EXata 5.1
Network Management Model Library

August 2013

SCALABLE Network Technologies, Inc.

600 Corporate Pointe, Suite 1200
Culver City, CA 90230

+1.310.338.3318 TEL
+1.310.338.7213 FAX

SCALABLE-NETWORKS.COM

SCALABLE Network Technologies, Inc.

600 Corporate Pointe, Suit 1200

Culver City, CA 90230

+1.310.338.3318 TEL
+1.310.338.7213 FAX

SCALABLE-NETWORKS.COM

Table of Contents

1 Overview of Model Library

1.1 List of Models in the Library

The models described in the Network Management Model Library are listed in Table 1-1.

TABLE 1-1. Network Management Library Models

Model Name	Model Type	Section Number
Simple Network Management Protocol (SNMP)	Network Management	Section 2.1

1.2 Conventions Used

1.2.1 Format for Command Line Configuration

This section describes the general format for specifying parameters in input files, the precedence rules for parameters, and the conventions used in the description of command line configuration for each model.

1.2.1.1 General Format of Parameter Declaration

The general format for specifying a parameter in an input file is:

```
[<Qualifier>] <Parameter Name> [<Index>] <Parameter Value>
```

where

`<Qualifier>` The qualifier is optional and defines the scope of the parameter declaration. The scope can be one of the following: Global, Node, Subnet, and Interface. Multiple instances of a parameter with different qualifiers can be included in an input file. Precedence rules (see Section 1.2.1.2) determine the parameter value for a node or interface.

Global: The parameter declaration is applicable to the entire scenario (to all nodes and interfaces), subject to precedence rules. The scope of a parameter declaration is global if the qualifier is not included in the declaration.

Example:

```
MAC-PROTOCOL          MACDOT11
```

Node: The parameter declaration is applicable to specified nodes, subject to precedence rules. The qualifier for a node-level declaration is a list of space-separated node IDs or a range of node IDs (specified by using the keyword `thru`) enclosed in square brackets.

Example:

```
[5 thru 10] MAC-PROTOCOL          MACDOT11
```

Subnet: The parameter declaration is applicable to all interfaces in specified subnets, subject to precedence rules. The qualifier for a subnet-level declaration is a space-separated list of subnet addresses enclosed in square brackets. A subnet address can be specified in the IP dot notation or in the EXata N syntax.

Example:

```
[N8-1.0 N2-1.0] MAC-PROTOCOL      MACDOT11
```

Interface: The parameter declaration is applicable to specified interfaces. The qualifier for an interface-level declaration is a space-separated list of subnet addresses enclosed in square brackets.

Example:

```
[192.168.2.1 192.168.2.4] MAC-PROTOCOL MACDOT11
```

`<Parameter Name>`	Name of the parameter.
`<Index>`	Instance of the parameter to which this parameter declaration is applicable, enclosed in square brackets. This should be in the range 0 to *n* -1, where *n* is the number of instances of the parameter. The instance specification is optional in a parameter declaration. If an instance is not included, then the parameter declaration is applicable to all instances of the parameter, unless otherwise specified.
`<Parameter Value>`	Value of the parameter.

Note: There should not be any spaces between the parameter name and the index.

Examples of parameter declarations in input files are:

```
PHY-MODEL                                          PHY802.11b
[1] PHY-MODEL                                      PHY802.11a
[N8-1.0] PHY-RX-MODEL                              BER-BASED
[8 thru 10] ROUTING-PROTOCOL                       RIP
[192.168.2.1 192.168.2.4] MAC-PROTOCOL             GENERICMAC
NODE-POSITION-FILE                                 ./default.nodes
PROPAGATION-CHANNEL-FREQUENCY[0]                   2.4e9
[1 2] QUEUE-WEIGHT[1]                              0.3
```

Note: In the rest of this document, we will not use the qualifier or the index in a parameter's description. Users should use a qualifier and/or index to restrict the scope of a parameter, as appropriate.

1.2.1.2 Precedence Rules

Parameters without Instances

If the parameter declarations do not include instances, then the following rules of precedence apply when determining the parameter values for specific nodes and interfaces:

Interface > Subnet > Node > Global

This can be interpreted as follows:

- The value specified for an interface takes precedence over the value specified for a subnet, if any.
- The value specified for a subnet takes precedence over the value specified for a node, if any.
- The value specified for a node takes precedence over the value specified for the scenario (global value), if any.

Parameters with Instances

If the parameter declarations are a combination of declarations with and without instances, then the following precedence rules apply (unless otherwise stated):

Interface[i] > Subnet[i] > Node[i] > Global[i] > Interface > Subnet > Node > Global

This can be interpreted as follows:

- Values specified for a specific instance (at the interface, subnet, node, or global level) take precedence over values specified without the instance.

- For values specified for the same instance at different levels, the following precedence rules apply:
  - The value specified for an interface takes precedence over the value specified for a subnet, if any, if both declarations are for the same instance.
  - The value specified for a subnet takes precedence over the value specified for a node, if any, if both declarations are for the same instance.
  - The value specified for a node takes precedence over the value specified for the scenario (global value), if any, if both declarations are for the same instance.

1.2.1.3 Parameter Description Format

In the Model Library, most parameters are described using a tabular format described below. The parameter description tables have three columns labeled “Parameter”, “Values”, and “Description”. Table 1-2 shows the format of parameter tables. Table 1-4 shows examples of parameter descriptions in this format.

TABLE 1-2. Parameter Table Format

Parameter	Values	Description
<Parameter Name>	*<Type>*	*<Description>*
<Designation>	[<Range>]	
<Scope>	[<Default Value>]	
[<Instances>]	[<Unit>]	

Parameter Column

The first column contains the following entries:

- ***<Parameter Name>*:** The first entry is the parameter name (this is the exact name of the parameter to be used in the input files).
- ***<Designation>:*** This entry can be *Optional* or *Required*. These terms are explained below.
  - ***Optional***: This indicates that the parameter is optional and may be omitted from the configuration file. (If applicable, the default value for this parameter is included in the second column.)
  - ***Required***: This indicates that the parameter is mandatory and must be included in the configuration file.
- ***<Scope>:*** This entry specifies the possible scope of the parameter, i.e., if the parameter can be specified at the global, node, subnet, or interface levels. Any combination of these levels is possible.If the parameter can be specified at all four levels, the keyword “All” is used to indicate that.

  Examples of scope specification are:

  Scope: All

  Scope: Subnet, Interface

  Scope: Global, Node
- ***<Instances>:*** If the parameter can have multiple instances, this entry indicates the type of index. If the parameter can not have multiple instances, then this entry is omitted.

Examples of instance specification are:

Instances: channel number

Instances: interface index

Instances: queue index

Values Column

The second column contains the following information:

- **<*Type*>:** The first entry is the parameter type and can be one of the following: Integer, Real, String, Time, Filename, IP Address, Coordinates, Node-list, or List. If the type is a List, then all possible values in the list are enumerated below the word "List". (In some cases, the values are listed in a separate table and a reference to that table is included in place of the enumeration.)

