alarm has changed status and been acknowledged in the supervised system.

Note: Spontaneous acknowledgement requests are not forwarded to the agent system.

Table 9 Acknowledge Values

Value	Acknowledge Status
-Acknowledge=0	Not acknowledged
-Acknowledge=1	Acknowledged
-Acknowledge=2	Acknowledgment regretted

Table 10 Acknowledge Value Definitions

Not Acknowledged (0)	The alarm is not acknowledged in the agent system.
Acknowledged (1)	The alarm is locally acknowledged in the agent system.
Ack. Regretted (2)	The acknowledgment of the alarm is canceled (the use of this value should be avoided if possible).

6.11.1.8 Record Type

Type Number

Value 1-9

Description The RecordType attribute indicates the type of record. See

Table 11 Record Type Values

Value	Record Type
-RecordType=1	Alarm (and alarm clear)
-RecordType=2	Event
-RecordType=3	Simple Alarm
-RecordType=4	Repeated Alarm
-RecordType=5	Synchronization Alarm
-RecordType=6	Heartbeat Alarm
-RecordType=7	Synchronization Started
-RecordType=8	Synchronization Ended
-RecordType=9	Synchronization Event

Table 12 Record Types Value Definitions

Alarm (1)	An alarm or an alarm clear record. This is the normal record type.
Event (2)	An event record. There is never a clear message for this alarm. An event is a notification with no clear.
Simple Alarm (3)	An alarm that is not a part of a future alarm synchronization. A simple alarm is also called an internal alarm. No alarm cease is expected for this type of alarm. This is typically an internal error

	message. Use Alarm (1) or Event (2) records if possible.
Repeated Alarm (4)	A repeated alarm, which means that the alarm is already sent to the management system. A repeated alarm is used, for example, with the attribute TrendIndication if an alarm increases or decreases its severity.
Sync. Alarm (5)	An alarm record within an alarm synchronization phase.
Heartbeat Alarm (6)	An alarm (or alarm clear) record due to a loss of a connection discovered by, for example, the absence of heartbeat messages. Use Alarm (1) record if possible.
Sync. Started (7)	A record indicating the start of the alarm synchronization phase.
Sync. Ended (8)	A record indicating the end of the alarm synchronization phase.
Sync. Event (9)	An event record (unacknowledged) within an alarm synchronization phase.

6.11.1.9 Specific Problem

Type Large Number

Description The SpecificProblem attribute is optional, however the BNSI manager must be capable of handling this attribute. It specifies a further refinement of the alarm than the Probable Cause is giving. If this attribute is used then it is one of the attributes that uniquely specifies the alarm.

The connection to some alarm sources may require additional entities for the mapping, in which case a new translation map must be created. There is no default mapping, since this is specific for each connected agent system.

The interpretation of the value is normally network element specific; that is, each specific problem value is normally unique/special for one network element type. However, there are of course exceptions.

Note: This attribute is not sent in an alarm record by BNSI agent when BNSIv3 is used. Instead BNSI agent sends a new attribute SpecificProblemText.

6.11.1.10 Proposed Repair Action

Type Number

Description The ProposedRepairAction attribute is optional. Connections to some alarm sources may require additional entities for the mapping.

Table 13 ProposedRepairAction Values in the Default Translation Map

Value	Proposed Repair Action
-ProposedRepairAction=0	Unknown
-ProposedRepairAction=1	No repair action required
-ProposedRepairAction=2	Repair action required

Table 14 ProposedRepairAction Value Definitions

Unknown (0)	It is unknown whether a repair action is needed or not.
No Repair Action (1)	The element may have been able to correct or bypass the faulty equipment by means of back-up equipment or protection switching.
Repair Action (2)	The fault is not cleared until repair of the faulty equipment has been carried out.

Values between 3 and 99 are reserved for future definitions. A value of 100 or greater is a customized value and the interpretation depends on the network element.

Note: This attribute is not sent in an alarm record by BNSI agent when BNSIv3 is used. Instead BNSI agent sends a new attribute ProposedRepairActionText.

6.11.1.11 MonitoredAttributeValue

Type Number

Values Both positive and negative values

Description The MonitoredAttributeValue attribute is optional. It is the value of the monitored attribute.

Example 8 MonitoredAttributeValue

The example shows the use of ThresholdInfo, MonitoredAttribute, and MonitoredAttributeValue.

```
%a
-RecordType=1
-ObjectOfReference=MSC,C1
-EventTime=19960912104500
-EventType=11
-PerceivedSeverity=1
-ProbableCause=331
-SpecificProblem=5
-ObjectTimeZone=MET
-TrendIndication=2
-ThresholdInfo=rflosses/attempts >= 0.5
-MonitoredAttribute=rflosses
-MonitoredAttributeValue=50
%A
```

6.11.1.12 State Change

Type Number

Value Values between 3 and 99 are reserved for future definitions. A value of 100 or greater is a customized value.

Description This optional attribute defines the type of state change. It must be used in records with the record type set to 2 (Event).

Table 15 State Change Values

Value	State Change Type
-StateChange=0	Unspecified
-StateChange=1	CleanRestart
-StateChange=2	AcknowledgeAll

Table 16 State Change Value Definitions

Unspecified (0)	A general and unspecified notification about a status change in the agent or in the agent system. The free text must specify the problem or the status change.
CleanRestart (1)	There are no active alarms (and no unacknowledged). This may be ignored by the management system. This status change is also detected by the manager at a new alarm synchronization.
AcknowledgeAll (2)	All currently active alarms are now acknowledged. This may be ignored by the management system.

Example 9 State Change

```
%a
-RecordType=2
-ObjectOfReference=EricS,Son
-EventTime=19981010101010
-StateChange=2
%A
```

6.11.1.13 Event Type

Type Number

Value The values are represented in the translation map.

Values between 0 and 255 are defined in standard table or reserved for future definitions. A value of 256 or greater is a customized value.

Description The EventType attribute defines the category of the alarm. The attribute categorizes the alarm.

Connections to some alarm sources may require additional entities for the mapping, in which case a new translation map should be created. The default mappings in the *Standard Translation Map* on page 67 are part of the standard BNSI.

Note: This attribute is not sent in an alarm record by BNSI agent when BNSIv3 is used. Instead BNSI agent sends a new attribute EventTypeText.

6.11.1.14 Trend Indication

Type Number

Value 0-2

Description The TrendIndication attribute is optional. It is used for alarms that may change severity after they have been issued the first time.

Table 17 Trend Indication Values

Value	Trend Indication
-TrendIndication=0	Less severe
-TrendIndication=1	No change
-TrendIndication=2	More severe

Table 18 Trend Indication Value Definitions

Less Severe (0)	There is at least one outstanding alarm of a severity higher than that of the current alarm.
No Change (1)	The perceived severity reported in the alarm is the same as the highest of any of the outstanding alarms.
More Severe (2)	The perceived severity in the current alarm is higher than that reported in any of the outstanding alarms.

6.11.1.15 Threshold Indication

Type Number

Value 1-3

Description The ThresholdIndication attribute is optional. It is used for performance measurement alarms that may change value after they have been issued the first time.

Table 19 Trend Indication Values

Value	Threshold Indication
-ThresholdIndication=1	Decreasing
-ThresholdIndication=2	Same
-ThresholdIndication=3	Increasing

Table 20 Trend Indication Value Definitions

Decreasing (1)	The threshold counter value reported in the alarm is lower than the earlier reported alarm.
Same (2)	The threshold counter value reported in the alarm is the same as the earlier reported alarm.
Increasing (3)	The threshold counter value in the current alarm is higher than that reported in the earlier alarm.

6.11.2 Text Attribute Types

Text is any sequence of characters ending with an end-of-line character.

Text strings can contain any printable character.

The text strings may be over several rows if the startup parameter multiple lines was used. The end of these multiple lines of text is marked by a line beginning with one of the following characters: a hyphen (-), percentage-sign (%) or a hash character (#).

The maximum length of the text is usually given for each attribute (see c-AttributeDefinitions). The maximum length of any single text attribute value is 2047 characters, where each line may not exceed 255 characters, and the maximum number of lines is 64.

6.11.2.1 SpecificProblemText

Type Text

Value Max 255 characters

Description The BNSI manager must be capable of handling this new attribute. It specifies a further refinement of the alarm than the Probable Cause is giving. If this attribute is used then it is one of the attributes that uniquely specifies the alarm.

The interpretation of the text is normally network element specific; that is, each specific problem text is normally unique or special for one network element type. However, there are some exceptions.

This attribute is only supported by BNSIv3 and it is sent in clear text in the alarm record.

6.11.2.2 ProbableCauseText

Type Text

Value Max 255 characters

Description The ProbableCauseText attribute further qualifies the alarm compared to the event type attribute.

This attribute is only supported by BNSIv3 and it is sent in clear text in the alarm record.

Example 10 One BNSIv3 Alarm with a Minimum of Attributes

```
%a
-RecordType=1
-ObjectOfReference=Perth3,ew:2
-EventTime=19980202231234
-EventTypeText=Processing error
-PerceivedSeverity=2
-ProbableCauseText=Software error
%A
```

6.11.2.3 ThresholdInfo

Type Text

Value Max 200 characters

Description The ThresholdInfo attribute is optional. It specifies the reason for the threshold crossing which caused the alarm.

Example 11 ThresholdInfo and ThresholdIndication

```
%a
-RecordType=4
-ObjectOfReference=Singapore-C
-EventTime=19981213031530
-EventType=2
-PerceivedSeverity=1
-ProbableCause=304
-TrendIndication=3
-ThresholdInfo=bids
%A
```

6.11.2.4 EventTypeText

Type Text

Value Max 50 characters

Description The EventTypeTextattribute defines the category of the alarm. The attribute categorizes the alarm.

This attribute is only supported by BNSIv3 and it must be in clear text in the alarm record.