  Table 1-3 shows the values a parameter can take for each type.

TABLE 1-3. Parameter Types

Type	Description
Integer	Integer value Examples: `2`, `10`
Real	Real value Examples: `15.0`, `-23.5`, `2.0e9`
String	String value Examples: `TEST`, `SWITCH1`
Time	Time value expressed in EXata time syntax (refer to *EXata User's Guide*) Examples: `1.5S`, `200MS`, `10US`

Filename	Name of a file in EXata filename syntax (refer to *EXata User's Guide*)
	Examples:
	`../../data/terrain/los-angeles-w`
	(For Windows and UNIX)
	`C:\scalable\exata\5.1\scenarios\WF\WF.nodes`
	(For Windows)
	`/root/scalable/exata/5.1/scenarios/WF/WF.nodes`
	(For UNIX)

Path	Path to a directory in EXata path syntax (refer to *EXata User's Guide*)
	Examples:
	`../../data/terrain` (For Windows and UNIX)
	`C:\scalable\exata\5.1\scenarios\default`
	(For Windows)
	`/root/scalable/exata/5.1/scenarios/default`
	(For UNIX)

IP Address	IPv4 or IPv6 address Examples: `192.168.2.1`, `2000:0:0:0::1`

TABLE 1-3. Parameter Types (Continued)

Type	Description
IPv4 Address	IPv4 address Examples: `192.168.2.1`
IPv6 Address	IPv6 address Examples: `2000:0:0:0::1`
Coordinates	Coordinates in Cartesian or Lat-Lon-Alt system. The altitude is optional. Examples: `(100, 200, 2.5)`, `(-25.3478, 25.28976)`
Node-list	List of node IDs separated by commas and enclosed in "{" and "}". Examples: `{2, 5, 10}`, `{1, 3 thru 6}`
List	One of the enumerated values. Example: See the parameter `MOBILITY` in Table 1-4.

Note: If the parameter type is List, then options for the parameter available in EXata and the commonly used model libraries are enumerated. Additional options for the parameter may be available if some other model libraries or addons are installed. These additional options are not listed in this document but are described in the corresponding model library or addon documentation.

- ***<Range>*:** This is an optional entry and is used if the range of values that a parameter can take is restricted. The permissible range is listed after the label "*Range:*" The range can be specified by giving the minimum value, the maximum value, or both. If the range of values is not restricted, then this entry is omitted.

  If both the minimum and maximum values are specified, then the following convention is used to indicate whether the minimum and maximum values are included in the range:

`(min, max)`	`min` < parameter value < `max`
`[min, max)`	`min` ≤ parameter value < `max`
`(min, max]`	`min` < parameter value ≤ `max`
`[min, max]`	`min` ≤ parameter value ≤ `max`

  `min` (or `max`) can be a parameter name, in which case it denotes the value of that parameter.

  Examples of range specification are:

  Range: ≥ `0`
  Range: `(0.0, 1.0]`
  Range: `[1, MAX-COUNT]`
  Range: `[1S, 200S]`

Note: If an upper limit is not specified in the range, then the maximum value that the parameter can take is the largest value of the type (integer, real, time) that can be stored in the system.

- **<*Default*>:** This is an optional entry which specifies the default value of an optional or conditional-optional parameter. The default value is listed after the label "*Default:*"
- **<*Unit*>:** This is an optional entry which specifies the unit for the parameter, if applicable. The unit is listed after the label "*Unit:*". Examples of units are: meters, dBm, slots.

Description Column

The third column contains a description of the parameter. The significance of different parameter values is explained here, where applicable. In some cases, references to notes, other tables, sections in the User's Guide, or to other model libraries may be included here.

Table 1-4 shows examples of parameter descriptions using the format described above.

TABLE 1-4. Example Parameter Table

Parameter	Values	Description
MOBILITY Optional *Scope:* Global, Node	List: • NONE • FILE • GROUP-MOBILITY • RANDOM-WAYPOINT Default: NONE	Mobility model used for the node. If MOBILITY is set to NONE, then the nodes remain fixed in one place for the duration of the simulation. See Table 7-11 for a description of mobility models.
BACKOFF-LIMIT Required *Scope:* Subnet, Interface	Integer *Range:* [4,10) *Unit:* slots	Upper limit of backoff interval after collision. A backoff interval is randomly chosen between 1 and this number following a collision.
IP-QUEUE-PRIORITY-QUEUE-SIZE Required *Scope:* All *Instances:* queue index	Integer *Range:* [1, 65535] *Unit:* bytes	Size of the output priority queue.
MAC-DOT11-DIRECTIONAL-ANTENNA-MODE Optional *Scope:* All	List • YES • NO *Default:* NO	Indicates whether the radio is to use a directional antenna for transmission and reception.

1.2.2 Format for GUI Configuration

The GUI configuration section for a model outlines the steps to configure the model using the GUI. The following conventions are used in the GUI configuration sections:

Path to a Parameter Group

As a shorthand, the location of a parameter group in a properties editor is represented as a path consisting of the name of the properties editor, name of the tab within the properties editor, name of the parameter group within the tab (if applicable), name of the parameter sub-group (if applicable), and so on.

Example

The following statement:

Go to **Default Device Properties Editor > Interfaces > Interface # > MAC Layer**

is equivalent to the following sequence of steps:

1. Open the Default Device Properties Editor for the node.
2. Click the **Interfaces** tab.
3. Expand the applicable Interface group.
4. Click the **MAC Layer** parameter group.

The above path is shown in Figure 1-1.

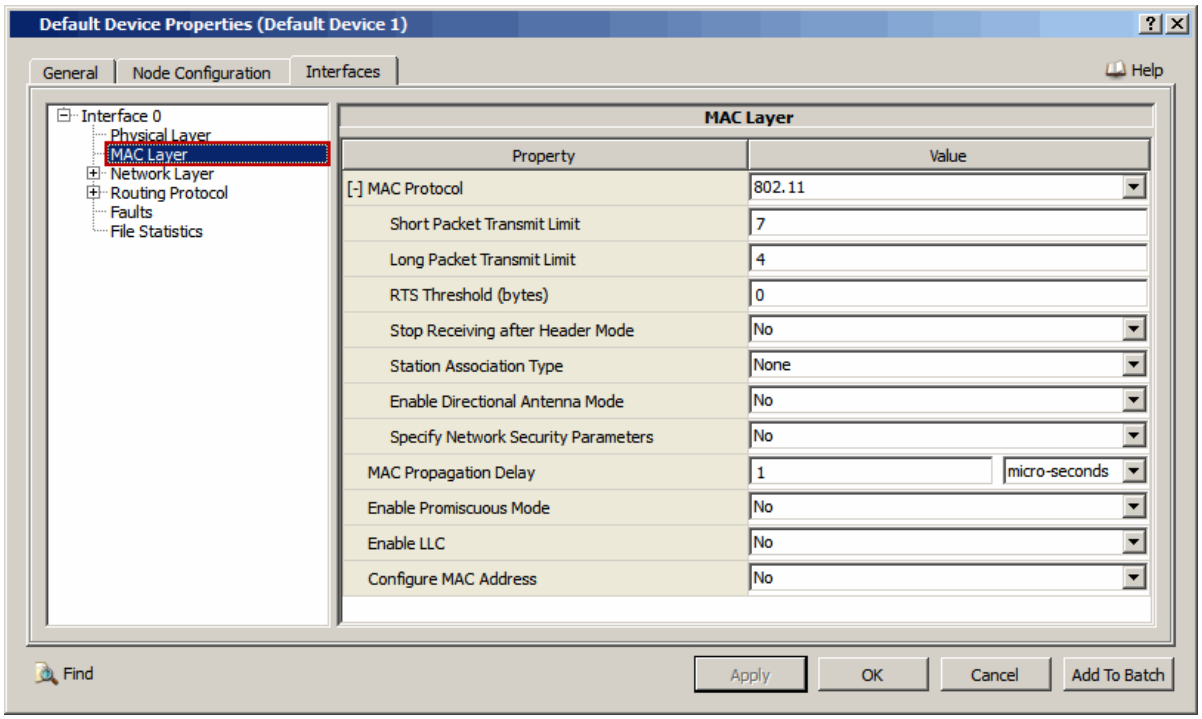


FIGURE 1-1. Path to a Parameter Group

Path to a Specific Parameter

As a shorthand, the location of a specific parameter within a parameter group is represented as a path consisting of all ancestor parameters and their corresponding values starting from the top-level parameter. The value of an ancestor parameter is enclosed in square brackets after the parameter name.

Example

The following statement:

> Set **MAC Protocol** *[= 802.11]* **> Station Association Type** *[= Dynamic]* **> Set Access Point** *[= Yes]* **>** *Enable Power Save Mode* to *Yes*

is equivalent to the following sequence of steps:

1. Set **MAC Protocol** to *802.11*.
2. Set **Station Association Type** to *Dynamic.*
3. Set **Set Access Point** to *Yes*.
4. Set **Enable Power Save Mode** to *Yes*.

The above path is shown in Figure 1-2.

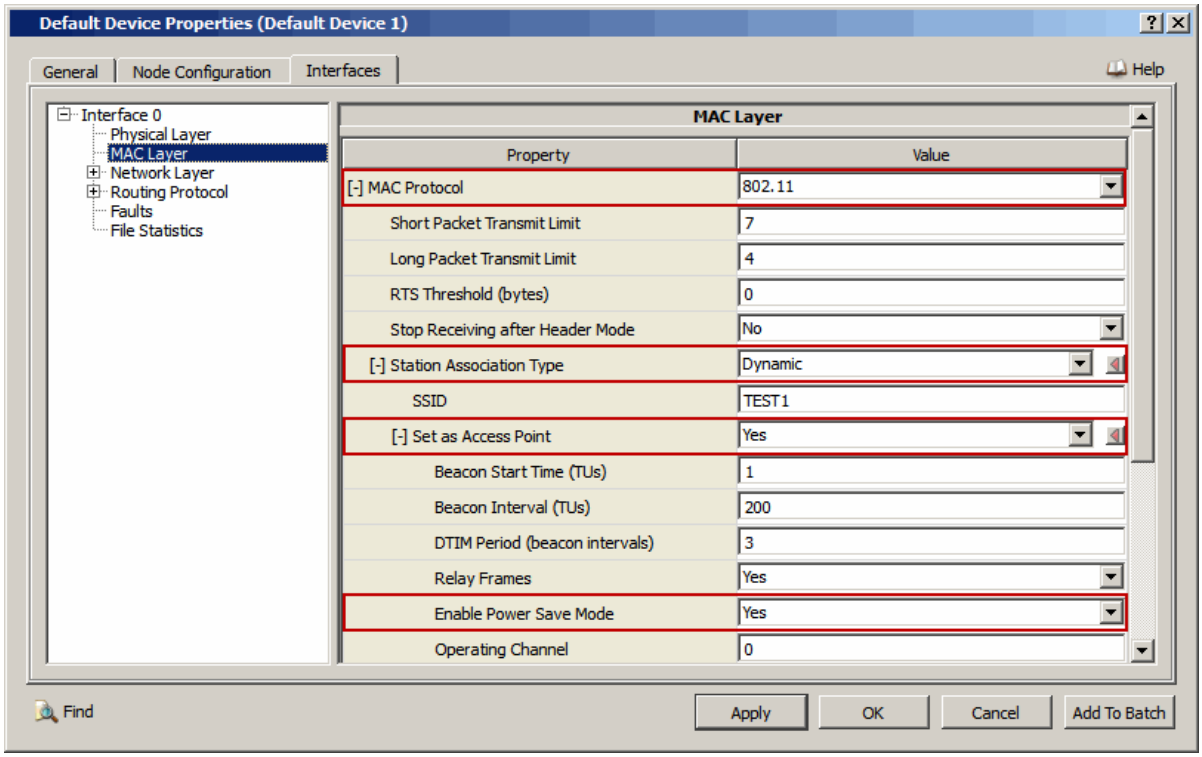


FIGURE 1-2. Path to a Specific Parameter

Parameter Table

GUI configuration of a model is described as a series of a steps. Each step describes how to configure one or more parameters. Since the GUI display name of a parameter may be different from the name in the configuration file, each step also includes a table that shows the mapping between the GUI names and command line names of parameters configured in that step. For a description of a GUI parameter, see the description of the equivalent command line parameter in the command line configuration section.

The format of a parameter mapping table is shown in Table 1-5.

TABLE 1-5. Mapping Table

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
<GUI Display Name>	*<Scope>*	*<Command Line Parameter Name>*

The first column, labeled “GUI Parameter”, lists the name of the parameter as it is displayed in the GUI.

The second column, labeled “Scope of GUI Parameter”, lists the level(s) at which the parameter can be configured. <*Scope*> can be any combination of: Global, Node, Subnet, Wired Subnet, Wireless Subnet, Point-to-point Link, and Interface.

Table 1-6 lists the Properties Editors where parameters with different scopes can be set.

Notes: 1. Unless otherwise stated, the “Subnet” scope refers to “Wireless Subnet”.

2. The scope column can also refer to Properties Editors for special devices and network components (such as ATM Device Properties Editor) which are not included in Table 1-6.

TABLE 1-6. Properties Editors for Different Scopes

Scope of GUI Parameter	Properties Editor
Global	Scenario Properties Editor
Node	Default Device Properties Editor (General and Node Configuration tabs)
Subnet Wireless Subnet	Wireless Subnet Properties Editor
Wired Subnet	Wired Subnet Properties Editor
Point-to-point Link	Point-to-point Link Properties Editor
Interface	Interface Properties Editor, Default Device Properties Editor (Interfaces tab)

The third column, labeled “Command Line Parameter”, lists the equivalent command line parameter.

Note: For some parameters, the scope may be different in command line and GUI configurations (a parameter may be configurable at fewer levels in the GUI than in the command line).