6.11.2.5 MonitoredAttribute

Type Text

Value Max 30 characters

Description This optional attribute is used to name the attribute monitored for performance measurement. The value is the name of the monitored attribute, and the value of this monitored attribute is given by the *MonitoredAttributeValue* on page 35.

6.11.2.6 Free Text

Type Text

Value Maximum 2047 characters (where each line may not exceed 255 characters with a maximum of 64 lines)

Description This is an optional attribute. Text rows in a record that do not start with a hyphen (-), percentage-sign (%) or a hash character (#) character are joined into a free text attribute, which acts as additional information.

This attribute must be avoided if possible, since this text cannot be interpreted by the management system. Use ProbableCause and SpecificProblem to encode the problem as far as possible.

Example 12 Free Text

In the following example the last two lines inside the record are the free text:

```
%a
-RecordType=1
-ObjectOfReference=LondonWest-4,EngQ,24,2
-EventTime=19970421082051
-EventType=4
-PerceivedSeverity=1
-ProbableCause=335
-SpecificProblem=84
-ProposedRepairActionText=unknown
-ObjectTimeZone=UTC
-NotificationIdentifier=8339
Power problem detected. Low backup supply crossed.
Threshold: trig2,35,37,19961201130525
%A
```

Allowed free text characters are characters with hexadecimal numbers 20-7E and A0-FF in accordance with Table 24.

6.11.2.7 Object Class Of Reference

Type Text

Value Maximum 30 characters

Description The ObjectClassOfReference attribute is optional. This attribute specifies the class of the object sending the alarm.

Example 13 Object Class of Reference

```
%a
-RecordType=1
-ObjectOfReference=Rome-M1,E1
-EventTime=19951224150000
-EventType=2
-PerceivedSeverity=2
-ProbableCause=0
-SpecificProblem=20
-ObjectClassOfReference=NE
Problem: Internal communication level 2
%A
```

6.11.2.8 ProposedRepairActionText

Type Text

Value Max 255 characters

Description The BNSI manager must be capable of handling this new attribute. This attribute defines whether a repair action is needed or not. Examples are Unknown, No repair action required and Repair action required.

Unknown means that it is not known whether a repair action is needed or not.

No repair action required means the element may have been able to correct or bypass the faulty equipment, by means of back-up equipment or protection switching.

Repair action required means the fault is not cleared until a repair action of the faulty equipment has been made.

This attribute is only supported by BNSIv3 and it is sent in clear text in the alarm record.

6.11.3 Time Attribute Types

The date and time is a 14-character long text string in the following format: <yyyymmddHMMSS> where yyyy is the year, mm is the month (01-12), dd is the day (01-31), HH is the hour in 24-hour format (00-23), MM is minutes and SS is seconds (00-59). For example, if the event occurs at a quarter past ten in the evening on the 2nd of August 2004, then the time should be stated as: 20040802221500.

If necessary, the time zone is given by a special attribute.

6.11.3.1 Event Time

Type Time

Description The EventTime attribute specifies the originating time of the sent record. It should specify (as accurately as possible) the time when the alarm, event or alarm clearing occurred in the equipment.

The format does not handle time zones, and the time should therefore be specified for the same time zone for all connected Network Elements (see also *Object Time Zone* on page 42).

6.11.3.2 Object Time Zone

Type Text

Value Max 15 characters

Description The ObjectTimeZone attribute is optional. It specifies the time zone in which the EventTime for the event is given.

If no time zone attribute is used, then the same time zone used by the management system is assumed. The interpretation of the attribute value is a configuration of the customer system.

6.11.4 Object Identifier Attribute Types

Each alarm record sent to a fault management system includes a reference to the network object that generated the alarm, also called the resource. This information is sent using the ObjectOfReference attribute.

In the 'information model' used in the protocol, the supervised object is the highest object in the hierarchy. The supervised object has attributes for controlling the communication interface associated with it. It is from the supervised object that a synchronization is requested and so on. The

supervised objects are called Agent Objects (AO). This means that the object identifiers of the BNSI protocol is a relative identifier. It is relative to the AO.

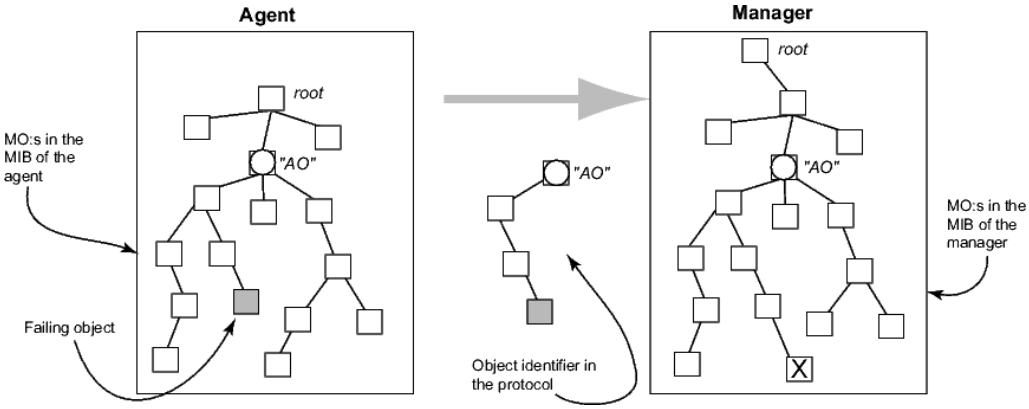

Figure 12 Object Identification in BNSI Protocol

The AO does not necessary have to be a managed object within the systems; this is an implementation issue. The AO can, for example, be a known position in the MIB hierarchy.

An AO contains any number of Element Objects (EO). EO contains other EOs on an arbitrary number of levels. The generic hierarchy is shown in the following figure.

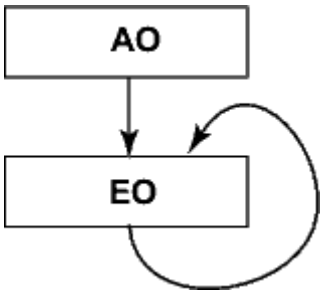

Figure 13 BNSI Naming Hierarchy

Alarms may be registered on any supervised AO or EO. The records are identified by a name that describes the path of the instance hierarchy. If an alarm is referenced to a managed object that is not represented, the alarm is registered on its nearest ancestor represented in the information model. In the most basic case, only supervised AOs have to be represented. Then, all alarms are registered on the AO.

If the agent needs to specify an absolute identifier, there is an optional attribute for this; for more information, see NetworkOfReference in the Alarm Management section that specifies the “top” of the object of reference.

Terminology

The concept of a Distinguished Name (DN), which is composed of a number of Relative Distinguished Names (RDNs), describes a containment hierarchy.

Furthermore, the DN is composed of a prefix part and a Local Distinguished Name (LDN).

On a semantic level, there is actually no difference even though the terminology differs. The object of reference in BNSI relates to the Local Distinguished Name, and the network of reference in BNSI is a prefix that, together with the object of reference, defines the Distinguished Name. The AO names and EO names relate to the RDNs.

6.11.4.1 Object of Reference

Every record sent over the BNSI interface identifies the concerned object using the ObjectOfReference attribute.

Type Object Identifier

Description The ObjectOfReference attribute identifies the network object that causes the alarm.

The string must contain at least the name of the supervised system. If a more specific name is given, the different parts of the name must be separated by commas (,). The corresponding TMN attribute is called the relative distinguished name.

The root of the distinguished name must be known by the manager. The root and the relative distinguished name gives the distinguished name. If the root is not known the agent may give this by the attribute NetworkOfReference.

This attribute is a direct mapping of the naming hierarchy model. It must start with the agent object name defined when the communication session was started, followed by a sequence of element object names. The basic syntax for the attribute is:

```
-ObjectOfReference=<AOname>[,<EOname1>[,<EOname2>...]]
```

<AOname>	The name of the supervised system given as the start argument to the agent program.
<EOname1>	The name of the top-level managed object that is the source of the alarm.
<EOname2>...]	Arbitrary level of internal managed object in the element.

The comma-sign (,), which separates the levels in the model, is reserved and you cannot use it within names. In addition, you cannot use spaces, equal signs (=) or slashes (/) in names. As a rule, use only alphanumeric characters.

The characters allowed in the object names are (given in hexadecimal numbers): 21-2B, 2D-7E and C0-FF, in accordance with Table 24.

The maximum length of any part of the name (that is, the content between two commas) is 50 characters. However, it is recommended that you keep the

names as short as possible. The total length of the ObjectOfReference attribute is 255 characters. This includes the attribute name (as described in *Interface Adapter* on page 10); therefore, the expected maximum length of the attribute value would be 235 characters - however, the maximum length is limited to 200 characters.

Names cannot be empty; that is, you cannot include two comma-signs one after another in the object of reference.

The AO name may not start with a hyphen character (-).

6.11.4.2 Proposed Structures of Object Identifiers

It is an advantage if the structure can be standardized. A standardized format makes it easier to coordinate with other applications to correlate alarms on affected services in the network.

6.11.4.2.1 Transmission Network Resources

The rules in this section are recommended to be followed for supervised systems that handle transport network.

For alarms related to transmission resources, the identification described below must be used. An example of the ObjectOfReference attribute could be:

```
-ObjectOfReference=MV36-2,Element1,Port12,1:2:1
```

`MV36-2`	The name given as the start parameter.
`Element1`	The name of the network element.
`Port12`	The name of a physical port.
`1:2:1`	The termination structure.

The part describing the internal resource object in the network element is an example of a network object that gives alarms in a transport network element. The parameters following the element parameters identify the concerned transmission resources with the sequence:

```
Port-ID,Mux-Level
```

The `Port ID` parameter uniquely identifies a particular port within the network element. The operator can set this identification in the element manager. The `Mux-Level` parameter represents the slot in the multiplex structure that the signal occupies. It can be empty or contain one or more figures separated by colon (:).