Table 1-7 is an example of a parameter mapping table.

TABLE 1-7. Example Mapping Table

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Define Area	Node	`OSPFv2-DEFINE-AREA`
OSPFv2 Configuration File	Node	`OSPFv2-CONFIG-FILE`
Specify Autonomous System	Node	N/A

TABLE 1-7. Example Mapping Table (Continued)

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Configure as Autonomous System Boundary Router	Node	AS-BOUNDARY-ROUTER
Inject External Route	Node	N/A
Enable Stagger Start	Node	OSPFv2-STAGGER-START

2 Network Management Models

This chapter describes features, configuration requirements and parameters, statistics, and scenarios for Network Mangement Models, and consists of the following sections:

- Simple Network Management Protocol (SNMP)

2.1 Simple Network Management Protocol (SNMP)

2.1.1 Description

Simple Network Management Protocol (SNMP) is a UDP-based network protocol which runs over IP using Port 161 and 162. It is used mostly in network management systems to monitor network-attached devices for conditions that warrant administrative attention. SNMP makes management data available in the form of variables on the managed systems, which describe the system configuration. These variables can then be queried (and sometimes set) by managing applications.

EXata offers the capability to manage nodes in a scenario by an SNMP manager. The SNMP managers can review the current network status, set the network properties, or assign *traps* to receive feedback from the managed nodes. EXata provides this feature by implementing SNMP agents on nodes in a scenario. SNMP agents can be enabled on all nodes and can be configured to handle the SNMP *get* and *se*t commands. Additional configuration is required to handle the *trap* command.

2.1.2 Features and Assumptions

This section describes the implemented features, omitted features, and assumptions and limitations of the SNMP model.

2.1.2.1 Implemented Features

- Command responder application
- Notification originator
- User-based Security Module (for SNMPv3)
- Authentication (for SNMPv3)
- Encryption (for SNMPv3)
- Access Control

2.1.2.2 Omitted Features

- This model supports SNMPv1, SNMPv2c and SNMPv3 only. SNMPv2 and SNMPv2u are not supported.
- This model provides only SNMP client features. Third-party SNMP managers must be used to interact with the SNMP client model.

2.1.2.3 Assumptions and Limitations

- Groups present in MIB-II are supported except EGP and Transmission.
- EXata is compiled with OpenSSL support.
- SNMP *Coldstart trap* is send at the initialization of scenario. In a wireless scenario, these traps may be dropped because it takes some time for routes to converge. In order to receive these at the manager, static routes can be configured.
- The implementation of the SNMP model is in accordance with Net-SNMP. Therefore, errors like TOO-BIG and GEN-ERR are never sent.
- Multiple SNMP managers cannot be configured for an SNMP client.

2.1.3 Command Line Configuration

To enable the SNMP agent, include the following parameter in the scenario configuration (.config) file:

```
[<Qualifier>]SNMP-ENABLED      YES
```

The scope of this parameter declaration can be Global or Node.

Note: The default value of SNMP-ENABLED is NO.

SNMP Agent Parameters

Table 2-1 lists the SNMP Agent configuration parameters.

TABLE 2-1. SNMP Agent Parameters

Parameter	Value	Description
SNMP-VERSION *Optional* *Scope:* Global, Node	List: • 1 • 2 • 3 *Default:* 1	SNMP version for the SNMP agent.
SNMP-NOTIFICATION-TYPE *Optional* *Scope:* Global, Node	List: • INFORM • NONE • TRAP *Default:* NONE	Enables trap or inform on the SNMP agent.
SNMP-MANAGER-ADDRESS *Optional* *Scope:* Global, Node	IPv4 Address *Default:* 192.168.0.1	Address of the SNMP manager where the trap information is relayed.
SNMPD-CONF-FILE *Required* *Scope:* Global, Node	Filename	Name of the SNMP Agent Configuration file. This file should have the extension “.conf”. The format of the SNMP Agent Configuration file is described in Section 2.1.3.1.

2.1.3.1 Format of the SNMP Agent Configuration File

This section describes the format of the SNMP Agent Configuration file. The configuration file should have extension ".conf" (e.g. agent.conf). Parameters can be entered in the file in any order.

- To specify the read-only access community name (only for SNMPv1 and SNMPv2c), enter the following line in the SNMP Agent Configuration file:

  ```
  rocommunity <community-name>
  ```

  where

  `<community-name>` Name of the community.

  Example:

  ```
  rocommunity public
  ```

- To specify the read-write access community name (only for SNMPv1 and SNMPv2c), enter the following line in the SNMP Agent Configuration file:

  ```
  rwcommunity <community-name>
  ```

  where

  `<community-name>` Name of the community.

  Example:

  ```
  rwcommunity private
  ```

- To create a USM-based SNMP user (only for SNMPv3), enter the following line in the SNMP Agent Configuration file (all parameters should be entered on the same line):

```
createUser [-e <engine-ID>] <user-name> <hash-algorithm> <auth-phrase>
<encrypt-algorithm> <priv-phrase>
```

where

`<engine-ID>`	Administratively unique identifier for the SNMPv3 engine. Specification of `<engine-ID>` is optional.
`<user-name>`	SNMPv3 authentication user name.
`<hash-algorithm>`	Hashing algorithm to be used. This can be `MD5` or `SHA`. `MD5`: Message Digest Algorithm `SHA`: Secure Hash Algorithm
`<auth-phrase>`	Authentication key. This is a string enclosed in `"` and `"`.
`<encrypt-algorithm>`	Encryption algorithm to be used. This can be `DEA` or `EAS`. `DEA`: Data Encryption Standard `EAS`: Advanced Encryption Standard
`<priv-algorithm>`	Privacy key. This is a string enclosed in `"` and `"`.

Example:

```
createUser user MD5 "password" DES "password1"
```

- To specify a read-write user name (only for SNMPv3), enter the following line in the SNMP Agent Configuration file:

  ```
  rwuser <user-name> [<security>]
  ```

  where

  `<user-name>` SNMPv3 authentication user name.

  `<security>` Security type.

  This can be `auth`, `noauth`, or `priv`.

  - `auth`: Authentication and no privacy
  - `noauth`: No authentication and no privacy
  - `priv`: Authentication and privacy

  Specification of security type is optional. The default security level in SNMPv3 is authentication and no privacy.

  Example:

  ```
  rwuser user priv
  ```

- To specify a read-only user name (only for SNMPv3), enter the following line in the SNMP Agent Configuration file:

  ```
  rouser <user-name> <security>
  ```

  where

  `<user-name>` SNMPv3 authentication user name.

  `<security>` Security type.

  This can be `auth`, `noauth`, or `priv`.

  - `auth`: Authentication and no privacy
  - `noauth`: No authentication and no privacy
  - `priv`: Authentication and privacy

  Specification of security type is optional. The default security level in SNMPv3 is authentication and no privacy.