SDH:

```
AUG-# : TUG3-TU3-# : TUG2-TU2-# : TU1-VC1-#
```

PDH:

```
D3-# : D2-# : D1-#.
```

If a number has only 1 as its value, it must be removed from the sequence. For example, an STM-1 interface does not include the AUG number.

6.11.4.2.2 Switched PSTN Resource

The rules in this section are recommended to be followed for those supervised systems that manage switched PSTN network elements (exchanges).

For alarms related to exchanges, the identification must be: The element manager name, followed by the exchange name, and if needed, an equipment-id or a connection-id.

```
-ObjectOfReference=XMATE,AXE1,Route12
```

XMATE	The name given as the start parameter.
AXE1	The name of the exchange.
Route12	The name of a connection point.

6.11.4.2.3 Cellular Network Resource

For alarms related to cellular resources in the cellular access network, the identification described below is recommended.

An example of the ObjectOfReference attribute:

```
-ObjectOfReference=OSS6,BSC4,Site12
```

OSS6	The name of the element manager given at the start.
BSC4	The name of the network element.
Site12	The name of the equipment.

6.11.4.2.4 IP Network Resource

This is recommended for alarms related to computer network resources:

```
-ObjectOfReference=LAN3,192.170.170.1
```

LAN3	The name of the element manager given at the start.
192.170.170.1	The name of a computer host.

6.11.4.2.5 Support/Management System

If there is an internal error in the supervised system (the element manager) that is of interest to the management system (such as an error that occurs when the file system is full), it can be reported with the supervised system itself as ObjectOfReference.

6.11.4.2.6 Other Resource Identification

No rules are stated for how to handle resources which are not within the technology areas mentioned above. Examples of such resources are identifying hardware such as boards and external alarms like burglar alarms. The important issues to consider are still the same; this means that the operator must easily understand the identification and the identification must be unique so that there cannot be two alarms with the same identification and probable cause value.

Note: It is not as important in this scenario that the structure is standardized in detail, as these kinds of alarms are not used to integrate with tools.

6.11.4.3 Element Object Name with Type

The Element Objects can be given a type with the use of the syntax: `<type> = <value>`. The `<type>` and the `<value>` cannot contain an equals sign (=), as the equals sign is used to separate a type and a value component of the name.

This is not true for the Agent Object names. These must always be the same as the names specified by the manager at the beginning of the surveillance session.

The format of the ObjectOfReference attribute is:

```
-ObjectOfReference=XMATE,NE=AXE3,RSS=AXE3:4,ROUTE=12
```

The attribute NetworkOfReference can be used in a similar way:

```
-NetworkOfReference=network=South
```

The name `abc=123` is understood by some BNSI managers as: `abc` is the name of the type, and `123` is the real EO name. But some implementations see `abc=123` as the EO name.

The agent must only use this syntax if requested by the manager. For more information, see Interface Adapter section. The agent can ignore this request.

6.11.4.4 Backup Object Instance

Type Object Identifier

Description The BackupObjectInstance attribute is optional. It may be used if appropriate. The attribute specifies the name of a network object that functions as the backup for the network object that issued the alarm.

The format of the attribute is the same as that of the ObjectOfReference.

6.11.4.5 Network of Reference

Type Object Identifier

Description This optional attribute is an identification of the network in which the object of reference is found.

If this attribute is used it implies that the manager system and the agent system are using the same information model. The value of this attribute may be a hierarchy of networks separated by commas (,).

Note: The value of the attribute must contain at least one network name. If no network identifier exists the attribute must not be included in the record.

6.11.5 Additional Attributes

Optional attributes may be defined.

They must all start on a new line with the first character being a hyphen (-), followed by the attribute name, and an equals sign (=) between the attribute name and the value. The value must be an integer or a text string.

These attributes must be ignored by a manager not supporting them. If the management system can recognize any of these attributes, but does not have a way of presenting this attribute as an attribute of its own, then a text field (maybe an additional information field) may be used for these attributes.

Example of attributes in this group are: DeviceName, ErrorCode, AlarmCategory, AlarmId, CurrentTime and so on.

The value of these attributes may be of several rows if the manager has used the multiple-line parameter at the start of the agent. This is only applicable for text value attributes; for more information, see *Text Attribute Types* on page 38.

```
%a
-RecordType=1
-ObjectOfReference=Sony,Ericsson
-EventTime=20010101010101
```

```
-EventType=10
-PerceivedSeverity=2
-ProbableCause=317
-SpecificProblem=2
-NotificationIdentifier=637
-AdditionalInformation=Error message from disc system
Could not store file &180;event.lo&gacute;
-ProblemData=File storage problem
on disc portion 1-
File error /var/log/event.log
%A
```

Note: In the code block, the '-' character is used to separate the end of the last multiple-line additional attribute and the start of the free text.

6.12 Attributes Used in Different Record Types

The table shows a cross-reference of attributes in alarm records and different record types.

The same information is graphically indicated in the figure Record Syntax. In the table, (M) represents the mandatory attribute in a record, (O) represents the optional attribute and (-) indicates that the attribute is not applicable for this record type.

Each attribute may only appear once in each record.

In BNSIv3 the agent does not support the integer values for the attributes SP, ET, PC and PRA and instead sends them as clear text using the attributes SpecificProblemText, EventTypeText, ProbableCauseText and ProposedRepairActionText.

Note: The order of the attributes must be the same as they appear in the table.

Attribute		**Record Types**			
		(alarm) 1,3,4,5,6	**(event) 2,9**	**(sync) 7,8**	**supportedversions**
1	Record Type(1)	M	M	M	BNSI v1,v2 and v3
2	ObjectOfReference	M	M	M	BNSI v1,v2 and v3
3	NetworkOfReference	O	O	O	BNSI v1,v2 and v3
4	EventTime	M	M	-	BNSI v1,v2 and v3
5	EventType	M	O	-	BNSI v1 and v2
6	PerceivedSeverity(2)	M	O	-	BNSI v1,v2 and v3
7	ProbableCause	M	O	-	BNSI v1 and v2
8	SpecificProblem	O	O	-	BNSI v1 and v2
9	ObjectTimeZone	O	O	-	BNSI v1,v2 and v3
10	AlarmNumber	O	-	-	BNSI v1,v2 and v3
11	NotificationIdentifier	O	O	-	BNSI v1,v2 and v3
12	StateChange(3)	-	O	-	BNSI v1,v2 and v3
13	ObjectClassOfReference	O	O	-	BNSI v1,v2 and v3

Attribute		Record Types			
		(alarm) 1,3,4,5,6	(event) 2,9	(sync) 7,8	supportedversions
14	BackupObjectInstance	O	O	-	BNSI v1,v2 and v3
15	BackupStatus	O	O	-	BNSI v1,v2 and v3
16	ProposedRepairAction	O	O	-	BNSI v1 and v2
17	TrendIndication	O	-	-	BNSI v1,v2 and v3
18	Acknowledge	O	O	-	BNSI v1,v2 and v3
19	ThresholdIndication	O	O	-	BNSI v1,v2 and v3
20	ThresholdInfo	O	O	-	BNSI v1,v2 and v3
21	MonitoredAttribute	O	O	-	BNSI v1,v2 and v3
22	MonitoredAttributeValue	O	O	-	BNSI v1,v2 and v3
23	EventTypeText	M	O	-	Only BNSI v3
24	ProbableCauseText	M	O	-	Only BNSI v3
25	SpecificProblemText	O	O	-	Only BNSI v3
26	ProposedRepairActionText	O	O		Only BNSI v3
27	AdditionalAttributes	O	O	-	BNSI v1,v2 and v3
28	FreeText	O	O	-	BNSI v1,v2 and v3

(1) The type of the record is specified with the attribute RecordType.

(2) The Alarm and AlarmClearing records are of the same type; the attribute PerceivedSeverity identifies an AlarmClearing record.

(3) The StateChange record is actually an Event record which uses the StateChange parameter.

6.13 Alarm Record Example

This topic provides an example of a record containing all the defined attributes of BNSI.

The purpose of this example is to illustrate these attributes (except for the additional attributes, which are documented in the Attribute Definitions section). It is not intended to represent a realistic example of a record.

BNSIv3 supports the following new attributes which are included in the Alarm record: SpecificProblemText,ProbableCauseText,EventTypeText and ProposedRepairActionText.

Example 14 Alarm with All Attributes

```
%a
# Alarm record with all parameters used
-RecordType=1
-ObjectOfReference=Tjotahejti
-NetworkOfReference=World
-EventTime=19980421190430
-EventType=2
-EventTypeText=Quality of service
-PerceivedSeverity=1
-ProbableCause=1
-ProbableCauseText=PROCESSORS
-SpecificProblem=2
-SpecificProblemText=SYNCHRONIZATION ABORTED
-ObjectTimeZone=MET
```

```
-AlarmNumber=1
-NotificationIdentifier=9
-StateChange=2
-ObjectClassOfReference=NE
-BackupObjectInstance=Langtbortistan
-BackupStatus=1
-ProposedRepairAction=1
-TrendIndication=1
-Acknowledge=1
-ThresholdIndication=3
-ThresholdInfo=ANF greater than 1
-MonitoredAttribute=A
-MonitoredAttributeValue=2
This alarm record consist of 26 attributes,
with a total of 29 rows.
%A
```

7 Action Management

Action Management requires bi-directional communication between the manager and the agent, since messages are sent from the manager side to the agent side.

The manager can request actions to be executed by the agent. This implies two-way communication between the manager and the agent.

This two-way communication is introduced in BNSIv2 and is also present in BNSIv3, which means that a manager communicating with an agent in version 1 cannot send action messages, since there is nothing guaranteeing that the agent process is listening to the input channel and, therefore, there is a risk that the input buffer is filled up for an agent of type version 1.

Note: An Action Request goes from the manager to the agent, and the agent can answer with an Action Response.

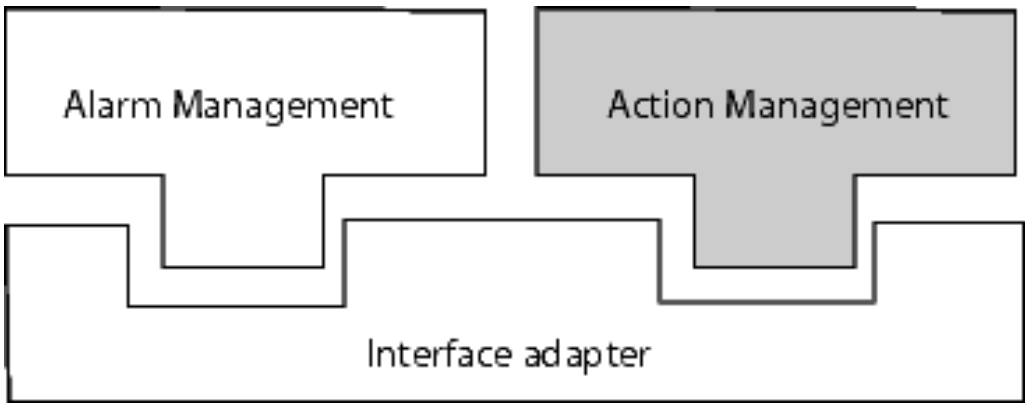

Figure 14 Major Functional Parts — Action Management

The start and termination of action management is performed together with the start and stop of the alarm management. This implies that at least basic alarm management must be supplied by the protocol to supply action management.

The start of action management and general syntax rules for the actions and action response are described in the Start of Action Management section. The other sections describe different sub-sets of actions.

The actions that control the alarm flow include actions for setting and resetting the usage of filters, and requests to re-synchronize active alarms. For an action message flow example, see *Action Management Data Flow* on page 59.

7.1 Action Management Syntax

The following example describes the syntax of the BNSI action request message flow expressed in the BNF notification.

This part of the action management goes from the manager to the agent. For information about the `chars` item, see the Character Set section.

Example 15 BNF Notification

```
actionflow ::=    [ {  alcontrol } ] eof
alcontrol ::=    ( alfilter | alautoack | alack | alterm |
alsync | alsyncspec )
alfilter ::=    &8220;*Filter=&8221; ( "On" | "Off" ) eol
alautoack ::=    &8220;*AutoAck=&8221; ( "On" | "Off" ) eol
alack ::=    &8220;*Acknowledge=&8221; digits eol
alterm1 ::=    &8220;*Terminate=&8221; digits eol
alterm2 ::=    &8220;*AlarmTerminate=&8221; digits eol
alsync ::=    &8220;*Synchronize&8221; eol
alsyncspec ::=    &8220;*Synchronize=&8221; chars "," chars eol
alxsync ::=    &8220;*XSynchronize&8221; eol
alxsyncspec ::= &8220;*XSynchronize=&8221; chars "," chars eol
eol ::=    &8216;lf&8217; | &8216;cr&8217; &8216;lf&8217; | &8216;cr&8217;
eof ::=    ( &8216;termination-signal&8217; | &8216;end-of-file&8217; )
digits ::=    { 0|1|2|3|4|5|6|7|8|9 }
chars ::=    Any sequence of 8-bit visual character.
```

Example 16 BNSI Action Response

The following example describes the syntax of the BNSI action response flow expressed in the BNF notification. This part of the action management goes from the agent to the manager and is mixed with the messages in the alarm management. For information about the `chars` item, see the Character Set section.

```
response ::=    ( exeresp | rejresp | failresp | ordresp |
denresp )
exeresp ::=    &8220;#Response=Executed &8221;  chars eol
rejresp ::=    &8220;#Response=Rejected &8221;  chars eol
failresp ::=    &8220;#Response=Failed &8221;  chars eol
ordresp ::=    &8220;#Response=Ordered &8221;  chars eol
denresp ::=    &8220;#Response=Denied &8221;  chars eol
eol ::=    &8216;lf&8217; | &8216;cr&8217; &8216;lf&8217; | &8216;cr&8217;
chars ::=    Any sequence of 8-bit visual character.
```

7.2 Start of Action Management

The action request session is started when the alarm surveillance session is started.

Action requests are allowed only for BNSIv2 and BNSIv3. The manager may only send action messages to a BNSI agent for v2 and v3.

Action messages are sent from the management system to the supervised system. Each message is sent as one text line and is received in the supervised system by the agent process. This process reads the lines from its input channel.

General Syntax of Action Messages

Each message line starts with an asterisk (*), which is directly followed by the request. Request attributes are placed after an equals sign (=) and multiple attributes are separated by commas (,). No space characters are allowed between the commas. The message line is terminated by a line feed character (<LF>). A carriage-return character (<CR>) can be used in connection with the line-feed character. A single message line may not exceed 255 characters, which includes the terminating line-feed character.

7.2.1 Action Response

A response on the execution of an action can be given by the agent in the form of an action response.

An action response starts with a hash character (#), followed by the word Response and an equals sign (=). After this reserved identification, a status word is supplied. Valid status words are listed in the table below. The status word is followed by a space, and then is the action row written last on the line. This is used by the manager to correlated the action response with the right action.

Table 21 Valid Statuses for Action Response

Action status	Description
#Response=Executed	Action fully completed.
#Response=Rejected	Action failed, but no resources were affected by the action, since nothing was done or a rollback was performed.
#Response=Failed	Action failed, but resources may be affected by the attempted action. This response can also be given if the action was partly completed.
#Response=Ordered	Queue to be done later, or action takes considerable time to complete execution. This type of response must be followed by a response of another type, such as Executed.
#Response=Denied	Authority check denied execution or it may be due to lack of resources in the local host or environment of the agent. The agent can also in this case use the response Rejected.

It is optional for the agent to use action responses. If the agent supports them, then the data flow of actions from manager to agent and the data flow of alarms/events/responses from agent to manager are asynchronous. Therefore, a manager never actively waits for an action response. But when it receives one, it can use it to handle a sequence of actions.

A single action response line cannot exceed 255 characters, which includes the terminating line-feed character.

Example 17

```
#Response=Executed *Terminate=1432
#Response=Executed *Acknowledge=20206
#Response=Denied *Filter=On
```

7.3 Alarm and Event Flow Control Actions

This section describes the actions that control the alarm flow.

7.3.1 Alarm and Event Filter Control

Action

If the alarm sending agent in the supervised system supports a filter, this filter can be turned on and off by a set of request messages. The default filter setting for an agent is to forward all events/alarms.

To activate and respectively to deactivate the filter, use the following:

```
*Filter=On
*Filter=Off
```

This is normally set at start of agent as startup parameter to the agent.

Response

An action response must be sent.

7.3.2 Automatic Alarm Acknowledgement Filter Control

Action

If the alarm sending agent in the supervised system supports a configuration filter for automatic acknowledgement of alarms in the supervised system when the alarms are forwarded to the management system, this filter can be activated and deactivated by a set of autoack-filter messages. The default automatic acknowledgement filter is not set to acknowledge any alarms.

To activate and deactivate a filtering of automatic acknowledgement are done by the action request messages:

```
*AutoAck=On
*AutoAck=Off
```

This is normally set at start of agent as startup parameter to the agent.

Response

An action response must be sent.

7.3.3 Alarm Synchronization

Action

Request the supervised system to generate and send 'spontaneous' alarm synchronization. The syntax of this action request line is:

Syntax

```
*Synchronize
```

Alternatively, the manager stops and starts the agent again with the synchronization parameter.

Response

An action response must be sent if the result is that the action is not of type Executed. If the result is to execute the action, then the alarm synchronization itself is the result and the action response can be omitted.

7.3.4 Element Specific Alarm Synchronization

Action

Request the supervised system to generate, and send 'spontaneous' alarm synchronization for one specified managed object (sub-element). Attribute values are described in the Object Identification section. Only one level of EOname is allowed (see the Records section in the Alarm Management).

Syntax

```
*Synchronize=<AOname>,<EOname>
```

Example 18 Alarm Synchronization

```
*Synchronize=EM4,NE7
```

Example 19 Element Specific Synchronization

A short example of an agent with a element specific synchronization phase (where only one alarm is currently active).

```
%a
-RecordType=7
-ObjectOfReference=EM4,NE7
%A
%a
-RecordType=5
-ObjectOfReference=EM4,NE7,EQ8
```

```
-EventTime=19981008040000
-EventType=10
-PerceivedSeverity=3
-ProbableCause=357
%A
%a
-RecordType=8
-ObjectOfReference=EM4,NE7
%A
```

Response

An action response must be sent if the result is that the action is not of type Executed. If the result is to execute the action, then the alarm synchronization itself is the result.

7.3.5 Extended Synchronization

Action

Request the supervised system to generate and send a 'spontaneous' extended synchronization.

Syntax

```
*XSynchronize
```

Alternatively, the manager stops and starts the agent again with the extended synchronization parameter.

Note: Do not use both alarm synchronization and extended synchronization towards one and the same agent. If the agent was started with extended synchronization, then use this action for synchronization. If the agent was started with alarm synchronization, then use the alarm synchronization action.

Response

An action response must be sent if the result is that the action is not of type Executed. If the result is to execute the action, then the alarm synchronization itself is the result.

7.4 Alarm Management Actions

This section describes the actions that control the alarm status. None of the actions result in an action response.

All these actions require that the agent is using alarm or event identification numbers on the event records sent to the manager. If the agent is not

supplying a NotificationIdentifier or an AlarmNumber for the alarm/event records, then the manager cannot use these actions.

7.4.1 Alarm Acknowledgement

Action

In the action management of BNSI there is an Acknowledge message to enable the management system to acknowledge alarms (or events) originating in the supervised system.