  Example:

  ```
  rouser default noauth
  ```

- To define a view (only for SNMPv3), enter the following line in the SNMP Agent Configuration file:

```
view <view-name> <type> <OID> [<mask>]
```

where

`<view-name>`	View name to build up a more complex collection of OIDs.
`<type>`	Type of view for an OID. This can be `included` or `excluded`. This is used to define a more complex view, e.g., by excluding certain sensitive objects from an otherwise accessible subtree.
`<OID>`	Identifier used to name an object of a MIB table.
`<mask>`	List of hex octets, optionally separated by `.` or `:`, enclosed in `[` and `]`, with the set bits indicating which sub-identifiers in the view OID to match against. Specification of `<mask>` is optional. If it is not specified, this defaults to matching the OID exactly (all bits set), thus defining a simple OID subtree.

Examples:

```
view demoWrite included .1.3.6.1.2.1.1.6
view demoRead excluded .1.3.6.1.2.1.4
view demoA included .1.3.6.1.2.1  [1.1.1.1.1.1]
view demoB included .1.3.6.1.2.1.1.1 [1.1.1]
view demoC included .1.3.6.1.2.1.2
view demoD included .1.3.6.1.2.1.1 [1.1.0.1.0.1.1]
view demoE included .1.3.6.1.2.1.2  [1.1.0.1.0]
view demoF included .1.3.6.1.2.11  [1.1.0.1.0.1]
```

- To map a community name to a security name (for SNMPv1 and SNMPv2c), enter the following line in the SNMP Agent Configuration file:

```
com2sec <name> <source> <community>
```

where

`<name>`	Security name.
`<source>`	Restricted source. This can be `default` (indicating global) or a range of source addresses. A restricted source can either be a specific hostname or address, or a subnet represented in the IP-address/mask format.
`<community>`	Name of the community.

Examples:

```
com2sec v1User default demopublic_v1
com2sec v2cUser default demopublic
com2sec v1User 10.10.10.1 demopublic
com2sec v2cUser 10.10.10.0/255.255.255.0 demopublic
com2sec v2cUser 10.10.10.0/24 demopublic
```

- To map a security name to a named group, enter the following line in the SNMP Agent Configuration file:

```
group <name> <model> <security>
```

where

`<name>`	Group name.
`<model>`	Security model. This can be `ksm`, `tsm`, `usm`, `v1`, or `v2c`. `ksm`: Kerberos Security Model `tsm`: Transport Security Model `usm`: User Security Model `v1`: SNMP Version 1 Security Model `v2c`: SNMP Version 2c Security Model
`<security>`	Security name.

Examples:

```
group demogroup1 v1  v1User
group demogroup1 v2c v2cUser
group demogroup3 usm usmUser
group demogroup4 usm  usmUser1
```

- To map a group of users or communities to a view, enter the following line in the SNMP Agent Configuration file (all parameters should be entered on the same line):

```
 access <name> [<context>] <model> <level> <match-type> <read> <write>
<notify>
```

where

`<name>` Name of group of users or communities.

`<context>` String indicating the context name.

Specification of `<context>` is optional. If it is omitted, it defaults to the null string, `""`.

`<model>` Security model.

This can be `any`, `ksm`, `tsm`, `usm`, `v1`, or `v2c`.

- `any`: Any of the security models listed below
- `ksm`: Kerberos Security Model
- `tsm`: Transport Security Model
- `usm`: User Security Model
- `v1`: SNMP Version 1 Security Model
- `v2c`: SNMP Version 2c Security Model

`<level>` Authentication level.

This can be `auth`, `noauth`, or `priv`.

- `auth`: Authentication and no privacy
- `noauth`: No authentication and no privacy
- `priv`: Authentication and privacy

Note: For v1 or v2c access, `<level>` should be `noauth`.

`<match-type>` Specifies how `<context>` should be matched against the context of the incoming request, either exactly or prefix-only.

`<match-type>` can be `exact` or `prefix`.

`<read>` View name to be used for `GET` requests.

`<write>` View name to be used for `SET` requests.

`<notify>` View name to be used for `TRAP/INFORM` requests.

Note: `<notify>` is not currently used.

Examples:

```
access demogroup1 ""   any   noauth    prefix    demoA none none
access demogroup1 ""   any   noauth    prefix    demoB none none
access demogroup2 ""   any   noauth    prefix    demoD none none
access demogroup2 ""   any   auth      prefix    demoRead none none
access demogroup3 ""   any   noauth    prefix    demoA none none
access demogroup3 ""   any   noauth    prefix    demoB none none
access demogroup3 ""   any   noauth    prefix   demoD none none
access demogroup3 ""   any   auth      prefix   demoRead none none
access demogroup4 ""   any   noauth    prefix   demoRead demoWrite none
```

2.1.4 GUI Configuration

To configure the SNMP Agent at a node in the GUI, do the following:

1. Go to **Default Device Properties Editor > Node Configuration > Network Management**.
2. Set **Enable SNMP Agent** to *Yes.*

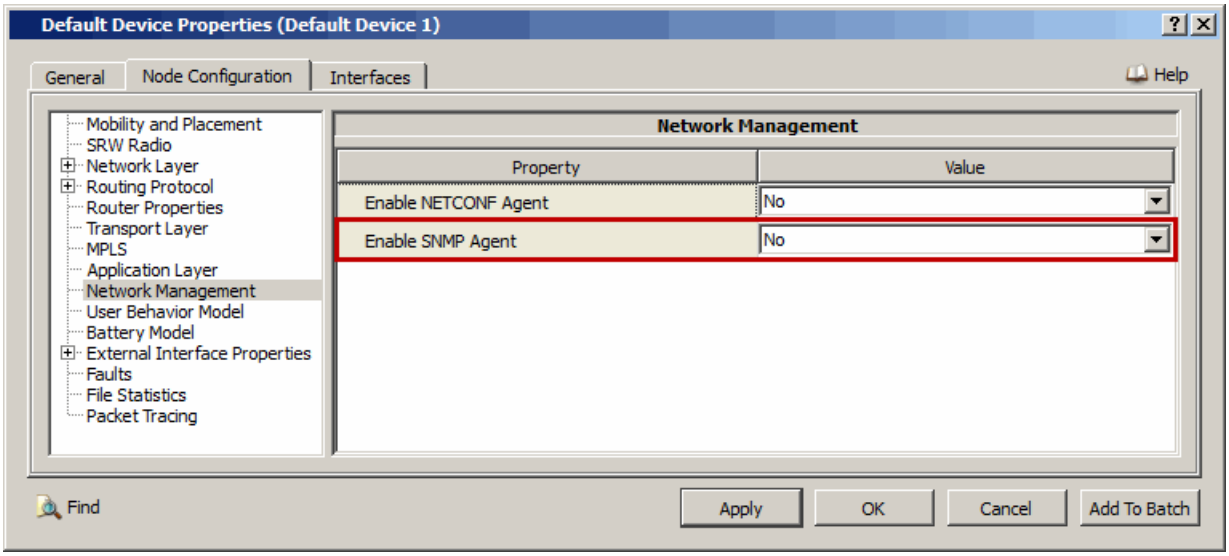


FIGURE 2-1. Enabling SNMP

TABLE 2-2. Command Line Equivalent of SNMP Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
Enable SNMP Agent	Node	SNMP-ENABLED