Syntax

```
*Acknowledge=<NotificationId>
```

Example 20 Alarm Acknowlegement

To be able to acknowledge alarms, the management system needs to know the NotificationIdentifier number, which then needs to be included in the alarm records (note that NotificationIdentifier is an optional attribute).

```
The output of the agent process is the following:
%a
-RecordType=1
-ObjectOfReference=nyc:45,switch-unit-31
-EventTime=20000101000001
-EventType=10
-PerceivedSeverity=3
-ProbableCause=346
-SpecificProblem=270
-NotificationIdentifier=1783
-Acknowledge=0
Program halted. Internal clock has illegal value.
Subsystem restarted.
%A
The output of the manager process to the agent:
*Acknowledge=1783
```

Response

No action response is sent as a result of alarm acknowledgment.

7.4.2 Alarm Termination

Action

To enable the management system to manually terminate alarms originating in the supervised system, there is a Termination message in the action management of BNSI.

Syntax

```
*Terminate=<NotidicationId>
```

7.5 Action Management Data Flow

Figure Action Management Traffic is an example of a messages sent in the action management of BNSI. It illustrates the direction of the different message types and the relative time relationship of between the different messages.

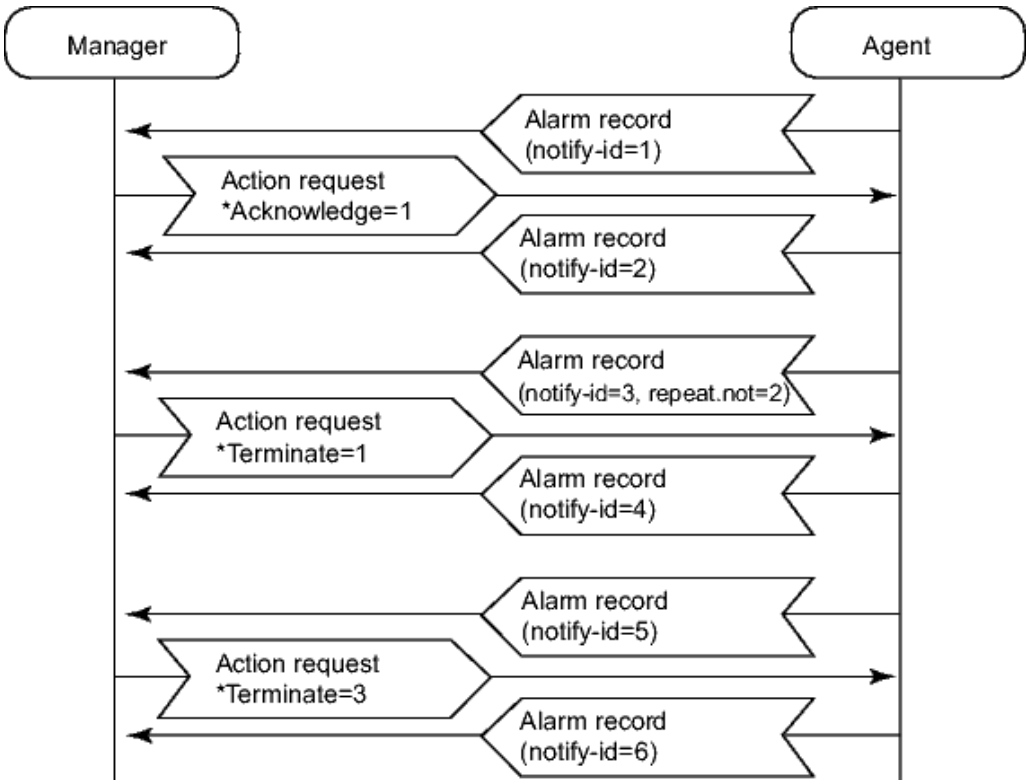

Figure 15 Action Management Traffic

8 Implementation Specification

The set of requirements (support levels) that can be fulfilled by a system supporting BNSI.

These lists are seen as an overview of the possible support of functionality that are proposed with BNSI agents and managers. Note that it is not mandatory to follow these support levels. The support levels have been added to limit the number of possible supported functionality levels of different implementations, and to make it easier to classify the functional support of a certain implementation.

Any combination of an agent and a manager is possible. That is, managers and agents can support different levels of functionality and still coexist. However, no system is better than its weakest link, meaning that it is the one with the lowest functionality support that decides the support level.

Table 22 Agent and Manager Combinations

Functional Support Level	Manager	Agent
Low Events and uncorrelated alarms.	Tiny Manager	Tiny Agent
Normal Alarms with associated alarm cease, alarm synchronization.	Alarm Manager	Alarm Agent
High Two-way alarm list synchronization.	Integration Manager	Integration Agent

8.1 Tiny Agent

Tiny Agent is used to interface a new equipment and have its events and alarms forwarded. It is normally the first step of the integration.

Tiny Agent can only handle events, that is, no clearance of an earlier event is ever sent. Tiny Agent is also referred to as the Event Agent. Since no alarms are handled, there is no need for synchronization. Some implementations of Tiny Agents can use alarms and alarm ceases but the user of the management system cannot be sure about the alarm status since no alarm synchronization is supported. If the manager requests a synchronization, the agent just ignores it. No version negotiation is necessary to support, optional attributes can exist and heartbeats must always be supported.

The required functionalities for the Tiny Agent are as follows:

1. The agent must be able to send event records (record type 2).
2. If instructed, the BNSI agent must ensure that an event or alarm or a heartbeat message is always sent within the specified time. This means

that if no event or alarm is sent within the time specified, a heartbeat message must be sent.

3. The way the attributes are interpreted in the event and alarm records must be uniform for all connected systems. That is, for all sub-ordinate elements the alarm attributes must be interpreted and translated in the same way by the BNSI agent.

4. The ObjectOfReference attribute must uniformly identify the failing network object according to section Object Identification.

5. If the BNSI agent system loses contact with a sub-ordinate element, an event must be sent to inform the management system about the situation.

6. If the BNSI agent system detects a communication problem with the management system, the communication must be closed. In this instance, the management system later tries to re-establish communication to synchronize with the supervised system.

7. The BNSI agent must support the mandatory attributes and free text, as described in Interface Adapter section. The optional attribute SpecificProblems is also used.

8. The agent must at least be able to support 7-bit ASCII code, that is, the first half of ISO 8859 character set. See the Character Set section from more information. It is better to support the full 8-bit character set.

9. The agent must not fail due to command line options or action messages which it does not understand. These types of options and messages are ignored instead.

8.2 Alarm Agent

Alarm Agent provides full support for alarms and events and can generate an alarm synchronization for the agent system.

Alarm Agents support extended synchronization. Version negotiation is optional and any support for action management is optional. To fulfill the requirements for this agent is to fulfill the requirements of the original BNSI agent specification. Alarm Agent is also referred to as the Basic Agent.

Alarm Agent is the normal functional level of a BNSI agent and is used when full alarm forwarding is needed, but integration from the management system to the agent system is not necessarily needed.

The required functionalities for the Alarm Agent are as follows:

1. The agent must fulfill all requirements of the Tiny Agent.

2. Each alarm and alarm clearing must be sent only once. Alarms cannot be duplicated or forgotten.

3. If the BNSI agent system loses contact with a sub-ordinate element, an alarm must be sent. When communication is re-established, an alarm clearing must be sent, and the management system must be updated with the current alarm situation in the sub-ordinate element. This replaces item 5 in the previous list.

4. The supervised system must be able to send all active alarms as a series of synchronization alarms. This must be done immediately but can be delayed if the reason for the synchronization request is problems in the DCN and the supervised system is busy retrieving the alarms from the connected sub-elements such as network elements. In the latter case, heartbeat messages must be sent prior to the start of the synchronization phase.

5. The agent must be able to send both alarms and alarm clearing. The agent uses, when possible, alarms and alarm clearings to report fault instead of events. The alarm and the corresponding alarm clearing must have the same identifier.

6. If the status of an active alarm is changed; for example if the severity of the alarm is changed (increased or decreased), this must be reported as a repeated alarm record (record type 4). Note that the clearing of the alarm is not considered as a repetition of the alarm.

7. Each alarm record sent must be distinguished uniquely by an identification comprised of the attributes EventType, ObjectOfReference, ProbableCause, SpecificProblem and AlarmNumber, if used.

8.3 Integration Agent

Integration Agent provides full support for alarms and events, and supports action management.

For example acknowledgement of alarms to be transferred in both directions, and termination of alarms from the management system to the agent system. Version negotiation is also supported. Integration Agent supports both alarm management and action management.

Integration Agent is used when the status of the alarms must be kept in sync between the management system and the agent system. And since the status of an alarm can be changed in both systems, this implies a bidirectional communication.

The required functionalities for the Integration Agent are as follows:

1. The agent must fulfill all requirements of the Alarm Agent.

2. Version messages are handled to support different support levels of functionality in the manager and the agent.

3. Termination codes and exit codes for the agents are defined and supported. See the Alarm Management section for more information.

4. Action request messages for alarm acknowledgement and for alarm termination are received and handled by the agent. This requires the use of NotificationIdentifier in the alarm records.

5. The agent supports the possibility to use an event and alarm filter on the forwarded events and alarms. The agent can also acknowledge the events and alarms in the local system when these are forwarded.

6. The agent is capable of reporting events and alarms in a specified time zone. See the Alarm Management section for more information.

7. Whenever feasible, the agent must give a report back to the manager on a requested action. See the Action Management section for more information. The action responses must be given as quick as possible. If the manager gains anything from reading these responses, it normally needs them approximately within 30 seconds. But this is up to each integration project to define.

8.4 Tiny Manager

Tiny Manager is a way of entering events and alarms in to a management system, but it can also be that the fault management system for which this BNSI manager is implemented do only support event logging

Tiny Manager supports alarm handling, but does not request synchronization of the agent as this is not supported. Optional attributes are handled. Heartbeats check must always be possible to handled by a manager. Tiny Manager considers all incoming events and alarms as separate events, in other words, no correlation between the alarm and alarm clearing is done.

The required functionalities for the Tiny Manager are as follows:

- The management system must be capable of starting BNSI agents in the supervised systems. This implies that the management system properly takes care of names for login accounts and any possible passwords, and that it must support at least one of the interface adapters for remote execution (ssh, rexec or remsh). See the Interface Adapter section for more information. The manager must be able to handle different interface adapters for different agents.

- The manager must be able to read and understand event records. It must also accept alarms and alarm clearings, but can consider these separate events (Record types: 1, 2, 3,4 and 6). See the SAlarm Management section for more information. Any optional attributes are allowed. If the optional attributes are not supported by the manager, they can be ignored or added to the free text attribute.

- The management system must support default translation maps and handling of additional maps. In other words, it must be possible to define additional mapping for the EventType and ProbableCause attributes. See the Standard Translation Map section for more information. If the

SpecificProblem and ProposedRepairAction attributes are translated, it is possible to define new mappings for these attributes.

Note: In BNSIv3, additional Translate Map is not supported.

- If the heartbeat surveillance fails, the management system must indicate or report this to the operator, and perform reconnection attempts within a specified time frame. This reconnect procedure must be repeated until the operator requests it to stop.
- Any corrupt records must be recognised and reported to the operator by the management system.
- The manager must be able to support an 8-bit character set. See the Character Set section for more information.
- The manager must accept both carriage-return and line-feed characters as an end of row. The manager must accept extra carriage-return characters in connection with the line-feed characters at end of each line.
- Events and alarms must be distinguished by the EventType, ObjectOfReference, ProbableCause and SpecificProblem attributes. It must also be possible to use the AlarmNumber attribute for distinguishing an alarm.

8.5 Alarm Manager

Alarm Manager handles alarms with alarm synchronization and supports requests for filters and version set negotiations.

This is normally the lowest level of supported functionality for a manager. To fulfill the requirements for this manager is to fulfill the requirements of the original BNSI manager specification. Alarm Manager is also referred to as the Basic Manager.

Alarm Manager should be used for passive alarm reception. In other words, it is passive in the sense that there is no action management, but it fully handles alarms.

The required functionalities for the Alarm Manager are as follows:

- Alarm Manager must fulfill all requirements of the Tiny Manager.
- The manager must support synchronization of alarm lists.
- The manager must detect and report all interrupted or erroneous synchronization phases.
- The manager can also receive new synchronization during the surveillance session.

- The manager must accept and handle a delayed synchronization.
- The manager must be able to handle alarms and alarm clearing records, and be able to correlate the alarm clearing with the corresponding alarm.
- If an agent fails to send event and alarm records or heartbeat messages within a specified time interval, the manager must be capable of detecting this absence.
- The manager must accept and handle repeated alarms (record type 4) in a proper way. However, one legal way of handling them are to ignore them, but the manager must at least detect that it is a repeated alarm and not confuse it with a new alarm instance.
- Alarm Manager must handle the AlarmNumber attribute. See the Alarm Management section for more information.

8.6 Integration Manager

Integration Manager is a full alarm management and action management manager.

Integration Manager handles all types of records and messages, and also handles request messages and spontaneous synchronization of sub-elements.

Integration Manager is used when the status of the alarms must be kept in sync between the agent system and the manager system. Since the status of an alarm is changed in both systems, this implies a bidirectional communication.

Note: Integration Manager is not supported on the RSH and REXEC interface adapters. It is only supported on SSH.

The required functionality for the Integration Manager is as follows:

- Integration Manager must fulfil all requirements of the Alarm Manager.
- Version messages are handled to support different support levels of functionality in the manager and the agent.
- Action request messages from the manager system to the agent system are supported.
- Error messages and exit messages must be read and reported to the user.
- If the manager is set up to use alarm filters or auto-acknowledgement, this is requested by the agents when started. See Interface Adapter section for more information.
- The manager must be capable of differentiating the EventTime attribute regarding the time zone specified in the ObjectTimeZone attribute. If no

ObjectTimeZone is supplied in the event or alarm record, the manager can use a local configer-data to determine in which timezone the agent is located.

- The manager must handle NotificationIdentifier attributes in the event and alarm records, since action management is supported and these identifiers are used to specify the object.
- The manager should use the action response messages to determine the status of the requested action. Note that the manager cannot trust the agent to always send a response, and that some responses can be heavily delayed. The wait time for responses must be configurable.
- At least the Terminate action and the Acknowledge action must be implemented.

9 Standard Translation Map

This section describes the default translation maps for event types and probable causes.

9.1 Event Type Mapping

In the default translation map the following EventType values are allowed.

Table 23 Even Type Mapping

Event Type	BNSI Value	Comment
Unknown event type (UKN)	0	Must not be used. If used, the type is unknown.
Communications Alarm (COM)	2	An alarm of this type is associated with the procedure and/or process required to convey information from one point to another (X.733).
Environmental Alarm (ENV)	3	An alarm of this type is associated with a condition related to an enclosure in which the equipment resides (X.733).
Equipment Alarm (EQP)	4	An alarm of this type is associated with an equipment fault (X.733).
Administrative Alarm (ADM)	6	An alarm of this type is associated with the system administration of network elements and equipments.
Switching or cross-connecting Alarm	7	An alarm of this type is associated with switching or cross-connecting equipment.
Performance Event (PRF)	9	A performance measurement event, probably originating from a threshold crossing.
Processing Error Alarm (PRO)	10	An alarm of this type is associated with a software or processing fault (X.733).
Quality of Service Alarm (QOS)	11	An alarm of this type is associated with a degradation in the quality of a service (X.733).
Integrity Violation	15	An indication that information may have been illegally modified, inserted or deleted (X.736).
Operational Violation	16	An indication that the provision of the requested service was not possible due to the unavailability, malfunction or incorrect invocation of the service (X.736).
Physical Violation	17	An indication that a physical resource has been violated in a way that suggests a security attack (X.736).

Event Type	BNSI Value	Comment
Security Service Violation	18	An indication that a security attack has been detected by a security service or mechanism (X.736).
Time Domain Violation	19	An indication that an event has occurred at an unexpected or prohibited time (X.736).

9.2 Probable Cause Mapping

This section lists the default translation map of ProbableCause, their equivalents in digits, and their corresponding recommended event types and severity. Sources of these probable causes are from M.3100, X.733, X.736 and GSM 12.11.

9.2.1 Probable Cause Mapping on M.3100 Sources (0-199)

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Indeterminate	0	Unknown	Major
Alarm Indication Signal (AIS)	1	Communications	Warning
Call Setup Failure	2	Communications	Warning
Degraded Signal	3	Communications	Minor
Far End Receiver Failure (FERF)	4	Communications	Warning
Framing Error	5	Communications	Major
Loss Of Frame (LOF)	6	Communications	Major
Loss Of Pointer (LOP)	7	Communications	Major
Loss Of Signal (LOS)	8	Communications	Major
Payload Type Mismatch	9	Communications	Minor
Transmission Error	10	Communications	Major
Remote Alarm Interface	11	Communications	Minor
Excessive Bit Error Rate (EBER)	12	Communications	Major
Path Trace Mismatch	13	Communications	Minor
Unavailable	14	Communications	Minor
Signal Label Mismatch	15	Communications	Minor
Loss Of Multi Frame	16	Communications	Major
Receive Failure	17	Communications	-
Transmit Failure	18	Communications	-
Modulation Failure	19	Communications	-
Demodulation Failure	20	Communications	-
Broadcast Channel Failure	21	Communications	-

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Connection Establishment Error	22	Communications	-
Invalid Message Received	23	Communications	-
Local Node Transmission Error	24	Communications	-
Remote Node Transmission Error	25	Communications	-
Routing Failure	26	Communications	-
Back Plane Failure	51	Equipment	Major
Data Set Problem	52	Equipment	Minor
Equipment Identifier Duplication	53	Equipment	Minor
External IF Device Problem	54	Equipment	Minor
Line Card Problem	55	Equipment	Minor
Multiplexer Problem	56	Equipment	Major
NE Identifier Duplication	57	Equipment	Warning
Power Problem	58	Equipment	Major
Processor Problem	59	Equipment	Major
Protection Path Failure	60	Equipment	Minor
Receiver Failure	61	Equipment	Major
Replaceable Unit Missing	62	Equipment	Warning
Replaceable Unit Type Mismatch	63	Equipment	Major
Synchronization Source Mismatch	64	Equipment	Warning
Terminal Problem	65	Equipment	Minor
Timing Problem	66	Equipment	Warning
Transmitter Failure	67	Equipment	Major
Trunk Card Problem	68	Equipment	Major
Replaceable Unit Problem	69	Equipment	Minor
Real Time Clock Failure	70	Equipment	-
Antenna Failure	71	Equipment	-
Battery Charging Failure	72	Equipment	-
DiskFailure	73	Equipment	-
Frequency Hopping Failure	74	Equipment	-
IODeviceError	75	Equipment	-
Loss of Syncronisation	76	Equipment	-
Loss of Redundancy	77	Equipment	-
Power Supply Failure	78	Equipment	-
Signal Quality Evaluation Failure	79	Equipment	-