3. Set the SNMP parameters listed in Table 2-3.

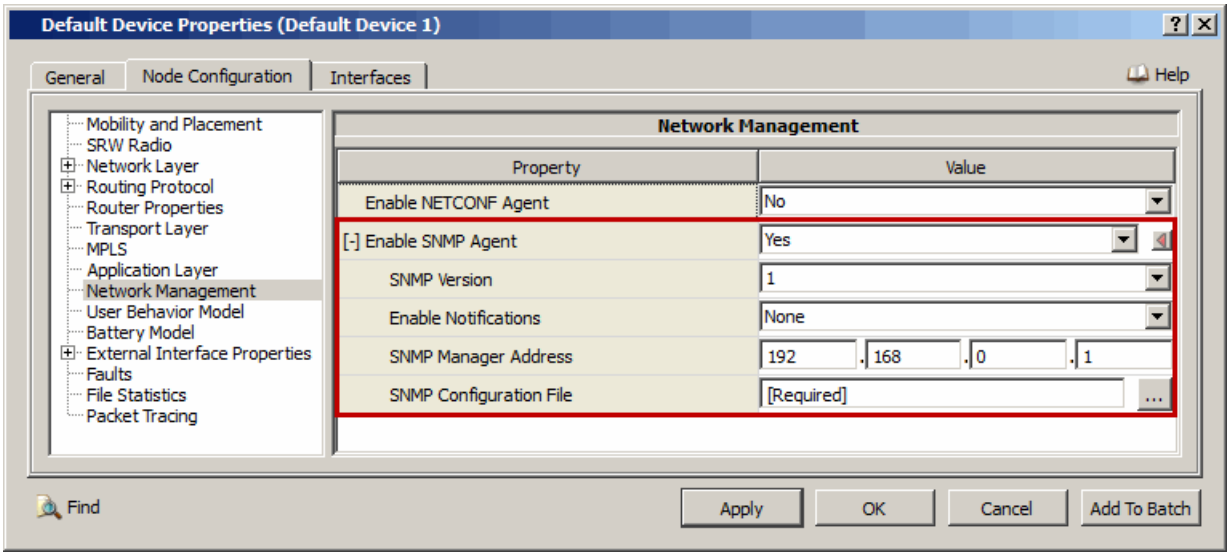


FIGURE 2-2. Configuring SNMP Agent Parameters

TABLE 2-3. Command Line Equivalent of SNMP Agent Configuration Parameters

GUI Parameter	Scope of GUI Parameter	Command Line Parameter
SNMP Version	Node	SNMP-VERSION
Enable Notifications	Node	SNMP-NOTIFICATION-TYPE
SNMP Manager Address	Node	SNMP-MANGER-ADDRESS
SNMP Configuration File	Node	SNMPD-CONF-FILE

2.1.5 Using SNMP

EXata implements an SNMP agent on each emulated node. An SNMP manager can communicate with these SNMP agents to manage the nodes in the emulated network. The same SNMP manager may also manage physical nodes simultaneously. The EXata SNMP agent gathers data from the management information base (MIB), which is the repository for information about device parameters and network data, and translates that information into a form compatible with SNMP. The EXata SNMP agent can also send traps to a management system, such as HP OpenView, IBM Tivoli, or SolarWinds Orion.

Using SNMP Manager

To use an SNMP manager with EXata, do the following:

1. Connect an operational host to the emulation server. Configure one of the emulated nodes as the default emulated node for the operational host, as described in *EXata User's Guide*.

2. Run the SNMP manager on the operational host. This in effect runs the SNMP manager on the default emulated node.

The SNMP manager can now be used to manage the various MIBs and Traps via the SNMP agent running on each emulated node.

SNMP Features

The EXata SNMP agent follows the SNMPv1 protocol standard and supports a comprehensive list of SNMP MIBs defined in accordance with the MIB-II standard. The agent implements the SNMP-GET, SNMP-GET-NEXT, SNMP-SET, and SNMP-TRAP messages. The list of MIBs supported includes System MIB, TCP MIB, UDP MIB, IP MIB, Interfaces MIB, and 802.11 MIB.

Figure 2-3 shows a snapshot of an off-the-shelf SNMP manager (LoriotPro) displaying nodes within an EXata scenario. In the figure, the manager has discovered these emulated nodes using ICMP messages.

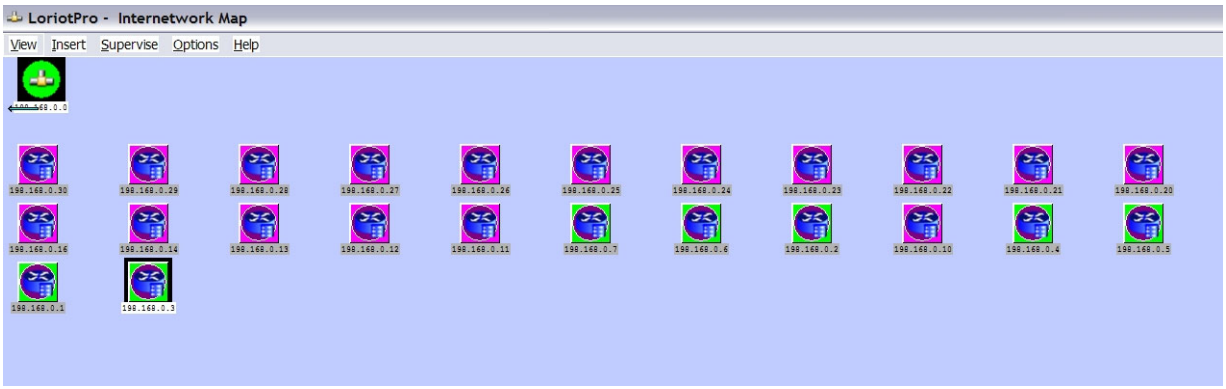
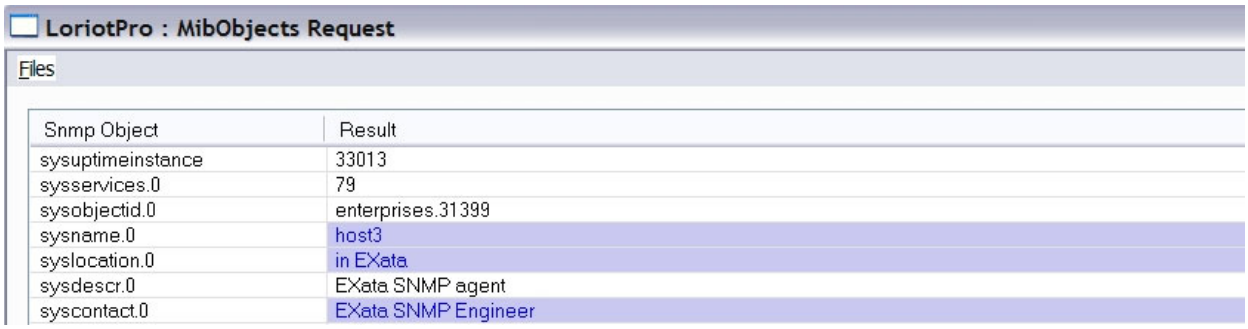


FIGURE 2-3. SNMP Manager Displaying Nodes Running EXata SNMP Agents

Figure 2-4, Figure 2-5, Figure 2-6, Figure 2-7 and Figure 2-8 show System, TCP, UDP, Interface, and IP MIBs respectively.