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Tranceiver Failure	80	Equipment	-
Protection Mechanism Failure	81	Equipment	-
Protecting Resource Failure	82	Equipment	-
Air Compressor Failure	101	Environmental	Minor
Air Conditioning Failure	102	Environmental	Minor
Air Dryer Failure	103	Environmental	Warning
Battery Discharging	104	Environmental	Minor
Battery Failure	105	Environmental	Major
Commercial Power Failure	106	Environmental	Major
Cooling Fan Failure	107	Environmental	Major
Engine Failure	108	Environmental	Major
Fire Detector Failure	109	Environmental	Major
Fuse Failure	110	Environmental	Major
Generator Failure	111	Environmental	Critical
Low Battery Threshold	112	Environmental	Major
Pump Failure	113	Environmental	Major
Rectifier Failure	114	Environmental	Minor
Rectifier High Voltage	115	Environmental	Minor
Rectifier Low F Voltage	116	Environmental	Minor
Ventilations System Failure	117	Environmental	Minor
Enclosure Door Open	118	Environmental	Warning
Explosive Gas	119	Environmental	Major
Fire	120	Environmental	Critical
Flood	121	Environmental	Critical
High Humidity	122	Environmental	Major
High Temperature	123	Environmental	Major
High Wind	124	Environmental	Minor
Ice Build Up	125	Environmental	Minor
Intrusion Detection	126	Environmental	Major
Low Fuel	127	Environmental	Minor
Low Humidity	128	Environmental	Minor
Low Cable Pressure	129	Environmental	Major
Low Temperature	130	Environmental	Minor
Low Water	131	Environmental	Minor
Smoke	132	Environmental	Critical
Toxic Gas	133	Environmental	Major
Cooling System Failure	134	Environmental	-
External Equipment Failure	135	Environmental	-
External Point Failure	136	Environmental	-

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Storage Capacity Problem	151	Processing error	Minor
Memory Mismatch	152	Processing error	Major
Corrupt Data	153	Processing error	Major
Out Of CPU Cycles	154	Processing error	Minor
Software Environment Problem	155	Processing error	Minor
Software Download Failure	156	Processing error	Minor
Loss of Real Time	157	Processing Error	-
Application Subsystem Failure	158	Processing Error	-
Configuration or Customisation Error	159	Processing Error	-
Database Inconsistency	160	Processing Error	-
File Error	161	Processing Error	-
Out of Memory	162	Processing Error	-
Software Error	163	Processing Error	-
Timeout Expired	164	Processing Error	-
Underplaying Resource Avaliable	165	Processing Error	-
Version Mismatch	166	Processing Error	-
Bandwidth Reduced	201	Quality of Service	-
Congestion	202	Quality of Service	-
Excessive Error Rate	203	Quality of Service	-
Excessive Response Time	204	Quality of Service	-
Excessive Retransmission Rate	205	Quality of Service	-
Reduced Logging Capability	206	Quality of Service	-
System Resources Overload	207	Quality of Service	-

9.2.2 Probable Cause Mapping on X.733 Sources (300-399)

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Adapter Error	301	Equipment	Major
Application Subsystem Failure	302	Processing error	Major
Bandwidth Reduction	303	Quality of service	Minor
Call Establishment Error	304	Communications	Warning
Communication Protocol Error	305	Communications	Minor

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Communication Subsystem Failure	306	Communications	Major
Configuration or Customizing Error	307	Processing error	Minor
Congestion	308	Quality of service	Major
Corrupt Data	309	Processing error	Major
CPU Cycles Limit Exceeded	310	Processing error	Warning
Data Set or Modem Error	311	Equipment	Warning
Degraded Signal	312	Communications	Minor
DTE-DCE Interface Error	313	Communications	Major
Enclosure Door Open	314	Environmental	
Equipment Malfunction	315	Equipment	Minor
Excessive Vibration	316	Environmental	Minor
File Error	317	Processing error	Minor
Fire Detection	318	Environmental	Critical
Flood Detection	319	Environmental	Critical
Framing Error	320	Communications	Major
Heating or Ventilation or Cooling System Problem	321	Environmental	Minor
Humidity Unacceptable	322	Environmental	Major
Input/Output Device Error	323	Equipment	Minor
Input Device Error	324	Equipment	Minor
LAN Error	325	Communications	Minor
Leak Detection	326	Environmental	Major
Local Node Transmission Error	327	Communications	Major
Loss of Frame	328	Communications	Major
Loss of Signal	329	Communications	Major
Material Supply Exhausted	330	Environmental	Minor
Multiplexer Problem	331	Equipment	Major
Out of Memory	332	Processing error	Major
Output Device Error	333	Equipment	Minor
Performance Degraded	334	Quality of service	Major
Power Problem	335	Equipment	Critical
Pressure Unacceptable	336	Environmental	Major
Processor Problem	337	Equipment	Major
Pump Failure	338	Environmental	Major
Queue Size Exceeded	339	Quality of service	Major
Receive Failure	340	Equipment	Major
Receiver Failure	341	Equipment	Major
Remote Node Transmission	342	Communications	Major

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Error			
Resource at or Nearing Capacity	343	Quality of service	Major
Response Time Excessive	344	Quality of service	Major
Re-transmission Rate Excessive	345	Quality of service	Major
Software Error	346	Processing error	Minor
Software Program Abnormally Terminated	347	Processing error	Major
Software Program Error	348	Processing error	Minor
Storage Capacity Problem	349	Processing error	Major
Temperature Unacceptable	350	Environmental	Major
Threshold Crossed	351	Quality of service	Major
Timing Problem	352	Equipment	Minor
Toxic Leak Detected	353	Environmental	Major
Transmit Failure	354	Equipment	Major
Transmitter Failure	355	Equipment	Major
Underlaying Resource Unavailable	356	Processing error	Major
Version Mismatch	357	Processing error	Minor

9.2.3 Probable Cause Mapping on GSM 12.11 Sources (500-599)

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
A-bis to BTS interface failure	501	Equipment	-
A-bis to TRX interface failure	502	Equipment	-
Antenna problem	503	Equipment	-
Battery breakdown	504	Equipment	-
Battery charging fault	505	Equipment	-
Clock synchronisation problem	506	Equipment	-
Combiner problem	507	Equipment	-
Disk problem	508	Equipment	-
Equipment failure	509	Equipment	-
Excessive receiver temperature	510	Equipment	-
Excessive transmitter output power	511	Equipment	-
Excessive transmitter temperature	512	Equipment	-
Frequency hopping degraded	513	Equipment	-

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
Frequency hopping failure	514	Equipment	-
Frequency redefinition failed	515	Equipment	-
Line interface failure	516	Equipment	-
Link failure	517	Equipment	-
Loss of synchronisation	518	Equipment	-
Lost redundancy	519	Equipment	-
Mains breakdown with battery back-up	520	Equipment	-
Mains breakdown without battery back-up	521	Equipment	-
Power supply failure	522	Equipment	-
Receiver antenna fault	523	Equipment	-
Receiver Failure	524	Equipment	-
Receiver multicoupler failure	525	Equipment	-
Reduced transmitter output power	526	Equipment	-
Signal quality evaluation fault	527	Equipment	-
Timeslot hardware failure	528	Equipment	-
Transceiver problem	529	Equipment	-
Transcoder problem	530	Equipment	-
Transcoder or rate adapter problem	531	Equipment	-
Transmitter antenna failure	532	Equipment	-
Transmitter antenna not adjusted	533	Equipment	-
Transmitter failure	534	Equipment	-
Transmitter low voltage or current	535	Equipment	-
Transmitter off frequency	536	Equipment	-
Database inconsistency	537	Processing error	-
File system call unsuccessful	538	Processing error	-
Input parameter out of range	539	Processing error	-
Invalid parameter	540	Processing error	-
Invalid pointer	541	Processing error	-
Message not expected	542	Processing error	-
Message not initialised	543	Processing error	-
Message out of sequence	544	Processing error	-

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
System call unsuccessful	545	Processing error	-
Timeout expired	546	Processing error	-
Variable out of range	547	Processing error	-
Watch dog timer expired	548	Processing error	-
Cooling system failure	549	Environmental	-
External equipment failure	550	Environmental	-
External power supply failure	551	Environmental	-
External transmission device failure	552	Environmental	-
Fan failure	553	Environmental	-
High humidity	554	Environmental	-
High temperature	555	Environmental	-
Intrusion detected	556	Environmental	-
Low humidity	557	Environmental	-
Low temperature	558	Environmental	-
Smoke detected	559	Environmental	-
Excessive Error Rate	560	Quality of service	-
Reduced alarm reporting	561	Quality of service	-
Reduced event reporting	562	Quality of service	-
Reduced logging capability	563	Quality of service	-
System resources overload	564	Quality of service	-
Broadcast channel failure	565	Communications	-
Connection establishment error	566	Communications	-
Invalid message received	567	Communications	-
Invalid MSU received	568	Communications	-
LAPD link protocol failure	569	Communications	-
Local alarm indication	570	Communications	-
Remote alarm indication	571	Communications	-
Routing failure	572	Communications	-
SS7 protocol failure	573	Communications	-
Transmission error	574	Communications	-
RNC-RNC link Error	575	Communications	-
I610 Loc End to End	600	Communications	-
I610 Ais End to End	601	Communications	-
I610 Rdi End to End	602	Communications	-
I610 Lock Segment Link Side	603	Communications	-

Probable Cause	BNSI Value	Recommended Event Type	Recommended Severity
I610 Lock Segment Core Side	604	Communications	-
I610 Ais Segment Link Side	605	Communications	-
I610 Ais Segment Core Side	606	Communications	-
I610 Rdi Segment Link Side	607	Communications	-
I610 Rdi Segment Core Side	608	Communications	-

10 Character Set

BNSI uses an 8-bit ISO8859 character set.

An agent is required to support the first half, which is equal to the ASCII character set.

The manager must support the entire character set. No 16-bit character set is supported by this BNSI version.