Snmp Object	Result
sysuptimeinstance	33013
sysservices.0	79
sysobjectid.0	enterprises.31399
sysname.0	host3
syslocation.0	in EXata
sysdescr.0	EXata SNMP agent
syscontact.0	EXata SNMP Engineer

FIGURE 2-4. System MIB Describing the EXata SNMP Agent

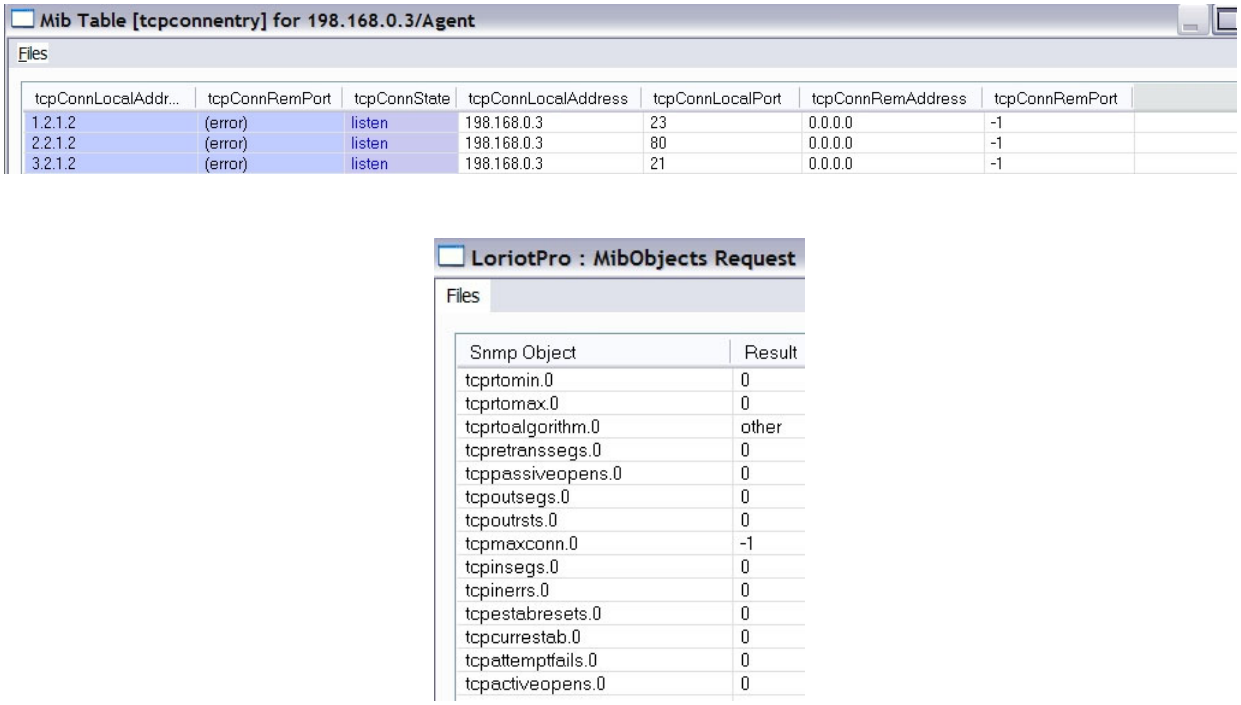


FIGURE 2-5. TCP MIB Connection Table Listing the Open Ports Along with TCP Statistics

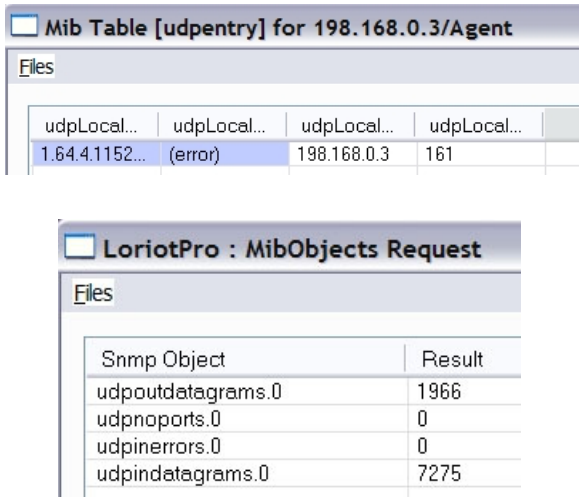


FIGURE 2-6. UDP MIB Showing UDP Table Entry and UDP Datagram Statistics for a Given Node

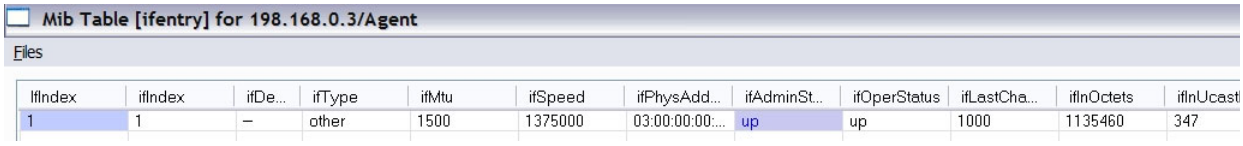


FIGURE 2-7. IF MIB Detailing the Interface Information for a Given Node

Mib Table [ipaddrentry] for 198.168.0.3/Agent

Files

ipAdEntA...	ipAdEntA...	ipAdEntIfI...	ipAdEntN...	ipAdEntB...	ipAdEntR
1.64.4.1152...	198.168.0.3	1	255.255.255...	1	0

LoriotPro : MibObjects Request

Files

Snmp Object	Result
iproutingdiscards.0	0
ipreasmtimeout.0	0
ipreasmreqds.0	0
ipreasmoks.0	0
ipreasmfails.0	0
ipoutrequests.0	2248
ipoutnoroutes.0	0
ipoutdiscards.0	0
ipinunknownprotos.0	0
ipinreceives.0	8270
ipinhdrerrors.0	0
ipindiscards.0	0
ipindelivers.0	8269
ipinaddrerrors.0	0
ipfragoks.0	0
ipfragfails.0	0
ipfragcreates.0	0
ipforwdatagrams.0	1
ipforwarding.0	forwarding
ipdefaultttl.0	64