Table 24 BNSI 8–Bit Character Set

00 nul	01(1)	02 (1)	03 (1)	04 eot	05 (1)	06 (1)	07 (1)
08(1)	09 (1)	0A lf	0B vt	0C (1)		0E (1)	0F (1)
10 (1)	11 (1)	12 (1)	13 (1)	14 (1)	15 (1)	16 (1)	17 (1)
18	19	1A	1B	1C	1D	1E	1F
20 sp	21 !	22 “	23 #	24 $	25 %	26 &	27 '
28 (	29)	2A *	2B +	2C ,	2D -	2E .	2F /
30 0	31 1	32 2	33 3	34 4	35 5	36 6	37 7
38 8	39 9	3A :	3B ;	3C <	3D =	3E >	3F ?
40 @	41 A	42 B	43 C	44 D	45 E	46 F	47 G
48 H	49 I	4A J	4B K	4C L	4D M	4E N	4F O
50 P	51 Q	52 R	53 S	54 T	55 U	56 V	57 W
58 X	59 Y	5A Z	5B [	5C \	5D]	5E ^	5F _
60 `	61 a	62 b	63 c	64 d	65 e	66 f	67 g
68 h	69 i	6A j	6B k	6C l	6D m	6E n	6F o
70 p	71 q	72 r	73 s	74 t	75 u	76 v	77 w
78 x	79 y	7A z	7B {	7C |	7D }	7E ~	7F
80 (1)	81 (1)	82 (1)	83 (1)	84 (1)	85 (1)	86 (1)	87 (1)
88 (1)	89 (1)	8A (1)	8B (1)	8C (1)	8D (1)	8E (1)	8F (1)
90 (1)	91 (1)	92 (1)	93 (1)	94 (1)	95 (1)	96 (1)	97 (1)
98 (1)	99 (1)	9A (1)	9B (1)	9C (1)	9D (1)	9E (1)	9F (1)
A0 sp	A1 ¡	A2 ¢	A3 £	A4 ¤	A5 ¥	A6 |	A7 §
A8 ¨	A9 ©	AA ª	AB «	AC ¬	AD –	AE ®	AF ¯
B0 °	B1 ±	B2 2	B3 3	B4 ´	B5 u	B6 ¶	B7 ·
B8 ¸	B9 1	BA º	BB »	BC 1/4	BD 1/2	BE 3/4	BF ¿
C0 À	C1 Á	C2 Â	C3 Ã	C4 Ä	C5 Å	C6 Æ	C7 Ç
C8 È	C9 É	CA Ê	CB Ë	CC Ì	CD Í	CE Î	CF Ï
D0 D	D1 Ñ	D2 Ò	D3 Ó	D4 Ô	D5 Õ	D6 Ö	D7 x
D8 Ø	D9 Ù	DA Ú	DB Û	DC Ü	DD Y	DE P	DF ß
E0 à	E1 á	E2 â	E3 ã	E4 ä	E5 å	E6 æ	E7 ç
E8 è	E9 é	EA ê	EB ë	EC ì	ED í	EE î	EF ï
F0 d	F1 ñ	F2 ò	F3 ó	F4 ô	F5 õ	F6 ö	F7 /

F8 ø	F9 ù	FA ú	FB û	FC ü	FD y	FE p	FF ÿ

(1) This number does not produce any printable characters.

11 Extended Examples

This section shows examples of the alarm printouts presented in different ways to cover as many aspects as possible to further clarify the BNSI.

11.1 Octet Sequence of Alarm Printout

This section describes an example of an alarm record with its corresponding octet sequence.

Octet Sequence in Hexadecimal	Text Sequence
25 61 0A	%a
2D 52 65 63 6F 72 64 54 79 70 65 3D 31 0A	-RecordType=1
2D 4F 6A 65 63 74 4F 66 52 65 66 65 72 65 63 65 3D 45 4D 31 2C 4E 45 31 0A	-ObjectOfReference=EM1,NE1
2D 76 65 6E 74 54 69 6D 65 3D 31 39 39 38 30 39 32 30 31 33 33 36 30 34 0A	-EventTime=19980920133604
2D 76 65 6E 74 54 79 70 65 3D 34 0A	-EventType=4
2D 50 65 72 63 65 69 76 65 64 53 65 76 65 72 69 74 79 3D 31 0A	-PerceivedSeverity=1
2D 50 72 6F 62 61 62 6C 65 43 61 75 73 65 3D 35 36 0A	-ProbableCause=56
2D 53 70 65 63 69 66 69 63 50 72 6F 62 6C 65 6D 3D	-SpecificProblem=137
25 41 0A	%A
23 48 42 0A	#HB

11.2 Alarm Printout with Comments

To specify an output line containing a keyboard character that is non-printable, the name of the key is given in capital letters, surrounded by angle brackets. In the following example, the tabulation character is represented by the character string <TAB> and the line-feed by <LF>.

For this example, assume you have two active alarms when you start the agent, and that within one minute after the agent is started you receive a new alarm. The agent is terminated after another minute.

The agent program is started with the following command (see Request for Start of Surveillance section):

```
SendAlarms ABC -sync -hbint 30
```

The output of the agent is similar to the following example (see the comments after the example):

```
#Version=2<LF>
#Start of agent:<TAB>ABC<LF>

%a<LF>
-RecordType=7<LF>                          1.
-ObjectOfReference=ABC<LF>
%A<LF>

%a<LF>
-RecordType=5<LF>                          2.
-ObjectOfReference=ABC,NE1<LF>
-EventTime=20040918083130<LF>              3.
-EventType=2<LF>
-PerceivedSeverity=1<LF>
-ProbableCause=329<LF>
-SpecificProblem=20<LF>
%A<LF>

%a<LF>
-RecordType=5<LF>
-ObjectOfReference=ABC,NE2<LF>             4.
-EventTime=20040918090043<LF>
-EventType=2<LF>
-PerceivedSeverity=2<LF>
-ProbableCause=320<LF>
-SpecificProblem=48<LF>
%A<LF>

%a<LF>
-RecordType=8<LF>                          5.
-ObjectOfReference=ABC<LF>
%A<LF>

#HB<LF>                                    6.

%a<LF>
-RecordType=1<LF>                          7.
-ObjectOfReference=ABC,NE1<LF>
-EventTime=19970918091307<LF>
-EventType=11<LF>
-PerceivedSeverity=2<LF>
-ProbableCause=303<LF>
-SpecificProblem=20<LF>
Bandwidth reduced<LF>
%A<LF>

#HB<LF>

#Exit=0<LF><EOF>                           8.
```

1. Record type 7 indicates the beginning of an alarm synchronization.
2. Alarm record in a synchronization.
3. The date and time must be in this format.
4. `RecordType`, `ObjectOfReference`, `EventType` and `ProbableCause` must always be a part of an alarm record.
5. Record type 8 indicates the end of an alarm synchronization.
6. Heartbeat messages must be sent if no other data is sent within the HB interval time.
7. Alarm record.
8. Any additional text can be included.

11.3 Expanded Alarm Record

The following example is a record containing all the defined attributes of BNSI. This example is not a realistic example of a record, it just shows all the attributes (except for additional attributes).

Example 21 Alarm with All Attributes

```
%a
# Alarm record with all parameters used
-RecordType=1
-ObjectOfReference=Tjotahejti
-NetworkOfReference=World
-EventTime=19980421190430
-EventType=2
-PerceivedSeverity=1
-ProbableCause=1
-SpecificProblem=2
-ObjectTimeZone=MET
-AlarmNumber=1
-NotificationIdentifier=9
-StateChange=2
-ObjectClassOfReference=NE
-BackupObjectInstance=Langtbortistan
-BackupStatus=1
-ProposedRepairAction=1
-TrendIndication=1
-Acknowledge=1
-ThresholdIndication=3
-ThresholdInfo=ANF greater than 1
-MonitoredAttribute=A
-MonitoredAttributeValue=2
This alarm record consist of 23 attributes,
with a total of 27 rows.
%A
```

11.4 Longer Alarm Printout

The following example is a longer BNSI alarm printout example of how a normal BNSI alarm sequence can look like.

Example 22

```
%a
-RecordType=1
-ObjectOfReference=xt,p18
-EventTime=20040119163900
-EventType=2
-PerceivedSeverity=3
-ProbableCause=305
-SpecificProblem=703
Code errors
No input clock
%A

%a
-RecordType=1
-ObjectOfReference=xt,p18
-EventTime=20040119163904
-EventType=2
-PerceivedSeverity=2
-ProbableCause=320
-SpecificProblem=714
Loss of multiframe sync
%A

%a
-RecordType=2
-ObjectOfReference=xt
-EventTime=20040119163904
```

```
-EventType=10
-PerceivedSeverity=4
-ProbableCause=332
-SpecificProblem=37
Database interface fault
Process out of memory
Process auto-restart
%A

%a
-RecordType=1
-ObjectOfReference=xt,p18
-EventTime=20040119163909
-EventType=2
-PerceivedSeverity=2
-ProbableCause=329
-SpecificProblem=729
No carrier state of
baseband module
%A

%a
-RecordType=1
-ObjectOfReference=xt,pwr
-EventTime=20040119163920
-EventType=4
-PerceivedSeverity=2
-ProbableCause=351
-SpecificProblem=705
-ProposedRepairAction=2
12V supply below threshold
%A

%a
-RecordType=1
-ObjectOfReference=xt,pwr
-EventTime=20040119163922
-EventType=4
-PerceivedSeverity=1
-ProbableCause=335
-SpecificProblem=711
Battery not fully charged
after complete discharge
%A

%a
-RecordType=1
-ObjectOfReference=xt,pwr
-EventTime=20040119163925
-EventType=4
-PerceivedSeverity=1
-ProbableCause=335
-SpecificProblem=719
AC input is not detected
%A

%a
-RecordType=1
-ObjectOfReference=xt,p18
-EventTime=20040119163929
-EventType=2
-PerceivedSeverity=5
-ProbableCause=305
-SpecificProblem=703
%A

%a
-RecordType=1
-ObjectOfReference=xt,pm
-EventTime=20040119163931
-EventType=11
-PerceivedSeverity=3
-ProbableCause=351
-SpecificProblem=551
-ThresholdIndication=3
-ThresholdInfo=f/t>0.3
%A

%a
-RecordType=1
```

```
-ObjectOfReference=xt,pm
-EventTime=20040119163932
-EventType=11
-PerceivedSeverity=2
-ProbableCause=351
-SpecificProblem=551
-TrendIndication=2
-ThresholdIndication=3
-ThresholdInfo=f/t>0.5
%A
#HB
```

References

[1]	*ENM Glossary* 1/0033-AOM 901 151
[2]	*Typographic Conventions* 3/1551-FCK 101 05