Mib Table [iprouteentry] for 198.168.0.3/Agent

Files

ipRouteDest	ipRouteDest	ipRouteIfI...	ipRouteM...	ipRouteM...	ipRouteM...	ipRouteM...	ipRouteN...	ipRouteT...	ipRoutePr...	ipRoute
1.64.4.1152...	198.168.0.30	1	3	-1	-1	-1	198.168.0.15	indirect	other	0
2.64.4.1152...	198.168.0.29	1	3	-1	-1	-1	198.168.0.15	indirect	other	0
3.64.4.1152...	198.168.0.28	1	3	-1	-1	-1	198.168.0.15	indirect	other	0
4.64.4.1152...	198.168.0.27	1	3	-1	-1	-1	198.168.0.9	indirect	other	0
5.64.4.1152...	198.168.0.26	1	3	-1	-1	-1	198.168.0.7	indirect	other	0
6.64.4.1152...	198.168.0.25	1	4	-1	-1	-1	198.168.0.7	indirect	other	0
7.64.4.1152...	198.168.0.24	1	3	-1	-1	-1	198.168.0.15	indirect	other	0
8.64.4.1152...	198.168.0.23	1	2	-1	-1	-1	198.168.0.15	indirect	other	0
9.64.4.1152...	198.168.0.22	1	2	-1	-1	-1	198.168.0.9	indirect	other	0
10.64.4.115...	198.168.0.21	1	2	-1	-1	-1	198.168.0.9	indirect	other	0
11.64.4.115...	198.168.0.20	1	3	-1	-1	-1	198.168.0.9	indirect	other	0
12.64.4.115...	198.168.0.19	1	2	-1	-1	-1	198.168.0.7	indirect	other	0
13.64.4.115...	198.168.0.18	1	3	-1	-1	-1	198.168.0.15	indirect	other	0
14.64.4.115...	198.168.0.17	1	2	-1	-1	-1	198.168.0.15	indirect	other	0
15.64.4.115...	198.168.0.16	1	2	-1	-1	-1	198.168.0.9	indirect	other	0
16.64.4.115...	198.168.0.15	1	1	-1	-1	-1	198.168.0.15	direct	other	0
17.64.4.115...	198.168.0.14	1	2	-1	-1	-1	198.168.0.7	indirect	other	0
18.64.4.115...	198.168.0.13	1	2	-1	-1	-1	198.168.0.7	indirect	other	0
19.64.4.115...	198.168.0.12	1	2	-1	-1	-1	198.168.0.5	indirect	other	0
20.64.4.115...	198.168.0.11	1	2	-1	-1	-1	198.168.0.4	indirect	other	0
21.64.4.115...	198.168.0.10	1	1	-1	-1	-1	198.168.0.10	direct	other	0

FIGURE 2-8. IP MIB Entry Showing IP Address, Routing and Statistics Information for a Given Node

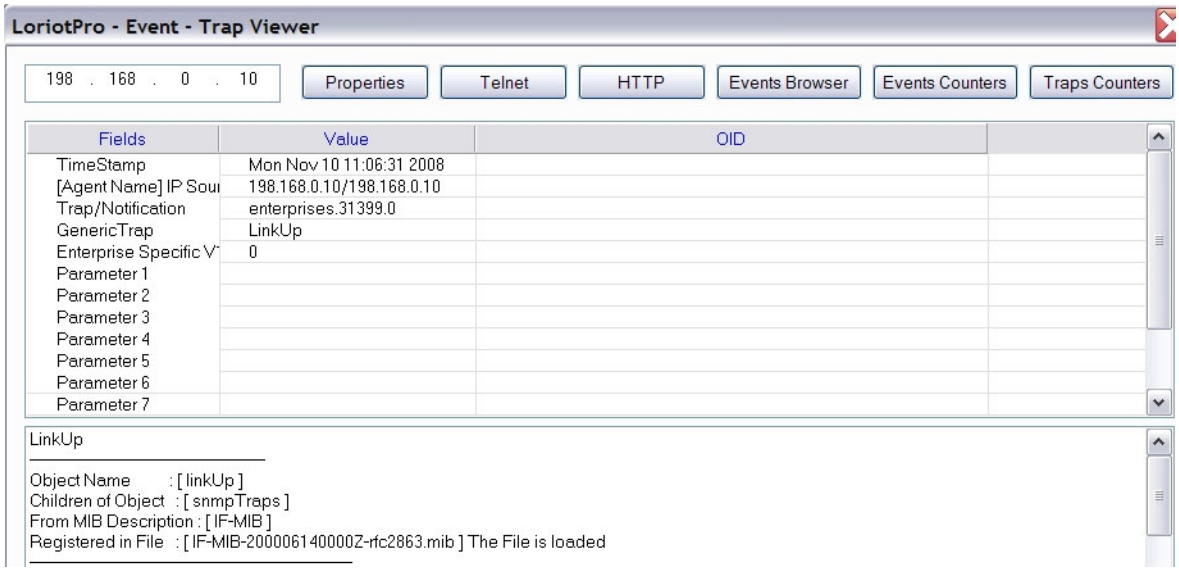


FIGURE 2-9. SNMP Trap Message Indicating Restart of Node 10 Following a Link Break

Note that to capture traps, the scenario configuration file should include the correct SNMP community string to which the manager belongs, and the correct IP address of the manager, which in this case is the IP address of the emulated node it is mapped to.

2.1.6 Statistics

Table 2-4 lists the SNMP statistics that are output to the statistics (.stat) file at the end of emulation.

TABLE 2-4. SNMP Statistics

Statistic	Description
Snmp Agent,Total in packets	Total number of packets received by the agent node.
Snmp Agent,Total out packets	Total number of packets sent by the agent node.
Snmp Agent,Total Get requests	Total number of Get-request commands received by the agent node.
Snmp Agent,Total GetNext requests	Total number of GetNext-request commands received by the agent node.
Snmp Agent,Total GetBulk requests	Total number of GetBulk-request commands received by the agent node.
Snmp Agent,Total responses	Total number of Responses sent by the agent node.
Snmp Agent,Total traps sent	Total number of Traps sent by the agent node.
Snmp Agent,Total informs sent	Total number of Informs sent by the agent node.
Snmp Agent,Total inform ack received	Total number of Inform acknowledgements received by the agent node.

2.1.7 References

1. RFC3411: “An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks.” D. Harrington, R. Presuhn, B. Wijnen. (Dec 2002)

2. RFC3412: “Message Processing and Dispatching for the Simple Network Management Protocol (SNMP).” J. Case, D. Harrington, R. Presuhn, B. Wijnen. (Dec 2002)
3. RFC3413: “Simple Network Management Protocol (SNMP) Applications.” D. Levi, P. Meyer, B. Stewart (Dec 2002)
4. RFC3414: “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3).” U. Blumenthal, B. Wijnen. (Dec 2002)
5. :RFC3415: “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP).” B. Wijnen, R. Presuhn, K. McCloghrie. (Dec 2002)
6. RFC1213: “Management Information Base for Network Management of TCP/IP-based internets: MIB-II.” K. McCloghrie. (march 1919)
7. http://net-snmp.sourceforge.net/wiki/index.php.